M1700

Double Prototyping M-Module

PUBLICATION NO. 980877

RACAL INSTRUMENTS

United States

(Corporate Headquarters and Service Center)
4 Goodyear St., Irvine, CA 92618-2002
Tel: (800) 722-2528, (949) 859-8999; FAX: (949-859-7139

5730 Northwest Parkway Suite 700, San Antonio, TX 78249
Tel: (210) 699-6799; FAX: (210) 699-8857

Europe

(European Corporate Headquarters and Service Center)
18 Avenue Dutartre, 78150 LeChesnay, France
Tel: +33 (0)1-39-3-22-22; FAX: +33 (0)1-39-23-22-25

29-31 Cobham Road, Wimborne, Dorset BH27-7PF United Kindom
Tel: +44 (0)-1202-872800; Fax: +44 (0)-1202-870810

Via Milazzo 25, 200892 Cinisello B, Milan, Italy
Tel: +39 (0)2-6123-901; FAX: +39 (0)2-6129-3606

Technologiepark Bergisch Gladbach, Friedrich-Ebert-Strasse, D-51429 Bergisch Gladbach, Germany
Tel: +49-2204-844200; FAX: +49-2204-844219

info@racalinstruments
sales@racalinstruments
helpdesk@racalinstruments
http://www.racalinstruments.com

PUBLICATION DATE: JULY 23, 2003

THANK YOU FOR PURCHASING THIS RACAL INSTRUMENTS PRODUCT.

For this product, or any other Racal Instruments product that incorporates software drivers, you may access our web site to verify and/or download the latest driver versions. The web address for driver downloads is:

http://www.racalinstruments.com/downloads

You will be asked to register one time only to gain access to the driver and product manual downloads sections. At registration a cookie will be placed on your computer if you choose to accept it. This is done to facilitate your use of these sections on future visits. You may refuse to accept the cookie and still have complete access to the software driver database but will have to re-register every time you visit. This cookie is for ease of use only and no information is gathered for, sold, or reported to, any third party organization.

If you have any questions about software driver downloads or our privacy policy, please contact us at

info@racalinstruments.com.

WARRANTY STATEMENT

All Racal Instruments, Inc. products are designed and manufactured to exacting standards and in full conformance to Racal's ISO 9001 procedures.

For the specific terms of your standard warranty, or optional extended warranty or service agreement, contact your Racal customer service advisor. Please have the following information available to facilitate service.

1. Product serial number
2. Product model number
3. Your company and contact information

You may contact your customer service advisor by:

E-Mail:	Helpdesk@racalinstruments.com	
Telephone:	+1 800 722 3262 +44(0) 8706 080134 +852 2405 5500	(USA) (UK) (Hong Kong)
Fax:	+1 949 859 7309 +44(0) 1628 662017 +852 2416 4335	(USA) (UK) (Hong Kong)

RETURN of PRODUCT

Authorization is required from Racal Instruments before you send us your product for service or calibration. Call your nearest Racal Instruments support facility. A list is located on the last page of this manual. If you are unsure where to call, contact Racal Instruments, Inc. Customer Support Department in Irvine, California, USA at 1-800-722-3262 or 1-949-859-8999 or via fax at 1-949-859-7139. We can be reached at: helpdesk@racalinstruments.com.

PROPRIETARY NOTICE

This document and the technical data herein disclosed, are proprietary to Racal Instruments, and shall not, without express written permission of Racal Instruments, be used, in whole or in part to solicit quotations from a competitive source or used for manufacture by anyone other than Racal Instruments. The information herein has been developed at private expense, and may only be used for operation and maintenance reference purposes or for purposes of engineering evaluation and incorporation into technical specifications and other documents which specify procurement of products from Racal Instruments.

DISCLAIMER

Buyer acknowledges and agrees that it is responsible for the operation of the goods purchased and should ensure that they are used properly and in accordance with this handbook and any other instructions provided by Seller. RII products are not specifically designed, manufactured or intended to be used as parts, assemblies or components in planning, construction, maintenance or operation of a nuclear facility, or in life support or safety critical applications in which the failure of the RII product could create a situation where personal injury or death could occur. Should Buyer purchase RII product for such unintended application, Buyer shall indemnify and hold RII, its officers, employees, subsidiaries, affiliates and distributors harmless against all claims arising out of a claim for personal injury or death associated with such unintended use.

FOR YOUR SAFETY

Before undertaking any troubleshooting, maintenance or exploratory procedure, read carefully the **WARNINGS** and **CAUTION** notices.

CAUTION
RISK OF ELECTRICAL SHOCK
DO NOT OPEN

This equipment contains voltage hazardous to human life and safety, and is capable of inflicting personal injury.

If this instrument is to be powered from the AC line (mains) through an autotransformer, ensure the common connector is connected to the neutral (earth pole) of the power supply.

Before operating the unit, ensure the conductor (green wire) is connected to the ground (earth) conductor of the power outlet. Do not use a two-conductor extension cord or a three-prong/two-prong adapter. This will defeat the protective feature of the third conductor in the power cord.

CAUTION
SENSITIVE ELECTRONIC DEVICES
DO NOT SHIP OR STORE NEAR STRONG ELECTROSTATIC, ELECTROMAGNETIC, MAGNETIC OR RADIOACTIVE FIELDS

Maintenance and calibration procedures sometimes call for operation of the unit with power applied and protective covers removed. Read the procedures and heed warnings to avoid “live” circuit points.

Before operating this instrument:

1. Ensure the proper fuse is in place for the power source to operate.

2. Ensure all other devices connected to or in proximity to this instrument are properly grounded or connected to the protective third-wire earth ground.

If the instrument:

- fails to operate satisfactorily
- shows visible damage
- has been stored under unfavorable conditions
- has sustained stress

Do not operate until, performance is checked by qualified personnel.

Chapter 1

Chapter 2

Chapter 3

Appendix A

Chapter 1
Getting Started

What’s in this Manual?

This manual contains a module description, configuration and wiring information, and general programming information for the M1700 (P/N 407869) Double-Wide Breadboard M-Module. Complete specifications for the breadboard are provided in Appendix A.

The M1700 is intended to be installed on a carrier such as the C&H Technologies Model VX405C M-Module Carrier. Refer to the Carrier documentation for M-Module installation information.

This chapter contains general features, a block diagram description, configuration information and connector pinouts for the M1700.

Module Description

The M1700 is a double-wide breadboard M-Module. As a double-wide module it occupies two adjacent slots in the Carrier. However, it is scored down the middle and can easily be “snapped” in half; with all the active M-Module interface circuits on one of the two halves. See Figure 1-2.

Even though the M1700 breadboard is a double-wide, in an C&H Technologies Model VX405C Carrier, the module requires only one VXIbus logical address. When installed in the Carrier the module functions as a standard VXIbus device.

General Features

The M1700 provides:

- A8/D16 Register-based M-Module interface circuitry.
- A programmable BUSY Timer (for delaying interrupt acknowledgment of command completion).
- A 16-bit input/output (I/O) Register.
- Read/Write decoder lines
- Support for internal/external interrupts.

Block Diagram Description

Figure 1-1 shows a simplified block diagram of the M1700. In order to effectively program the module you must understand its operation at a block diagram level. The following paragraphs describe the major sections in the block diagram.

Module Control

This block contains all of the logic for the module including carrier interface, registers, interrupt control, the programmable busy timer, I/O lines, control lines, etc. All lines except the latched data output line (BC0 - BC15) connect directly to the U101 controller FPGA.

The I/O register at 14H in A24 memory space is realized on this Breadboard. Writing to this register automatically outputs the data on the BC0 - BC15 lines; similarly, reading this register automatically reads the data on the TR0 - TR15 lines. More I/O registers can be expanded by using the decoder lines and additional circuits.

ID EEPROM

The ID EEPROM holds sixty-four 16-bit words of M-Module ID data and VXI M-Module data.

Input and Output Circuit

The breadboard data input lines (TR0 - TR15) connect directly from the J103 pads to the U101 controller FPGA. The data output lines are latched by U107 and U108 (74ACTQ273SM) and are available on J103. Control lines (LAT*, DBEN*, CRST*, etc.) and the I/O Decoder lines (WP0 - WP7 for write decoder lines and RP0 - RP7 for read decoder lines) are available on J102. Refer to Chapter 2 for information on using these lines.

Power Supplies

The M1700 Breadboard provides filtered ±12VDC and +5VDC supplies for use on the breadboard. Do not exceed the maximum current specifications for the breadboard (see Appendix A). The active electronics provided on the module use only the +5VDC supply.

Each power supply is protected by a positive-temperature-coefficient thermistor. When maximum current through the thermistor is reached, its resistance increases significantly thereby limiting the current; the thermistor acts like a resettable fuse. Table 1-1 lists the power supplies, thermistor protection reference designator (silkscreened on PC board), and the maximum current.

Table 1-1. M1700 Power Supply Protection

	Thermistor Ref. Designator	Maximum Current
+5VDC Supply	F201	1000mA
+12VDC Supply	F200	200mA
-12VDC Supply	F202	200mA

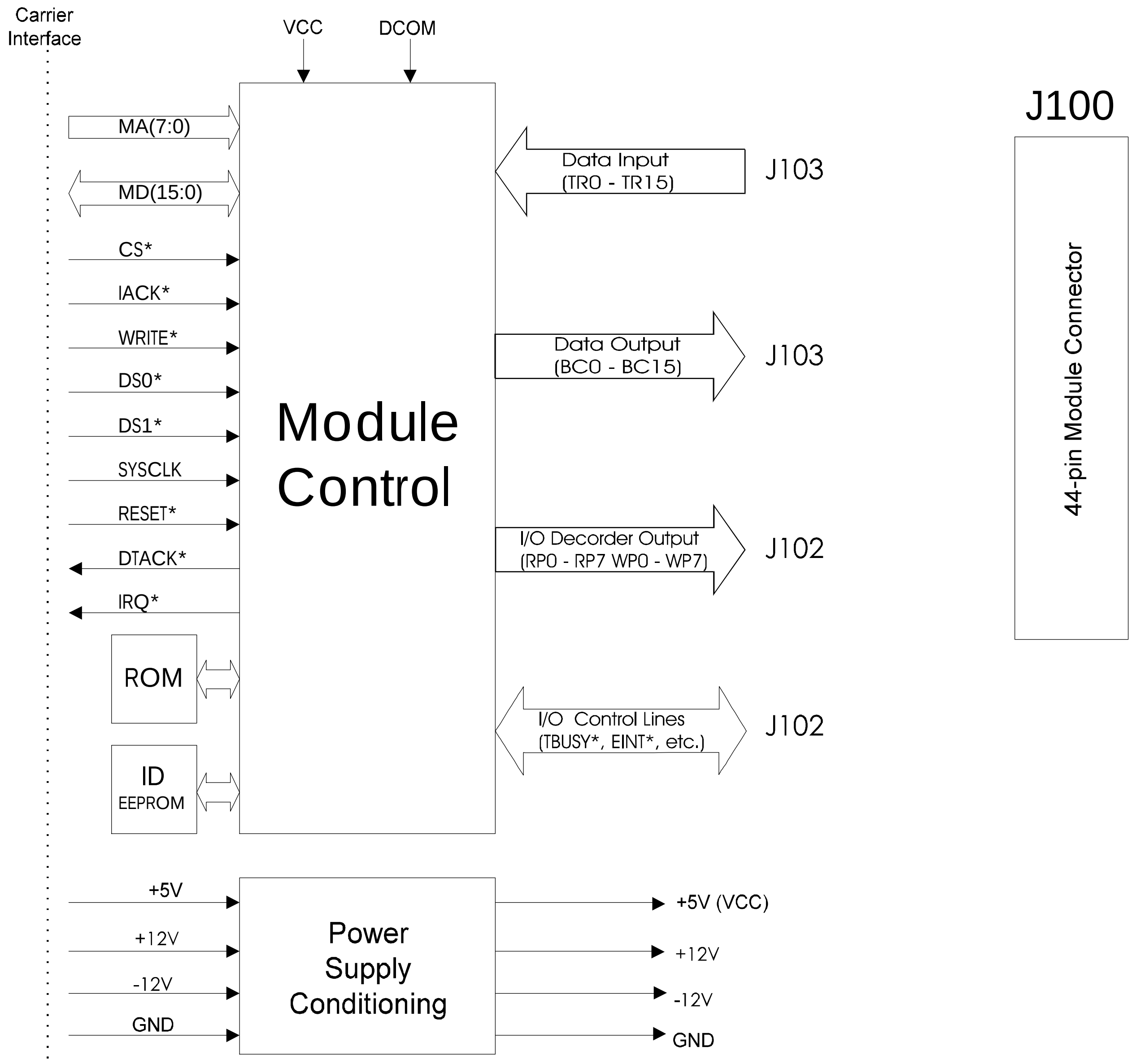


Figure 1-1. M1700 Block Diagram

Configuring the Module

This section provides general information to connect user wiring to the breadboard module and provides guidelines for mounting components on the module.

Note This section does not describe installing the breadboard module in a carrier. Since installation is dependent on the carrier used, you should refer to your carrier’s documentation for detailed installation instructions.

SHOCK HAZARD. Only qualified, service-trained personnel aware of the hazards involved should install, configure, or remove the M-Module. Disconnect all power sources from the mainframe, the terminal module and installed modules before installing or removing a module.

Caution **VOLTAGE/CURRENT. Pay careful attention to the limitation of maximum voltage/current and maximum power listed in Appendix A. Exceeding any limit or use outside the parameters specified may damage the module and/or carrier.**

Caution **STATIC ELECTRICITY. Static electricity is a major cause of component failure. To prevent damage to the electrical components on an M-Module or the carrier, observe anti-static techniques whenever installing, removing, or working on a carrier or M-Module.**

Connectors and Access Pads Description

There are several connectors and access pads on the M1700 Breadboard. Refer to Figure 1-2. Connector P100 provides the carrier interface to the active electronics on the module. Connector P101 provides access to the carrier interface; all pins on P101 are mapped to access pads on P102 (refer to P100, P101, and P102 Pin Assignment on page 8 for detailed pinout). Access pads J102 and J103 provide access to the buffered input and output lines, read and write decoder lines, busy line, and power supply connections (refer to J102 and J103 Pin Assignment on page 7 for detailed pinout). Connectors J100 and J104 provide connections for user field wiring. J100 directly maps pin-for-pin to access pads on J101; J104 maps pin-for-pin to access pads on J105. Figure 1-2 also shows the maximum component height above the board and maximum lead length below the board.

Note Observe the component height restrictions for Regions A and B on the Breadboard. Region A: 0.21 in (5.3mm); Region B: 0.42 in (10.5mm).

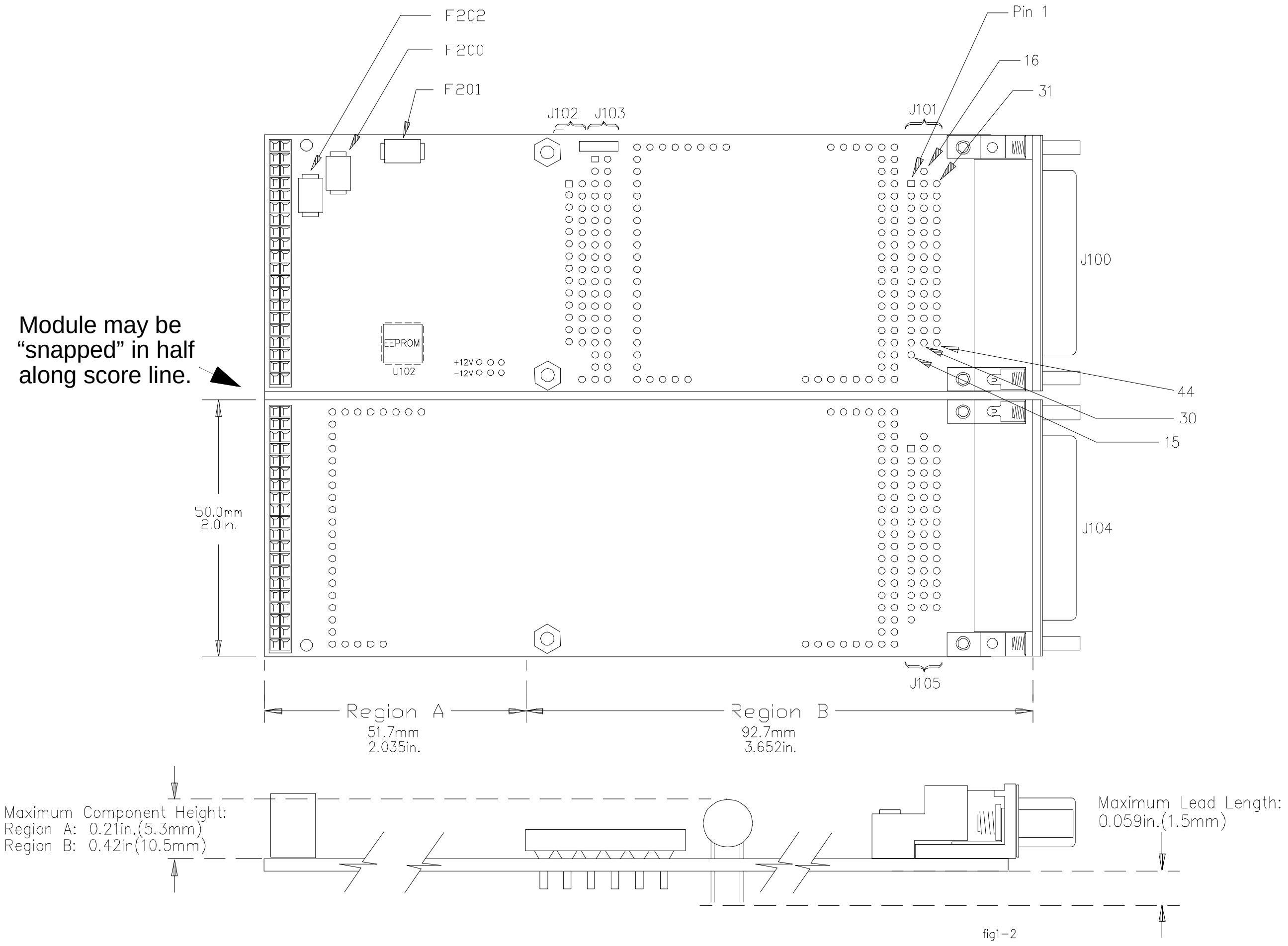

Module may be “snapped” in half along score line.

Figure 1-2. M1700 Connectors Layout

J100, J101, J104, and J105 Pin Assignment

J100, J101, J104, and J105 are for user field wiring. J100 maps pin-for-pin to J101 (see Figure 1-2) and J104 maps pin-for-pin maps to J105. Figure 1-3 shows the 44-pin field wiring connector pinout. Figure 1-6 shows how to assemble the field wiring connector and hood assembly.

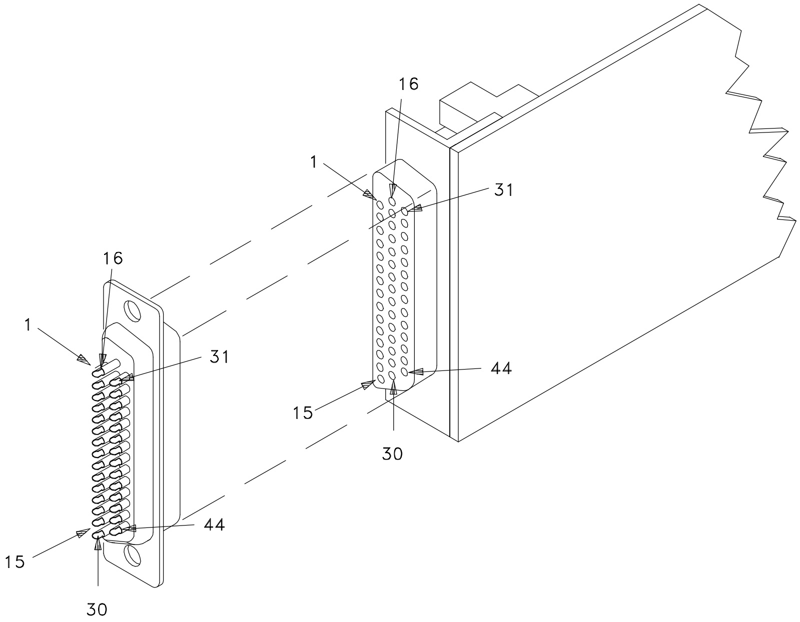
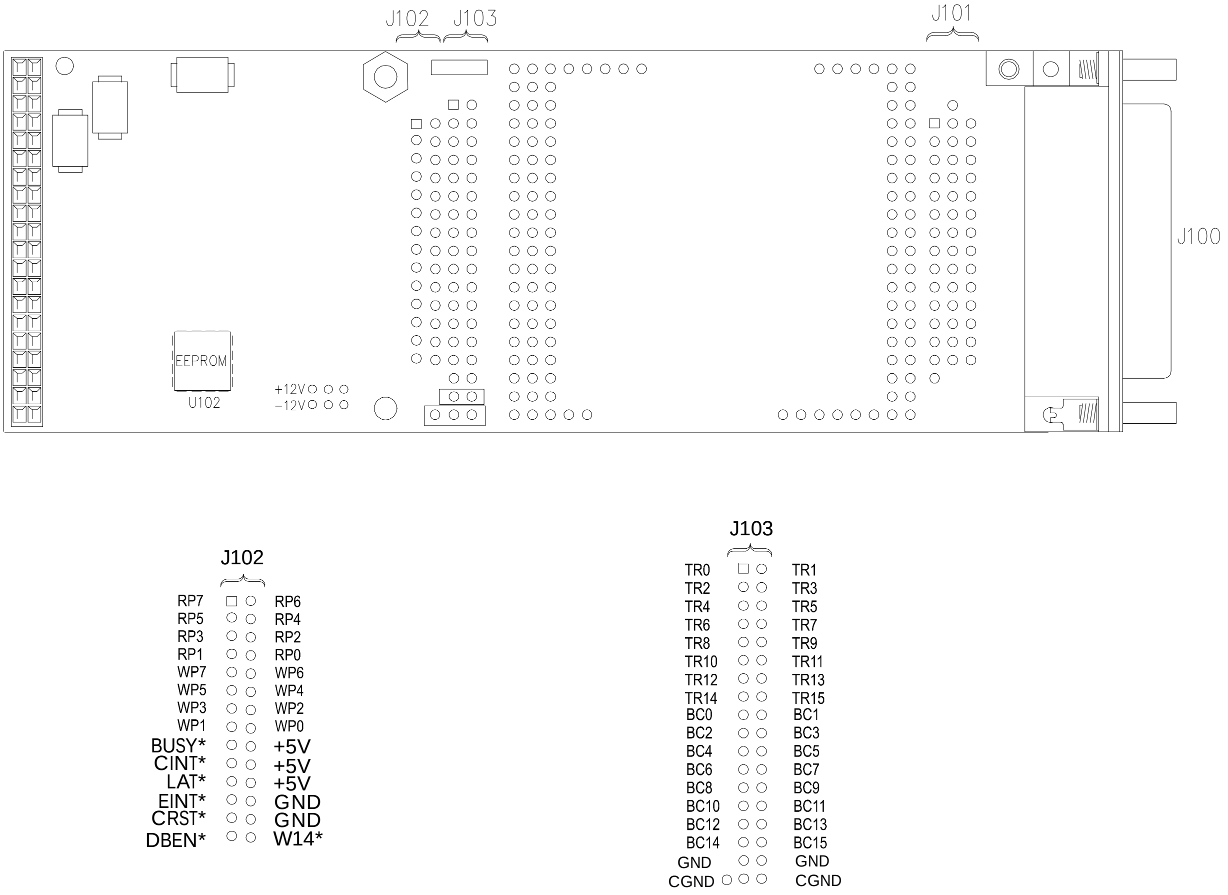


Figure 1-3. 44-Pin Field Wiring Connector

J102 and J103 Pin Assignment

Figure 1-4 shows the pinout for the J102 and J103 pads. Notice that the bottom row of pads on J103 (CGND) are connected to chassis ground through the carrier to the VXIbus mainframe. The ±12VDC pads are located near the bottom of J102.



J102	
RP7	RP6
RP5	RP4
RP3	RP2
RP1	RP0
WP7	WP6
WP5	WP4
WP3	WP2
WP1	WP0
BUSY*	+5V
CINT*	+5V
LAT*	+5V
EINT*	GND
CRST*	GND
DBEN*	W14*

J103	
TR0	TR1
TR2	TR3
TR4	TR5
TR6	TR7
TR8	TR9
TR10	TR11
TR12	TR13
TR14	TR15
BC0	BC1
BC2	BC3
BC4	BC5
BC6	BC7
BC8	BC9
BC10	BC11
BC12	BC13
BC14	BC15
GND	GND
CGND	CGND

Figure 1-4. J102, J103 Connector Pinouts

TR0 - TR15: Data Input Lines (register address 14_h in A24 Memory).

BC0 - BC15: Data Output Lines (register address 14_h in A24 Memory).

WP0 - WP7: Write Decoder Lines (address 20_h-$2E_h$ in A24 Memory).

RP0 - RP7: Read Decoder Lines (address 20_h-$2E_h$ in A24 Memory).

BUSY*: Delay Timer Output. (Delay Time Reg. Addr. 12_h in A24) output from breadboard.

CINT*: Clear Interrupt. A negative pulse is asserted during interrupt acknowledge if external interrupt request is pending, output from breadboard.

EINT*: User Supplied External Interrupt Request. Falling edge causes interrupt to host controller if enabled.

LAT*: Latch signal for write cycle, output from breadboard.

CRST*: Reset, low means breadboard being reset, output from breadboard.

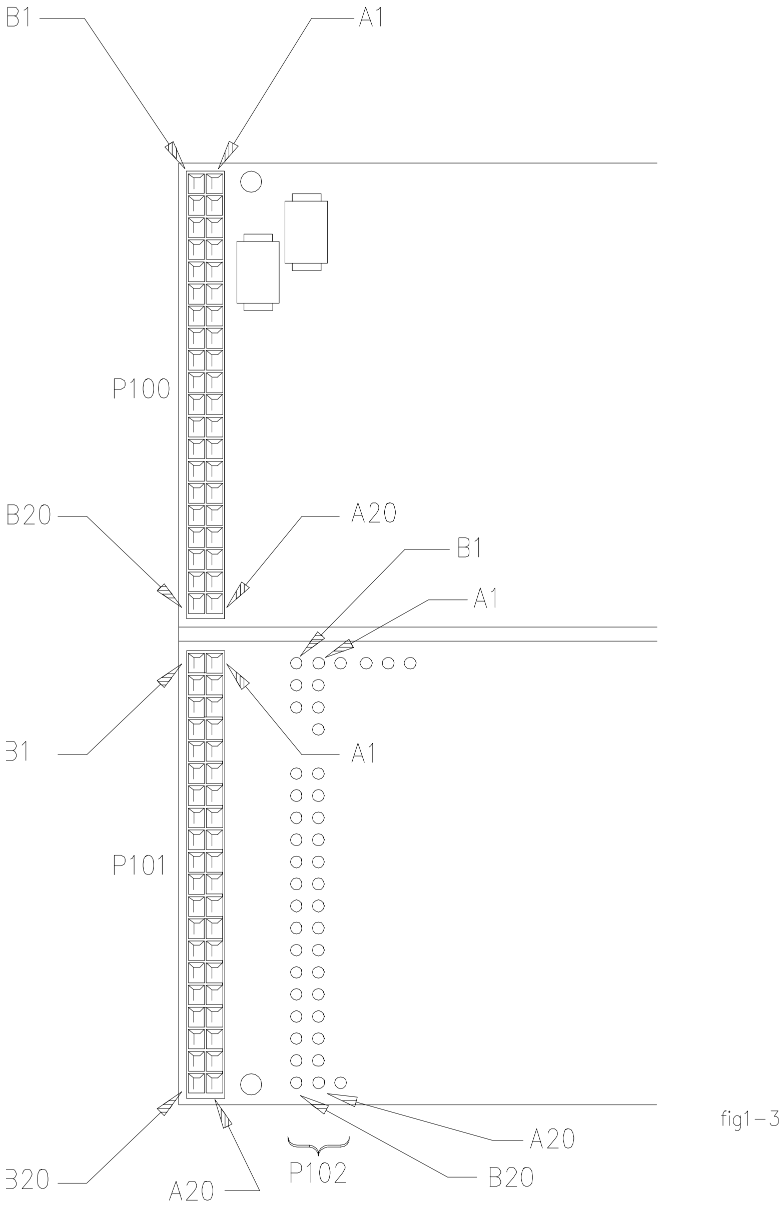
DBEN*: Data Bus Enable, low to enable data bus, output from breadboard.

W14*: Output Register Latch Signal of 14H in A24 memory space, rising edge latches data into external device, output from breadboard.

CGND: Chassis Ground

P100, P101, and P102 Pin Assignment

Figure 1-5 shows the connector pinouts for P100, P101, and P102. Pin 101 connects to and maps pin-for-pin to P102. The pin names are described on the next page.



Pin	Col A	Col B
1	CS*	Ground
2	MA01	+5V
3	MA02	+12V
4	MA03	-12V
5	MA04	Ground
6	MA05	DREQ*
7	MA06	DACK*
8	MA07	Ground
9	MD08	MD00
10	MD09	MD01
11	MD10	MD02
12	MD11	MD03
13	MD12	MD04
14	MD13	MD05
15	MD14	MD06
16	MD15	MD07
17	DS1*	DS0*
18	DTACK *	WRITE*
19	IACK*	IRQ*
20	RESET*	SYSCLK

Figure 1-5. P100, P101, P102 Connectors

MA01 to MA07: M-Module Address Bus, Input to M-Module.

CS*: Card Select Signal, Input to M-Module. Low when the M-Module is being accessed.

MD00 to MD15: M-Module Data Bus, Input / Output.

DACK*: Low when carrier is performing a DMA transfer, Input to M-Module.

DREQ*: Low to requests a DMA transfer, Output from M-Module.

DS0* - DS1*: Data Strobe Signal, Input to M-Module
00 = 16 bit data transfer
01 = D15 - D08 transfer
10 = D7 - D0 transfer
11 = no data transfer.

DTACK*: Data Acknowledge Signal, Output from M-Module. Low to acknowledge data transfer and terminate access.

IACK*: Interrupt Acknowledge Signal, Input to M-Module. Low when interrupt is being acknowledged.

IRQ*: Interrupt Request Signal, Output from M-Module. Low to request an interrupt.

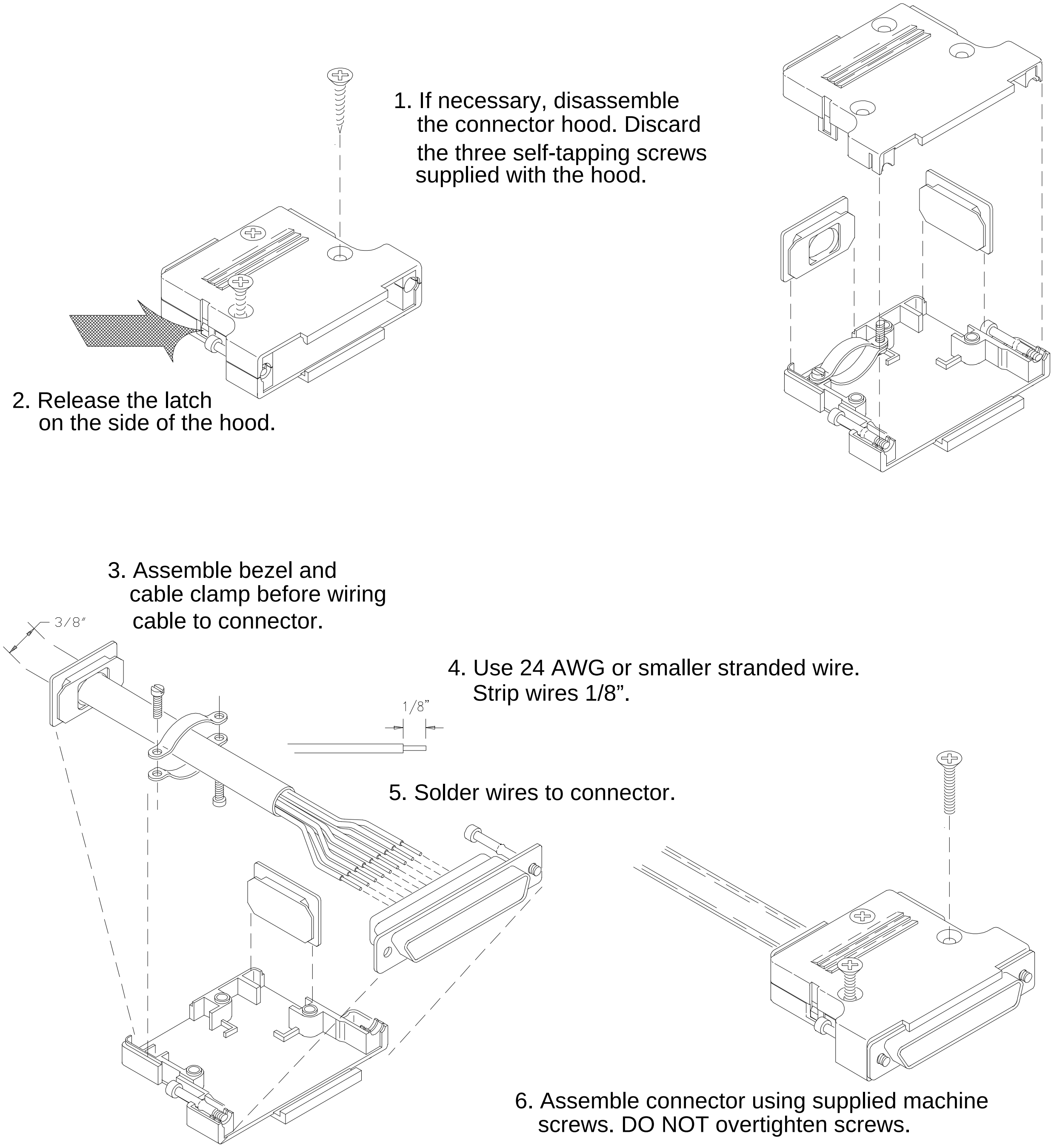
RESET*: System Reset Signal, Input to M-Module. Low to reset the module.

SYSCLK: 16MHz System Clock, Input to M-Module.

WRITE*: Write Signal, Input to M-Module. Low during writing cycle, high during reading cycle.

Assembling the Field Wiring Connector

Each M1700 module includes two 44-pin connector and hood kits. You must supply your own cable. The drawing below shows how to connect wiring and assemble the connector and hood.



1. If necessary, disassemble the connector hood. Discard the three self-tapping screws supplied with the hood.

2. Release the latch on the side of the hood.

3. Assemble bezel and cable clamp before wiring cable to connector.

4. Use 24 AWG or smaller stranded wire. Strip wires 1/8”.

5. Solder wires to connector.

6. Assemble connector using supplied machine screws. DO NOT overtighten screws.

Figure 1-6. Assembling the Field Wiring Connector and Hood

Chapter 2
Using the Breadboard

What’s in This Chapter

This chapter provides general operating information for the M1700 Double-wide Breadboard. It also provides a simple application example. The chapter contents include:

Registers

Table 2-1 lists the registers available on the M1700 Breadboard (Chapter 3 provides a detailed description of all of the registers on the M1700 .) This section provides information on using the Input/Output Register and the Read/Write Decoder Registers.

Table 2-1. M1700 Registers

Register Name	Register Address	Register Description
VXI ID Register	00_h in VXI A16 Memory	Provides Device Class and Manufacturer ID.
VXI Device Type Register	02_h in VXI A16 Memory	Provides Model Code for M-Module.
VXI Status/Control Register	04_h in VXI A16 Memory	Controls Reset and Provides module Status information.
A24 Offset Register	06_h in VXI A16 Memory	Provides A24 Offset Base Address.
Interrupt Selection Register	20_h in VXI A16 Memory	Selects VXI Interrupt Line.
Status Register	00_h in VXI A24 Memory	Selects Interrupt Source.
Control Register	02_h in VXI A24 Memory	Controls Reset and Enables/Disables Interrupt.
Interrupt Register	04_h in VXI A24 Memory	Selects Type of Interrupt.
Delay Timer (BUSY) Register	12_h in VXI A24 Memory	Sets Delay Value for Command completion interrupt.
Input/Output Register	14_h in VXI A24 Memory	Data Input from TR0 - TR15 Lines. Data Output to BC0 - BC15 Lines.
Read/Write Decoder Registers	20_h - $2E_h$ in VXI A24 Memory	Be used to select the latches or buffers supplied by users.

Writing Data to Output Registers

To output data over the data bus lines (BC0 - BC15), write the data to the I/O register at 14_h in A24 Memory. Figure 2-1 shows the timing of writing data to the I/O register. You only have access to the DBEN*, LAT*, W14*, and BUSY* control lines; the other control line waveforms (SYSCLK, DS0*, DS1*, CS*, DTACK*, and MA0-MA7) are provided for reference information only. Note:

- W14* has the same timing as LAT* when writing to 14H in A24 memory space.
- IRQ* is asserted at the end of the BUSY cycle (see "Using the Interrupt" on page 6). The BUSY* line pulse lasts 13mS if the module default is used. You can change the BUSY time by writing a value to the Delay Timer Register at address 12_h. Refer to Chapter 3 for details.
- MD15-MD0 are latched to BC15-BC0 at the rising edge of W14*.

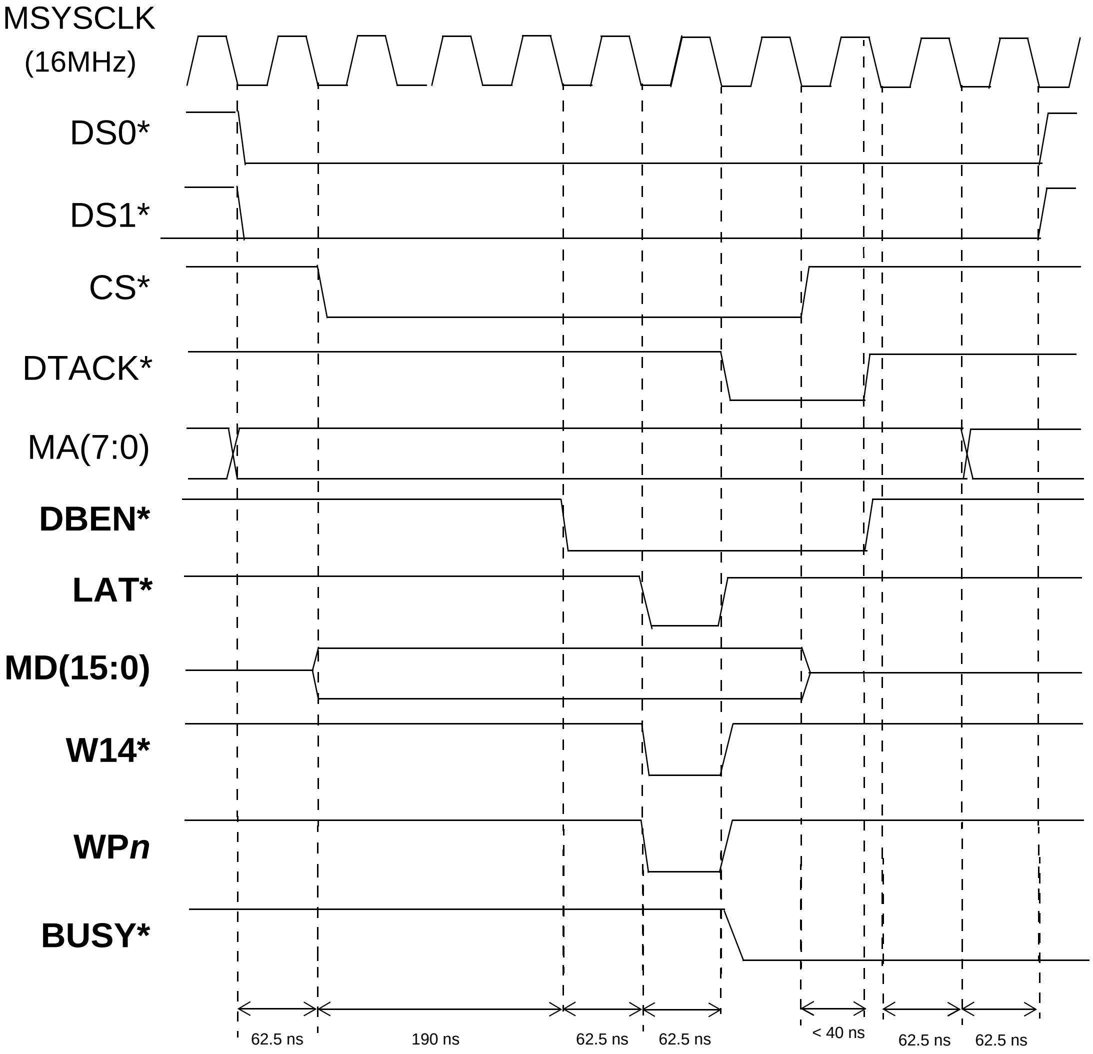


Figure 2-1. Timing of Writing to I/O Register

Using the Write Decoder Lines

You can use the eight Write Decoder lines (WP0 - WP7) as select lines of latcher to expand more output registers. Table 2-2 lists the Read/Write Decoder Register Addresses and the corresponding WP and RP lines.

Table 2-2. Read/Write Decoder Lines and Register Addresses

WP / RP Line	Register Address		WP / RP Line	Register Address
WP0/ RP0	20_h in A24 Memory		**WP4 / RP4**	28_h in A24 Memory
WP1 / RP1	22_h in A24 Memory		**WP5 / RP5**	$2A_h$ in A24 Memory
WP2 / RP2	24_h in A24 Memory		**WP6 / RP6**	$2C_h$ in A24 Memory
WP3 / RP3	26_h in A24 Memory		**WP7 / RP7**	$2E_h$ in A24 Memory

Figure 2-1 shows the timing required to write output data using one of the Read/Write Decoder Registers and the corresponding WP Line. You only have access to the DBEN*, LAT*, WP*n*, and BUSY control lines; the other control line waveforms (SYSCLK, DS0*, etc.) are provided for reference information only.

- WP*n* has the same timing as LAT* when writing to the corresponding register address.
- There is not BUSY when not writing to 14H in A24 memory space.
- MD15-MD0 are latched to latchers at the rising edge of WP*n*.

Figure 2-2 shows an example of using the WP0 Decoder Line to serve as a device latching signal.

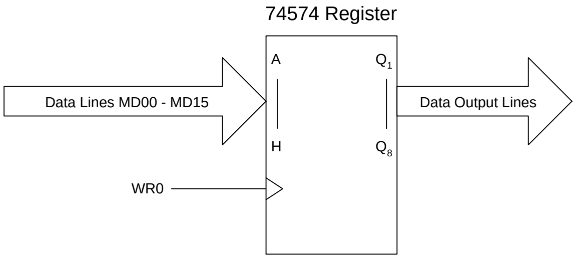


Figure 2-2. Using the Write Decoder Line WP0

To use the expanded register, write a data value to the appropriate Read/Write Decoder Register. For example, to use the circuit in Figure 2-2, you would write a 16-bit data value to the WP0 register (address 20_h in A24 memory).

Reading Data From Input Registers

To read data from the TR0 - TR15 Data Input lines, read the data from the I/O register at 14_h in A24 memory space. Figure 2-3 illustrates the timing required for reading data from this register on the M1700 Breadboard. You only have access to the DBEN* and LAT* control lines; the other control line waveforms are provided for reference information only.

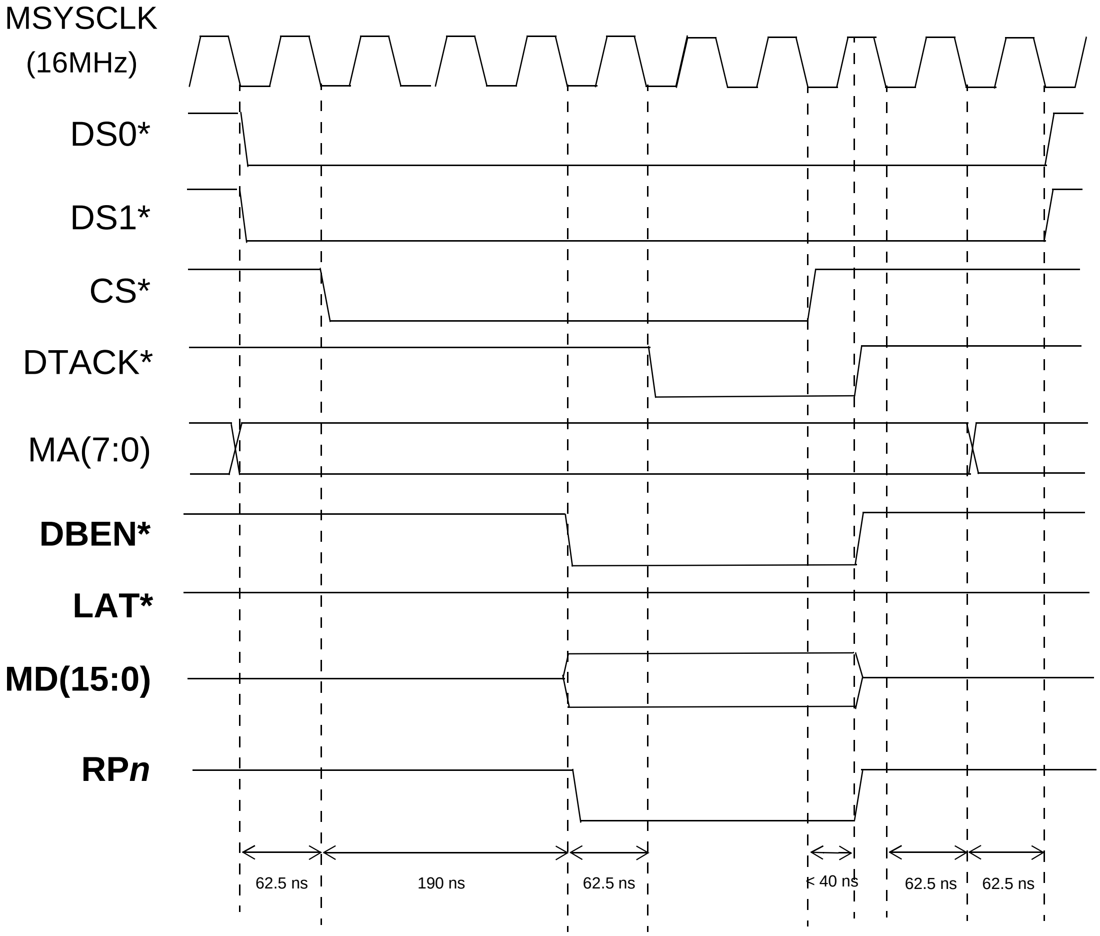


Figure 2-3. Timing of Reading Data from Registers

Using the Read Decoder Lines

Expanding input registers with read decoder lines RP0-RP7 is similar to expanding output registers (see Figure 2-2 on Page 2-3), except that a buffer (74245) is used as an input buffer, RPn is used as buffer enable signal, data direction is to MD15-MD0. The RP*n* lines can be used as select lines to specify which device the data is read from. Figure 2-3 also shows the timing of the RP*n* pulse. Table 2-3 lists the Read/Write Decoder Register Addresses and the corresponding WP and RP lines.

Table 2-3. Read/Write Decoder Lines and Register Addresses

WP / RP Line	Register Address		WP / RP Line	Register Address
WP0/ RP0	20_h in A24 Memory		**WP4 / RP4**	28_h in A24 Memory
WP1 / RP1	22_h in A24 Memory		**WP5 / RP5**	$2A_h$ in A24 Memory
WP2 / RP2	24_h in A24 Memory		**WP6 / RP6**	$2C_h$ in A24 Memory
WP3 / RP3	26_h in A24 Memory		**WP7 / RP7**	$2E_h$ in A24 Memory

Using the Interrupt

The Breadboard has a read only Interrupt Register located at address 04_h in A24 Memory. The contents of this register is used as an interrupt vector to determine which interrupt source caused the interrupt.

If IACK* (Type C interrupt) is present, the content of the Interrupt Register will be sent to the data bus and the IRQ* line will be released. Reading the Interrupt Register will clear the interrupt request signal IRQ* to support Type A interrupt.

Bits 2 through bit 15 of this register are not used. Bit 1 indicates an external interrupt existing if the content of bit 1 is “1”. See *Chapter 3, Register Description* for more information of this Interrupt Register.

IRQ* will be valid whenever the external interrupt occurs.

Figure 2-4 shows the timing requested for an interrupt.

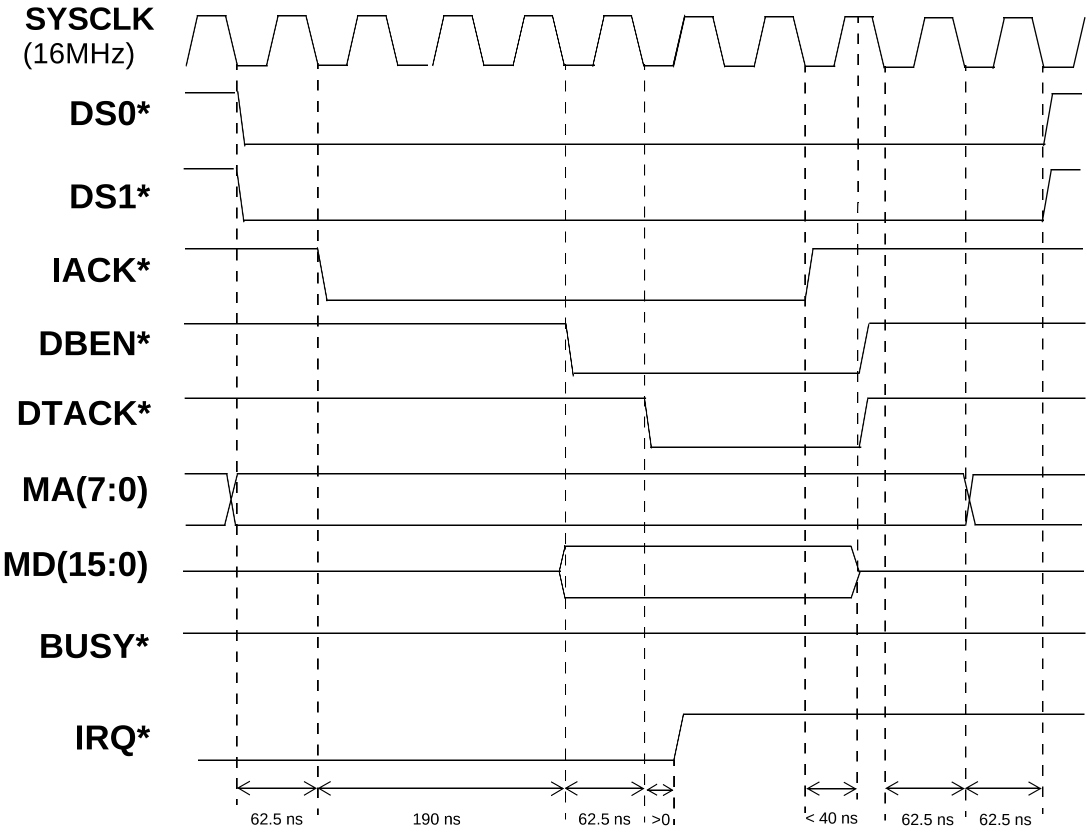


Figure 2-4. Timing for Interrupt Acknowledge Cycle

Program Example

The following program example demonstrates how to access the registers on the M1700 . The example uses the VXIplug&play register reads and writes.

The example programs were developed with the ANSI C language using the VISA extensions. The programs were written and tested in Microsoft® Visual C++ but should compile under any standard ANSI C compiler.

To run the programs you must have the VISA extensions, and a GPIB module installed and properly configured in your PC. A GPIB Slot Zero Module is required for the first program. The M1700 must be installed in an C&H Technologies Model VX405C A Carrier or equivalent for access to the A16 VXI registers.

The example programs reset the M1700 Breadboard M-Module. It then:

- Reads the ID, Device Type, Status, and A24 Memory Offset Registers.
- Writes a value to the I/O Register to output data on BC0 - BC15,
- Writes a value to the WP0 Register to output data on BC0 - BC15,
- Reads a value from the I/O Register to input data on TR0 - TR15,
- Reads a value from the RP0 Register to input data on TR0 - TR15.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

ViSession viRM,m_mod;
int main()
{

unsigned short id_reg,dt_reg ;              /* ID & Device Type Registers */
unsigned short stat_reg, a24_offset ;       /* Status & A24 Offset Reg */
short value;                                /* register variable */

ViStatus errStatus;                         /*Status from each VISA call*/

    /* Open the default resource manager  */
errStatus = viOpenDefaultRM ( &viRM);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOpenDefaultRM() returned 0x%x\n",errStatus);
   return errStatus;}

    /* Open the M-Module instrument session  */
errStatus = viOpen(viRM,"GPIB-VXI0::8",VI_NULL,VI_NULL,&m_mod);
if (VI_SUCCESS > errStatus){
   printf("ERROR: viOpen() returned 0x%x\n",errStatus);
   return errStatus;}

    /* read  and print the module's ID Register  */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x00,&id_reg);
```

```
if (VI_SUCCESS > errStatus){
    printf("ERROR: viIn16() returned 0x%x\n",errStatus);
    return errStatus;}
printf("ID register = 0x%4X\n", id_reg);

      /* read  and print the module's Device Type Register   */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x02,&dt_reg);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viIn16() returned 0x%x\n",errStatus);
    return errStatus;}
printf("Device Type register = 0x%4X\n", dt_reg);

      /* read and print the module's Status Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x04,&stat_reg);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viIn16() returned 0x%x\n",errStatus);
    return errStatus;}
printf("Status register = 0x%hx\n", stat_reg);

      /* read and print the module's A24 Offset Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x06,&a24_offset);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viOpen() returned 0x%x\n",errStatus);
    return errStatus;}
printf("A24 Offset register value = 0x%hx\n", a24_offset);

      /* Write a value of AA to I/O Register, Addr 0x14 */
errStatus = viOut16(m_mod,VI_A24_SPACE,0x14,0xAA);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viOut16() returned 0x%x\n",errStatus);
    return errStatus;}

      /* Write a value of FF to WP0 Register, Addr 0x20 */
errStatus = viOut16(m_mod,VI_A24_SPACE,0x20,0xFF);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viOut16() returned 0x%x\n",errStatus);
    return errStatus;}

      /* Read the value from I/O Register, Addr 0x14 */
errStatus = viIn16(m_mod,VI_A24_SPACE,0x14,&value);
if (VI_SUCCESS > errStatus){
    printf("ERROR: viOut16() returned 0x%x\n",errStatus);
    return errStatus;}
printf("0x14 register value = 0x%hx\n", value);
```

```
    /* Read the value from WP0 Register, Addr 0x20 */
errStatus = viIn16(m_mod,VI_A24_SPACE,0x20,&value);
if (VI_SUCCESS > errStatus){
   printf(“ERROR: viOut16() returned 0x%x\n”,errStatus);
   return errStatus;}
printf(“0x20 register value = 0x%hx\n”, value);

    /* Close the M-Module Instrument Session */
errStatus = viClose (m_mod);
if (VI_SUCCESS > errStatus) {
   printf(“ERROR: viClose() returned 0x%x\n”,errStatus);
   return 0;}

    /* Close the Resource Manager Session */
errStatus = viClose (viRM);
if (VI_SUCCESS > errStatus) {
   printf(“ERROR: viClose() returned 0x%x\n”,errStatus);
   return 0;}

return VI_SUCCESS;

}
```

Resetting the Module

There are two ways to reset the module and your own external circuitry. The CRST* signal line is available on J102 which is available to reset your circuitry (active low).

Hardware Reset

Whenever power on the system, the M1700 breadboard module will also be reset. This pulses the CRST* line low.

Software Reset

You can also write a “1” to bit 0 of the Control Register (address 04_h in A16 memory, refer to Chapter 3 for details). This will reset the M-Module and pulse the CRST* line. You must write a “0” to bit 0 of the Control Register after the reset is finished.

Using the Power Supplies

Each power supply available for use by external circuits is protected by a positive-temperature-coefficient thermistor. When maximum current through the thermistor is reached, it heats up, and its resistance increases significantly thereby limiting the current; the thermistor acts like a resettable fuse. Table 2-4 lists the power supplies, thermistor protection reference designator (silkscreened on PC board), and the maximum current.

Table 2-4. M1700 Power Supply Protection

	Thermistor Ref. Designator	Maximum Current	Connector	Application
+5VDC Supply	F201	1000mA	J102	Main power source for all systems. Used for supplying power to logic device.
+12VDC Supply	F200	200mA	Near J102	General purpose power for switching power convertors, analog devices and disc drives.
-12VDC Supply	F202	200mA	Near J102	General purpose power for analog devices and disc drives. Not recommended for power convertors.

Application Example

This section contains a simplified example application using the breadboard module as an 8-channel Form-C Relay module. See Figure 2-6. You must supply the relay driver IC, flyback protection, etc. These have been ommitted from the drawing for simplification.

Relay Selection

The I/O Register is used as the interface. To open one specific relay, you need to WRITE a "0" to the bit of the I/O Register which corresponds to that relay. To close a channel, you need to write a "1" to the corresponding I/O Register bit.

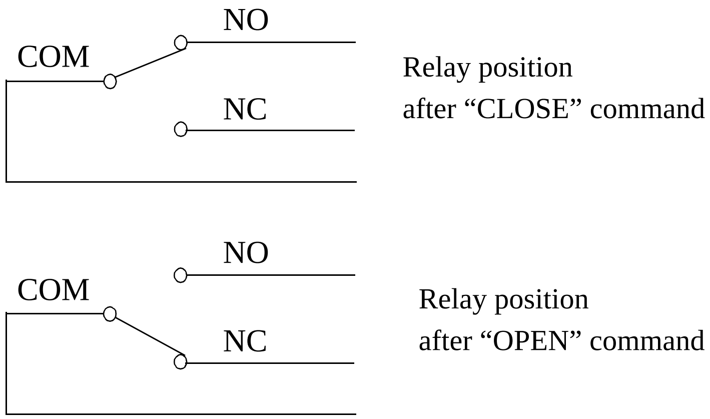
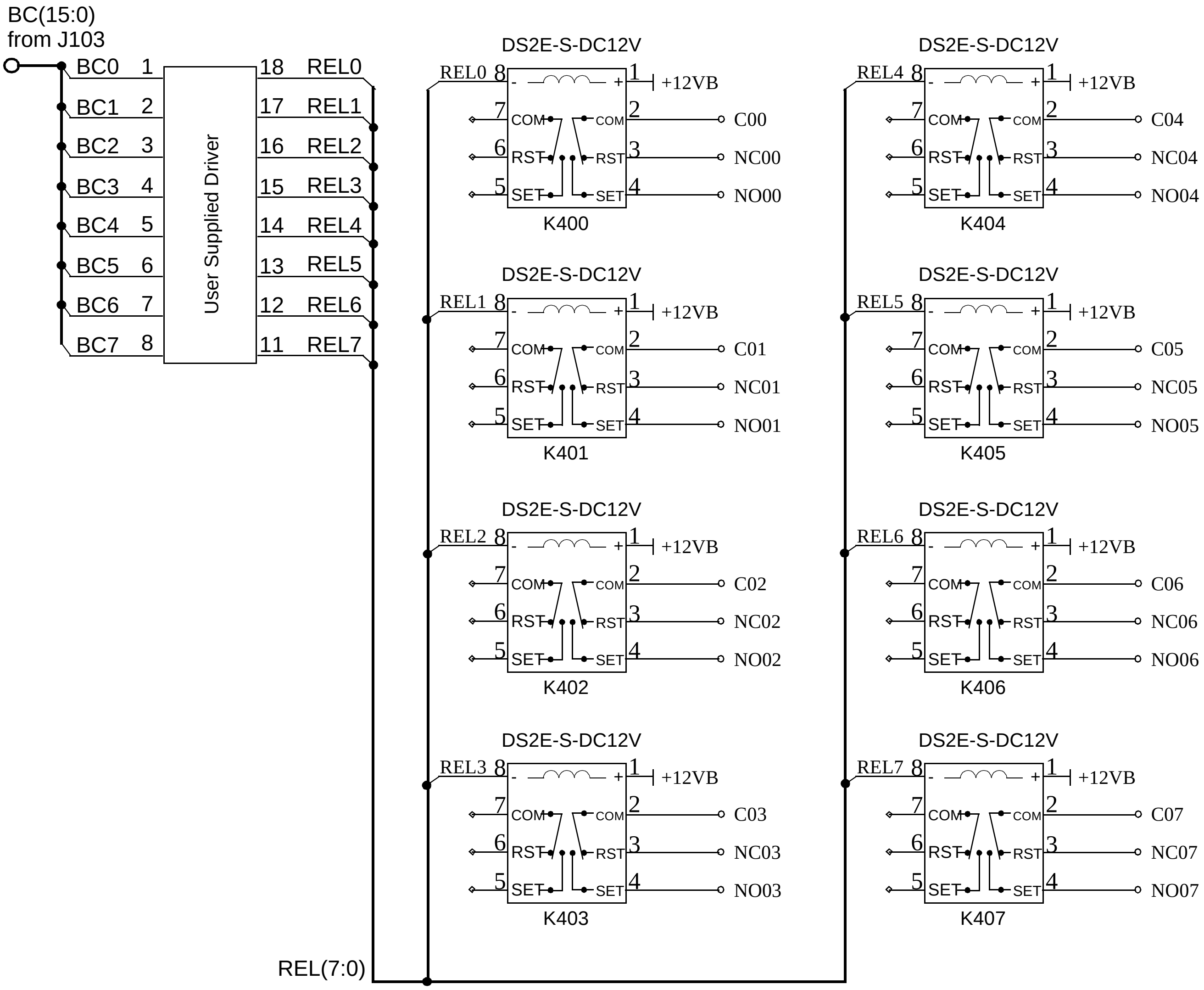


Figure 2-5. Write a "1" or "0" to the Register Bit to Close/Open the Relay

The status of all relays must be specified simultaneously. To change just one relay, it is necessary to change the corresponding bit and send the entire pattern again.



The 16-bit I/O Register of M1700 is used
to control and readback 8 Form-C Relays.

Figure 2-6. Using the Breadboard as an 8 channel Form C Relay Module

Chapter 3
Register Descriptions

About This Chapter

This chapter contains register addressing information; it also provides the register description of M1700 Double-Wide Breadboard M-Module. Chapter contents include:

Note Most of the descriptions in this chapter assume the M1700 is installed in an C&H Technologies Model VX405C Carrier. The Carrier provides access to the VXI registers.

Register Addressing in the VXIbus Environment

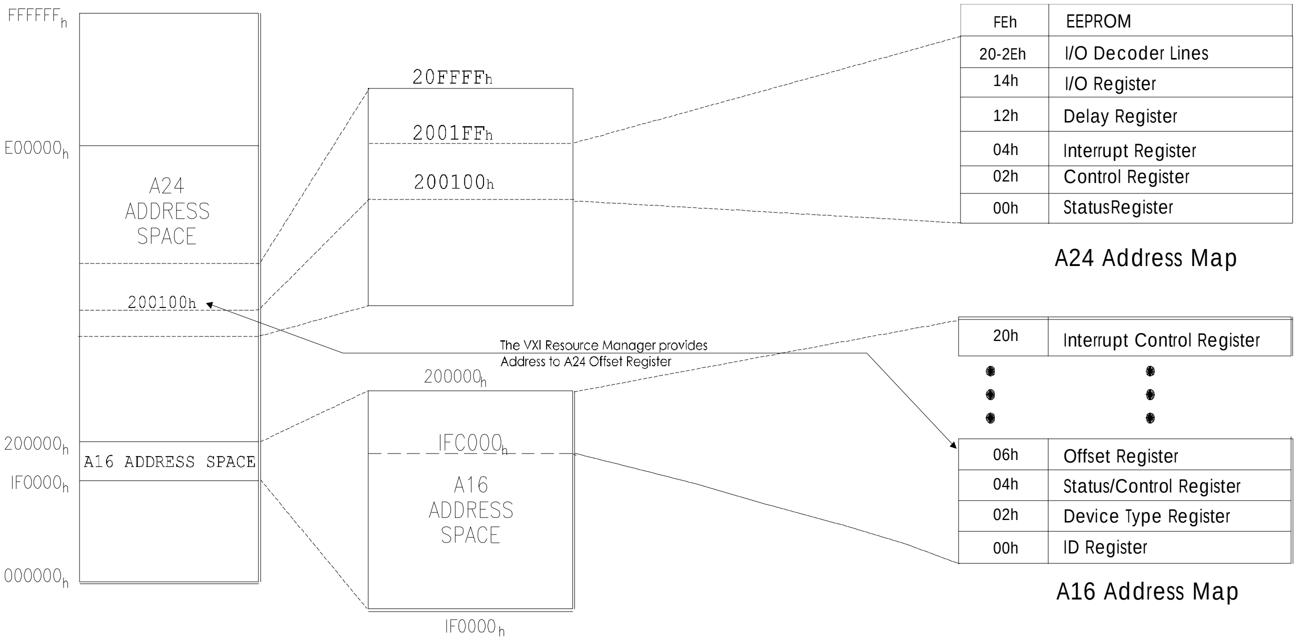
Logical Address

Each module in a VXIbus (VXI) system, whether VXI or M-Module, must have a unique logical address. The C&H Technologies Model VX405C Carrier or equivalent provides a logical address for each installed M-Module. Refer to the VX405C User Manual for details (if you are using a different carrier, refer to that carrier's documentation for register-based addressing information).

A16/A24 Memory Mapping

The VXI Specification allows for only 64 bytes of address space in A16 memory. However, the M-Module Specification defines 256 bytes of address space. To resolve this conflict, the C&H Technologies Model VX405C Carrier provides two memory segments for each installed M-Module. The first is in the VXI A16 memory space and contains the standard VXI registers. The second memory segment is in the VXI A24 memory space and contains all other M-Module registers (these registers are described starting on Page 3-10). Figure 3-1 shows the A16/A24 mapping for a typical M-Module.

Note The M-Module's ID word (from the ID EEPROM) is mapped into the VXI Manufacturer ID Register at address 00_h and the M-Module's VXI Device Type word is mapped into the VXI Device Type Register at address 02_h



* Base Address = $1FC000_{16}$ + (Logical Address * $64)_{16}$ or
= $2,080,768_{10}$ + (Logical Address*$64)_{10}$

A16 Register Address = Base Address + Register Offset

For M-Modules, the Register Address is Computed as:
Base Address = Value in Offset Register
Register Address = Base Address Combines the Register Offset
(A24 address space)

For Example, in above case, the I/O Register Address is:
I/O Register Address = 200100_{16} + 14_{16} = 200114_{16} or
= $2,097,408_{10}$ + 20_{10} = $2,907,428_{10}$

Figure 3-1. A16/A24 Memory Mapping

To access a register in A16 memory, you must specify a hexadecimal or decimal register address. This address consists of a base address plus a register offset. The A16 base address depends on whether or not you are using a GPIB Slot Zero, such as a Racal 1260-00C.

A16 Address Space Inside the Command Module

When using a GPIB Slot Zero Command Module, the base address is computed as:

$1FC000_h + (LADDR_h \cdot 40_h)$
or (decimal)
$2,080,768 + (LADDR \cdot 64)$

Where:
$1FC000_h$ (2,080,768) is the A16 starting address
LADDR is the module’s logical address

40_h (64) is the number of address bytes allocated per module

For example, if the M-Module has a logical address of 78_h (120) the A16 base address becomes:

$1FC000_h + (78_h \cdot 40_h) = 1FC000_h + 1E00_h = 1FDE00_h$
or (decimal)
$2,080,768 + (120 \cdot 64) = 2,080,768 + 7680 = 2,088,448$

A16 Address Space Outside the Command Module

When a GPIB Slot Zero Command Module is <u>not</u> part of your system, the base address is computed as:

$C000_h + (LADDR_h \cdot 40_h)$
or (decimal)
$49,152 + (LADDR \cdot 64)$

Where:
$C000_h$ (49,152) is the A16 starting address
LADDR is the module's logical address
40_h (64) is the number of address bytes allocated per module

For example, if the M-Module has a logical address of 78_h (120) the A16 base address becomes:

$C000_h + (78_h \cdot 40_h) = C000_h + 1E00_h = DE00_h$
or (decimal)
$49,152 + (120 \cdot 64) = 49,152 + 7680 = 56,832$

Addressing A16 Registers

As shown in Figure 3-1, VXI registers for an M-Module are mapped into A16 address space by the M-Module Carrier. To access one of these registers, add the A16 base address to the register offset. For example, an M-Module's VXI Status/Control Register has an offset of 04_h. To access this register (assuming the system does not have a GPIB Slot Zero Module), use the register address:

$1FDE00_h + 04_h = 1FDE04_h$
or (decimal)
$2,088,488 + 4 = 2,088,452$

Addressing A24 Registers

As shown in Figure 3-1, most of the registers for an M-Module are mapped into A24 address space. To access one of these registers:

1. Obtain the A24 base address by reading the VXI Offset Register (06_h) in A16 memory.

2. Add the A24 base address to the register offset (see Table 3-2).

For example, if the A24 base address is 200100_h, to access the Output Register (10_h):

$200100_h + 10_h = 200110_h$

or (decimal)
2,097,408 + 16 = 2,097,424

Program Example

The following sample program demonstrates how to write to and read from registers on the M1700 . The program was developed with the ANSI C language using the VISA extensions. The program was written and tested in Microsoft Visual C++ but should compile under any standard ANSI C compiler.

To run the program you must have the VISA extensions, installed and properly configured in your PC. A GPIB Slot Zero Module provides direct access to the VXI backplane. The M1700 must be installed in an M-Module Carrier for access to the A16 VXI Registers.

```
#include <visa.h>
#include <stdio.h>
#include <stdlib.h>

ViSession viRM,m_mod

int main()
{
   ViStatus errStatus;                          /*Status from each VISA call*/
                                                 session*/

/* Open the default resource manager  */
   errStatus = viOpenDefaultRM (&viRM);
   if(VI_SUCCESS > errStatus){
      printf(“ERROR: viOpenDefaultRM() returned 0x%x\n”,errStatus);
      return errStatus;}

/* Open the M-Module instrument session  */
errStatus = viOpen(viRM,INSTR_ADDR, VI_NULL,VI_NULL,&m_mod);
   if(VI_SUCCESS > errStatus){
      printf(“ERROR: viOpen() returned 0x%x\n”,errStatus);
      return errStatus;}

   /* read  and print the module's ID Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x00,&id_reg);
   if (VI_SUCCESS > errStatus){
      printf(“ERROR: viIn16() returned 0x%x\n”,errStatus);
      return errStatus;}
      printf("ID register = 0x%hx\n”, id_reg);

   /* read  and print the module's Device Type Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x02,&dt_reg);
   if (VI_SUCCESS > errStatus){
      printf(“ERROR: viIn16() returned 0x%x\n”,errStatus);
      return errStatus;}
   printf(“Device Type register = 0x%hx\n", dt_reg);

   /* read and print the module's Status Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x04,&stat_reg);
   if (VI_SUCCESS > errStatus){
        printf(“ERROR: viIn16() returned 0x%x\n”,errStatus);
        return errStatus;}
    printf("Status register = 0x%hx\n", stat_reg);

/* read and print the module's A24 Offset Register */
errStatus = viIn16(m_mod,VI_A16_SPACE,0x06,&a24_offset);
    if(VI_SUCCESS > errStatus){
    printf(“ERROR: viOpen() returned 0x%x\n”,errStatus);
        return errStatus;}
    printf("A24 Offset register value = 0x%hx\n", a24_offset);


    /* set the BUSY Delay Timer to 15mS  (see Busy Timer) */
    errStatus = viOut16 (m_mod,VI_A24_SPACE,0x12,0x1D6);
    if (VI_SUCCESS > errStatus){
        printf(“ERROR: viOut16() returned 0x%x\n”,errStatus);
        return errStatus;}

    /* Output a Value of 5 on DO Data Lines */
    errStatus = viOut16 (m_mod,VI_A24_SPACE,0x14,0x05);
    if (VI_SUCCESS > errStatus){
        printf(“ERROR: viOut16() returned 0x%x\n”,errStatus);
        return errStatus;}

/* Close the M-Module Instrument Session */
    err_status = viClose (m_mod);
    if (VI_SUCCESS > errStatus) {
        printf(“ERROR: viClose() returned 0x%x\n”,errStatus);
        return 0;}

/* Close the Resource Manager Session */
    err_status = viClose (viRM);
    if (VI_SUCCESS > errStatus) {
        printf(“ERROR: viClose() returned 0x%x\n”,errStatus);
        return 0;}

    return VI_SUCCESS;
    }
```

Register Descriptions

There are two sets of registers in the memory window of M-Modules. One is the same as a standard VXIbus instrument register in A16 address space (provided by the M-Module Carrier), the other is in A24 address space.

Registers in A16 Address Space

The five registers including the VXI ID Register, VXI Device Type Register, Status/Control Register, Offset Register, Interrupt Control These register are mapped in A16 address space.

Table 3-1 lists the five registers in the A16 memory space. The following paragraphs describe each register.

Table 3-1. VXIbus A16 Memory Instrument Registers

Address Mapping	Registers
00_h	VXI ID Register
02_h	VXI Device Type Register
04_h	VXI Status/Control Register
06_h	VXI Offset Register
20_h	M-Module Interrupt Control Register

VXI ID Register

The ID Register is a read only register at address 00_h and provides instrument identification information.

b+00_h	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read	Device Class		Address Space		Manufacturer ID											

- **Device Class:** this bit should always be 11 indicating a register-based device.
- **Address Space:** 00 indicating A16/A24 device
- **Manufacturer ID**: 4091 (decimal) for Racal Instruments M-Modules

VXI Device Type Register

The Device Type Register is a read only register at address 02_h. Reading this register returns a unique identifier for each M-Module.

b+02_h	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Undefined															
Read	Required Memory				M-Module Model Code											

- **Required Memory**: F_h indicating 256 byte block required.
- **M-Module Model Code**: 258_h for the M1700 .

VXI Status/Control Register

The Status/Control Register is a read/write register (address 04$_h$) that controls the module and indicates its status.

b+04$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write (Control)	A24 Enable	Reserved													Sysfail Inhibit	Reset
Read (Status)	A24 Active	MODID*	M-Module Device Dependent										Ready	Passed	Device Dependent	

- **A24 Enable.** A 1 in this bit means access to the devices A24 registers is enabled.
- **Sysfail Inhibit.** Writing a 1 disables the M-Module from driving the SYSFAIL* line.
- **Reset.** Writing a 1 then a 0 to this bit forces the M-Module to reset.
- **A24 Active.** A 1 in this bit indicates the M-Module's registers in A24 memory space can be accessed. Default = 1.
- **MODID*.** A 1 in this bit indicates that the M-Module is not selected via the P2 MODID line. A 0 indicates the M-Modules is selected by a high state on the P2 MODID line.
- **Ready.** A 1 in this bit indicates that the M-Module is ready to accept commands. A 0 indicates the M-Module is busy and not ready to accept commands.
- **Passed.** A 1 in this bit indicates the M-Module passed its self test successfully. A 0 indicates the M-Module is either executing or has failed its self test.

VXI Offset Register

The Offset Register (address 06$_h$) contains the value of the base address for accessing registers in the A24 address space.

b+06$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Register written to by VXI Resource Manager. Do not attempt to write to this register.															
Read	A24 Space Base address for those M-Modules needing A24 memory															

Interrupt Selection Register

The Interrupt Selection Register (address 20_h) specifies which VXI interrupt line the M-Module will use. M-Modules may generate interrupts to indicate that a SCPI command has completed. These interrupts are sent to and acknowledged by the Slot Zero Controller Module or other system controller via one of seven VXI backplane interrupt lines. Different controllers treat the interrupt lines differently, and you should refer to your controller's documentation to determine how to set the interrupt level. Slot Zero Modules configured as VXI Resource Managers treat all interrupt lines as having equal priority. For interrupters using the same line, priority is determined by which slot they are installed in; lower-numbered slots have higher priority than higher-numbered slots. Slot Zero Modules service line 1 by default, so it is normally correct to leave the interrupt level set to the factory default of IRQ1.

b+20_h	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Reserved												INT	VXI Interrupt Line		
Read	Reserved												INT 1	VXI Interrupt Line 0 0 1		

If your controller's documentation instructs you to change the interrupt level, you need to specify the level in the VXI Interrupt Selection Register. To cause the M-Module to interrupt on one of the VXI interrupt lines, write to the appropriate bits (refer to table below). To disable the module's interrupt, set the bits to 000. Selecting other than the default interrupt line 1 is not recommended. Reading the default value of this register returns the value XXX9_h.

Bits 2 - 0	Selected Interrupt Line
000	NONE (Interrupt Disabled)
001	IRQ1 (default)
010	IRQ2
011	IRQ3
100	IRQ4
101	IRQ5
110	IRQ6
111	IRQ7

M-Module specifications define three types of interrupts. The INT bit (bit 3) determines which M-Module interrupt style is supported. If INT is set to a 0, the M-Module supports interrupt types A and B. If INT is set to a 1, the M-Module supports interrupt type C (this is the default).

Type A Interrupts The interrupting M-Module removes the interrupt request upon a register access (software method) to the interrupting M-Module (such as reading the Status Register). DTACK* is not asserted during interrupt acknowledge.

Type B Interrupts The interrupting M-Module removes the interrupt request via a hardware method (on IACK* going low) but provides no vector information for the interrupt. This is the same as Type C interrupts except that no vector is supplied and DTACK* is not asserted.

Type C Interrupts The interrupting M-Module removes the interrupt request via a hardware method and provides an interrupt vector on the data bus and DTACK* is asserted during the interrupt acknowledge cycle. The M-Module removes the interrupt request by IACK* going low.

In VXI specifications however, only two types of interrupts are defined; RORA (Release on Register Access) and ROAK (Release on Acknowledge). The VX405C Carrier converts M-Module Type A interrupts to RORA and Types B and C interrupts to ROAK (default).

RORA Interrupts The interrupting device provides its logical address on the data bus (MD0 - MD7) during the interrupt acknowledge cycle that was initiated in response to its interrupt request. It does not remove the interrupt request until its Status/Control register is accessed.

ROAK Interrupts The interrupting device removes the interrupt request upon the presence of a properly addressed interrupt acknowledge cycle and provides its logical address on the data bus (MD0 - MD7). A cause/status byte is also placed on the data bus (MD15 - MD8)

Registers in A24 Address Space

There are several registers including a Status Register, Control Register, Interrupt Register, Delay Register, the Input/Output Register, and the Read/Write Decoder registers plus an EEPROM in A24 address space of the Breadboard Module. The Module also provides a set of decoder lines located from 20$_h$ through 2E$_h$. Table 3-2 lists the address mapping

Table 3-2. M1700 Registers in A24 Memory

Address	Registers
FE$_h$	EEPROM
30$_h$ - FC$_h$	Reserved
20$_h$ - 2E$_h$	Read/Write Decoder Registers
16$_h$ - 1E$_h$	Reserved
14$_h$	I/O Register
12$_h$	Delay Register
10$_h$	Reserved
(06$_h$ - 0F$_h$)	Reserved
04$_h$	Interrupt Register
02$_h$	Control Register
00$_h$	Status Register

A24 Status Register

The offset of Status Register is 00$_h$. It is a Read only register

b+00$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Reserved															
Read	Reserved								Busy*	Reserved					EIRQX	RIRQX

- BUSY*: 0-Circuit is busy (not stable yet).
- EIRQX: 1-External interrupt
- RIRQX:1-Circuit interrupt.

Control Register

The offset of Control Register is 02$_h$. It is a Read/Write register

b+02$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Reserved													EENABLE	RENABLE	Soft Reset
Read	Reserved													EENABLE	RENABLE	Soft Reset

- EENABLE: 1 - Enable External interrupt;
- RENABLE: 1 - Enable circuit interrupt (After BUSY timer);
- Soft Reset: 1 - Soft Reset M1700 .

When power-on or reset, all bits of Control Register are set to zero.

Interrupt Register

The offset of Interrupt Register is 04$_h$. It is a Read only register.

b+04$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Reserved															
Read	Reserved															RIRQX

- RIRQX: 1 - Circuit interrupt

BUSY Delay Timer Register

The offset of BUSY Delay Timer Register is 12$_h$. It is a write only register. The value of this register determines the delay time between command execution and asserting interrupt.The delay time is determined by the formula:

Delay Time = (Register value + 1) * 0.031875 ms

where: **Register value** can be from 0000$_h$ through FFFF$_h$

- The default value of **Delay Time** is 13 ms (register value = 407$_{10}$ or 197$_h$).

I/O Register

The offset of the Input/Output Register is 14$_h$. Reading this Register will get the status of TR0 - TR15 lines.Writing to this Register will latch MD15-MD0 into two 74ACTQ273 latches (BC0 - BC15).

b+14$_h$	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write	Output Bits (BC15 - BC0)															
Read	Input Bits (TR15 - TR0)															

- Writing a 1 sets bit high; writing a 0 sets bit low.

Read/Write Decoder Lines Registers

Writing a value to one of these registers asserts the corresponding WP line. Reading from one of these registers asserts the corresponding RP line. These decoder lines can be used to expand the I/O registers. Refer to Chapter 2 for timing information and using these registers.

b+20_h - b+$2E_h$	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
Write	Output Bits (BC15 - BC0)															
Read	Input Bits (TR15 - TR0)															

Table 3-3 lists the Read/Write Decoder Register Addresses and the corresponding WP and RP lines.

Table 3-3. Read/Write Decoder Lines and Register Addresses

WP / RP Line	**Register Address**		**WP / RP Line**	**Register Address**
WP0/ RP0	20_h in A24 Memory		**WP4 / RP4**	28_h in A24 Memory
WP1 / RP1	22_h in A24 Memory		**WP5 / RP5**	$2A_h$ in A24 Memory
WP2 / RP2	24_h in A24 Memory		**WP6 / RP6**	$2C_h$ in A24 Memory
WP3 / RP3	26_h in A24 Memory		**WP7 / RP7**	$2E_h$ in A24 Memory

EEPROM

There is an EEPROM on the Breadboard for identification. The offset of it is FE_h in A24 address space. The size of the EEPROM is 64 words (128 byte) and all the characteristics of M-Module are stored in it. The first 16 words (0 through 15) are defined in the M-Module Specifications; words 16 - 63 are described in Table 3-4.

Table 3-4. EEPROM Contents

Word #	Description	Value
0	Sync code	5346_h
1	module number (binary code)	1659_h
2	revision number (binary code)	0001
3	module characteristics	1868_h
4 - 7	reserved	
8 - 15	User-defined	
16	VXI Sync code	$ACBA_h$ (2's complement of 0x5346)
17	VXI-ID	$CFFF_h$
18	VXI-Device Type	$F258_h$
19 - 31	Reserved	
32 - 47	User-defined	
48 - 63	Reserved	

Appendix A
M1700 Specifications

The M1700 Double-Wide Breadboard Module complies with the Mezzanine Concept M-Module Specification.

The specifications of M1700 are listed in the following table:

ITEM	SPECIFICATIONS
User Component Area	70 cm^2 (10.86 in^2) occupying two slots.
Grid Hole Spacing	2.54 mm (0.1 in.)
Grid Hole Inside Diameter	1.17 mm (0.046 in.)
Maximum Component Height	5.3 mm (0.21 in.) above board in Region A, 10.5 mm (0.42 in.) above board in Region B. Refer to “M1700 Connections Layout on 1-5”.
Maximum Lead Length	1.2 mm (0.078 in.) below board. Refer to “M1700 Connections Layout on 1-5”.
Maximum Power Dissipation (per module)	Determined by mainframe cooling and/or C&H Technologies Model VX405C Cooling Capacity. Do not exceed 40 Watts.
Power Supplies	+5 VDC ±5% @ 100mA maximum +12 VDC ±5% @ 200mA maximum -12VDC ±5% @ 200mA maximum
Connectors	Two 44-pin connectors (user interface), Two 40-pin connectors to the carrier

Racal Instruments

REPAIR AND CALIBRATION REQUEST FORM

To allow us to better understand your repair requests, we suggest you use the following outline when calling and include a copy with your instrument to be sent to the Racal Repair Facility.

Model____________________Serial No.____________________ Date________________________

Company Name______________________________Purchase Order #_________________________

Billing Address____

City

____

State/Province Zip/Postal Code Country

Shipping Address____

City

____

State/Province Zip/Postal Code Country

Technical Contact______________________Phone Number (   )_____________________

Purchasing Contact____________________Phone Number (   )_____________________

1. Describe, in detail, the problem and symptoms you are having. Please include all set up details, such as input/output levels, frequencies, waveform details, etc.

____

____

____

____

2. If problem is occurring when unit is in remote, please list the program strings used and the controller type.____

____

____

3. Please give any additional information you feel would be beneficial in facilitating a faster repair time (i.e., modifications, etc.)____

____

____

____

4. Is calibration data required? Yes No (please circle one)

Call before shipping

Ship instruments to nearest support office.

Note: We do not accept
“collect” shipments.