Contents

HP E2251A M-Module Carrier User's Manual

Edition 1

Certification

Hewlett-Packard Company certifies that this product met its published specifications at the time of shipment from the factory. Hewlett-Packard further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology (formerly National Bureau of Standards), to the extent allowed by that organization's calibration facility, and to the calibration facilities of other International Standards Organization members.

Warranty

This Hewlett-Packard product is warranted against defects in materials and workmanship for a period of three years from date of shipment. Duration and conditions of warranty for this product may be superseded when the product is integrated into (becomes a part of)other HP products. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For warranty service or repair, this product must be returned to a service facility designated by Hewlett-Packard (HP). Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country

HP warrants that its software and firmware designated by HP for use with a product will execute its programming instructions when properly installed on that product. HP does not warrant that the operation of the product, or software, or firmware will be uninterrupted or error free.

Limitation Of Warranty

The foregoing warranty shall not apply to defects resulting from improper or inadequate maintenance by Buyer, Buyer-supplied products or interfacing, unauthorized modification or misuse, operation outside of the environmental specifications for the product, or improper site preparation or maintenance.

The design and implementation of any circuit on this product is the sole responsibility of the Buyer. HP does not warrant the Buyer's circuitry or malfunctions of HP products that result from the Buyer's circuitry. In addition, HP does not warrant any damage that occurs as a result of the Buyer's circuit or any defects that result from Buyer-supplied products.

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Notice

The information contained in this document is subject to change without notice. HEWLETT-PACKARD (HP) MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. HP shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material. This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Hewlett-Packard Company. HP assumes no responsibility for the use or reliability of its software on equipment that is not furnished by HP.

U.S. Government Restricted Rights

The Software and Documentation have been developed entirely at private expense. They are delivered and licensed as "commercial computer software" as defined in DFARS 252.227- 7013 (Oct 1988), DFARS 252.211-7015 (May 1991) or DFARS 252.227-7014 (Jun 1995), as a "commercial item" as defined in FAR 2.101(a), or as "Restricted computer software" as defined in FAR 52.227-19 (Jun 1987)(or any equivalent agency regulation or contract clause), whichever is applicable. You have only those rights provided for such Software and Documentation by the applicable FAR or DFARS clause or the HP standard software agreement for the product involved

HP E2251A M-Module Carrier User's Manual
Edition 1

Documentation History

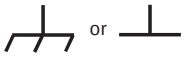
All Editions and Updates of this manual and their creation date are listed below. The first Edition of the manual is Edition 1. The Edition number increments by 1 whenever the manual is revised. Updates, which are issued between Editions, contain replacement pages to correct or add additional information to the current Edition of the manual. Whenever a new Edition is created, it will contain all of the Update information for the previous Edition. Each new Edition or Update also includes a revised copy of this documentation history page.

Edition 1 . . . . . June, 1997

Safety Symbols

Instruction manual symbol affixed to product. Indicates that the user must refer to the manual for specific WARNING or CAUTION information to avoid personal injury or damage to the product.

Indicates the field wiring terminal that must be connected to earth ground before operating the equipment — protects against electrical shock in case of fault.



Frame or chassis ground terminal—typically connects to the equipment's metal frame.

Alternating current (AC)

Direct current (DC).

Indicates hazardous voltages.

WARNING Calls attention to a procedure, practice, or condition that could cause bodily injury or death.

CAUTION Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNINGS

The following general safety precautions must be observed during all phases of 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 the product. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements.

Ground the equipment: For Safety Class 1 equipment (equipment having a protective earth terminal), an uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

DO NOT operate the product in an explosive atmosphere or in the presence of flammable gases or fumes.

For continued protection against fire, replace the line fuse(s) only with fuse(s) of the same voltage and current rating and type. DO NOT use repaired fuses or short-circuited fuse holders.

Keep away from live circuits: Operating personnel must not remove equipment covers or shields. Procedures involving the removal of covers or shields are for use by service-trained personnel only. Under certain conditions, dangerous voltages may exist even with the equipment switched off. To avoid dangerous electrical shock, DO NOT perform procedures involving cover or shield removal unless you are qualified to do so.

DO NOT operate damaged equipment: Whenever it is possible that the safety protection features built into this product have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the product until safe operation can be verified by service-trained personnel. If necessary, return the product to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

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 equipment: 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 a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

Declaration of Conformity

according to ISO/IEC Guide 22 and EN 45014

Manufacturer's Name: Hewlett-Packard Company
Loveland Manufacturing Center

Manufacturer's Address: 815 14th Street S.W.
Loveland, Colorado 80537

declares, that the product:

Product Name: Multi-Function Carrier

Model Number: HP E2251A

Product Options: All

conforms to the following Product Specifications:

Safety: IEC 1010-1 (1990) Incl. Amend 2 (1996)/EN61010-1 (1993)
CSA C22.2 #1010.1 (1992)
UL 3111-1 (1994)

EMC: CISPR 11:1990/EN55011 (1991): Group1 Class A
EN61000-3-2:1995 Class A
EN50082-1:1992
IEC 801-2:1991: 4kV CD, 8kV AD
IEC 801-3:1984: 3 V/m
IEC 801-4:1988: 1kV Power Line, 0.5kV Signal Lines
ENV50141:1993/prEN50082-1 (1995): 3 Vrms
ENV50142:1994/prEN50082-1 (1995): 1 kV CM, 0.5 kV DM
IEC1000-4-8:1993/prEN50082-1 (1995): 3 A/m
EN61000-4-11:1994/prEN50082-1 (1995): 30%, 10ms 60%, 100ms

Supplementary Information: The product herewith complies with the requirements of the Low Voltage Directive 73/23/EEC and the EMC Directive 89/336/EEC (inclusive 93/68/EEC) and carries the "CE" mark accordingly.

Tested in a typical configuration in an HP C-Size VXI mainframe.

Jim White, Quality Manager

April 7, 1997

European contact: Your local Hewlett-Packard Sales and Service Office or Hewlett-Packard GmbH, Department HQ-TRE, Herrenberger Straße 130, D-71034 Böblingen, Germany (FAX: +49-7031-14-3143)

Notes:

Please fold and tape for mailing

Reader Comment Sheet

HP E2251A M-Module Carrier User's Manual
Edition 1

You can help us improve our manuals by sharing your comments and suggestions. **In appreciation of your time, we will enter you in a quarterly drawing for a Hewlett-Packard Palmtop Personal Computer** (U.S. government employees are not eligible for the drawing).

Your Name	City, State/Province
Company Name	Country
Job Title	Zip/Postal Code
Address	Telephone Number with Area Code

Please list the system controller, operating system, programming language, and plug-in modules you are using.

fold here

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 37 LOVELAND, CO

POSTAGE WILL BE PAID BY ADDRESSEE

HEWLETT-PACKARD COMPANY
Measurement Systems Division
Learning Products Department
P.O. Box 301
Loveland, CO 80539-9984

cut along this line

fold here

Please pencil-in one circle for each statement below:

	Disagree ◄				► Agree
• The documentation is well organized.	O	O	O	O	O
• Instructions are easy to understand.	O	O	O	O	O
• The documentation is clearly written.	O	O	O	O	O
• Examples are clear and useful.	O	O	O	O	O
• Illustrations are clear and helpful.	O	O	O	O	O
• The documentation meets my overall expectations.	O	O	O	O	O

Please write any comments or suggestions below–be specific.

Chapter 1
Getting Started

What's in this Manual?

This manual contains a description of the HP E2251A M-Module Carrier (hereafter referred to as the Carrier) and generic M-Module[1] programming information using the Carrier. For M-Module installation instructions and instructions for installing the Carrier in a VXIbus mainframe, refer to the *HP E2251A Installation and Wiring Manual*.

HP E2251A M-Module Carrier Description

The HP E2251A M-Module Carrier allows up to six M-Modules[2] to be installed in a single C-Size VXI Mainframe slot, providing each M-Module with its own logical address and A16/A24 memory space. Each M-Module is accessed as though it is a standard VXIbus module occupying its own VXI slot in the cardcage. The Carrier:

- Provides a valid VXI logical address for each M-Module.
- Provides the four standard VXI registers for each M-Module (VXI Module ID, Device Type, Status / Control, and A24 Offset).
- Allows register-based access to the M-Module I/O space and memory.
- Supports Type A/B/C M-Module interrupts and handles them as VXI interrupts. Provides interrupt acknowledge daisy-chain for all six slots.
- Supports both D08/D16 data and A16/A24 addressing.
- Is completely transparent in the VXI system.
- Provides independently fused +5 and ±12V DC to each M-Module.

Figure shows the HP E2251A, RFI Dress Panels, and other accessories provided with the Carrier.

1. M-Modules comply with the VMEbus International Trade Association (VITA) Mezzanine Concept - M-Module Specifications.
2. You can install six M-Modules only if neither of the modules in the two interior slots require external cabling. If either of these interior modules require external cabling, the bottom-front slot (M0) must be left unused to accommodate the wiring.

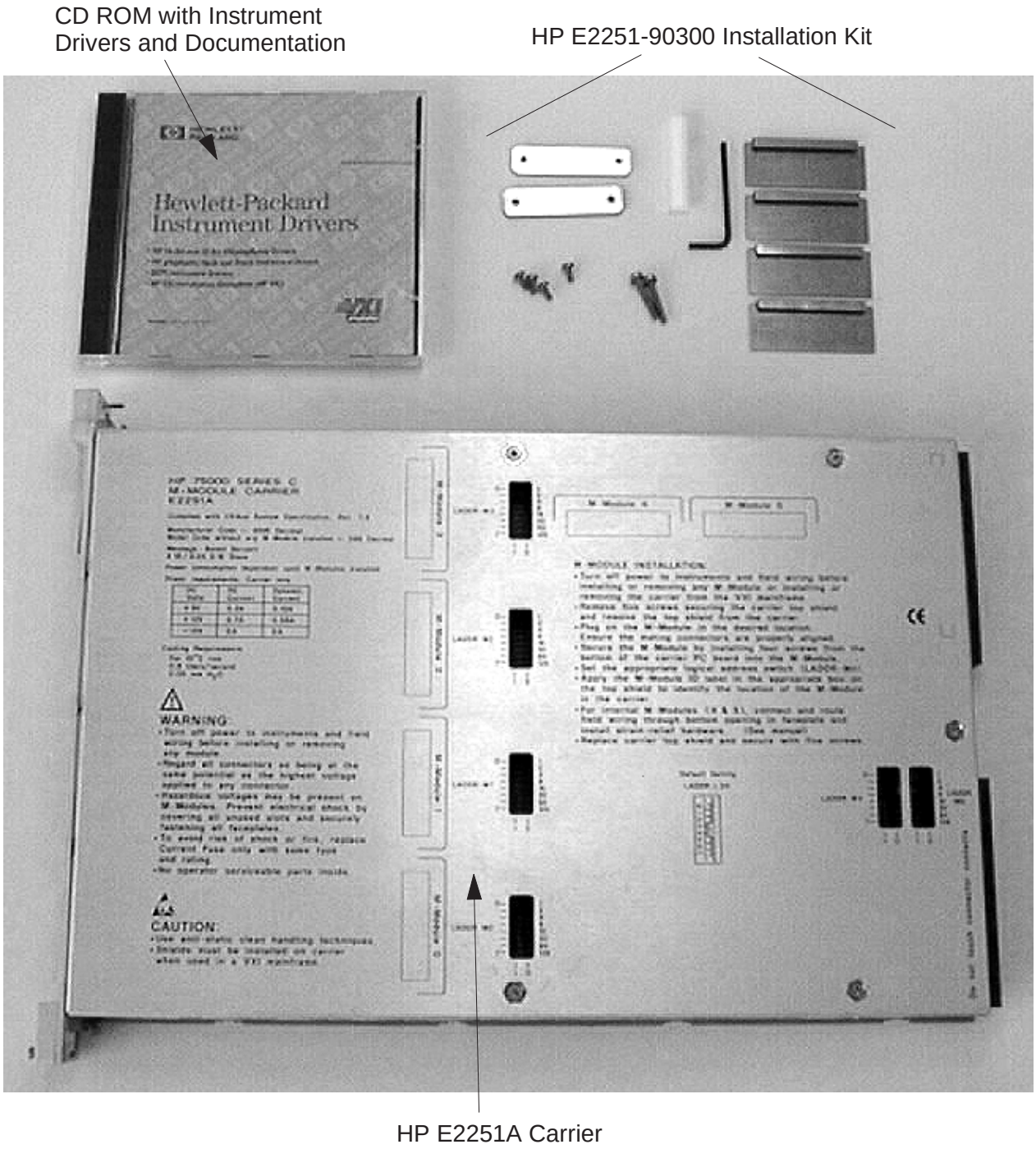


Figure 1-1. HP E2251A M-Module Carrier and Accessories

General Information

M-Modules are defined as modules which comply with the VMEbus International Trade Association (VITA) Mezzanine Concept M-Module Specification and have a 2-row, 40 pin connector for communication with the carrier. To make M-Modules work as standard VXIbus devices, the HP E2251A Carrier provides a logical address for each M-Module. The Carrier also reads the identification EEPROM from HP M-modules and copies data into the standard VXI registers (Manufacturer ID, Device Type, Status, etc.).

Based on the M-module specification, most M-Modules have an (optional) EEPROM which stores the characteristics of that module. The EEPROM is at least 16 words (32 bytes) deep. Note: all HP M-Modules have an EEPROM that is 64 words deep. The EEPROM words are defined in Chapter 3 of this manual.

Note If you are using a high level language driver, such as SCPI, you do not need to understand the register mapping of M-Modules. If you need to use register-based programming, refer to Chapter 3 of this manual and the register-based information chapter of the individual M-Modules.

You can use up to 31 M-Modules (up to 7 Carriers) with an HP E1406 Command Module or 255 M-Modules (up to 51 Carriers) with embedded controllers, VXLink, or MXIbus controllers.

Logical Addresses

Since the M-Module specification does not provide for the logical addresses required by the VXIbus specification, the HP E2251A Carrier provides a scheme for establishing logical addresses for each M-Module. The Carrier itself does not have a logical address and is transparent to the VXIbus. A set of logical address switches are located on the Carrier for each M-Module. The logical address provides a unique reference for addressing specific registers to control the M-Modules. Logical addresses set on the Carrier must not be used by any other M-Module or VXI module in the VXI mainframe. Addresses must be a multiple of 8 (that is, 8, 16, 24, . . . 256) if used with an HP E1406 Command Module. The HP E2251A Carrier does not support Dynamic Configuration of Logical Addresses.

Note Slots not used on the Carrier should have their logical address switches set to 0 (zero). Refer to the *HP E2251A Installation and Wiring Manual* for information on setting the logical addresses.

Power-on

When power is applied to the HP E2251A Carrier, it reads the EEPROM of each Hewlett-Packard M-Module and stores the Manufacturer ID and Device Type into registers on the Carrier. These registers on the Carrier are associated with an individual logical address supplied by the Carrier. The VXI Resource Manager allocates the proper A16/A24 register space for each of the M-Modules.

Register Addressing

VXI specifications provide for only 64 bytes of I/O space in A16 memory. However, the M-Module specifications provide for multiple 256 bytes of I/O space. To resolve this conflict, the HP E2251A Carrier provides two memory segments for the M-Modules: the first is in standard VXI A16 memory space and provides the standard VXI registers (Module ID, Device Type, Status/control, etc.), and the second is in A24 memory space for all of the other M-Module registers.

Interrupt Handling

The HP E2251A Carrier supports M-Module Type A, B, and C interrupts. Just as in standard VXI mainframes, the HP E2251A Carrier daisy-chains the IACKIN/IACKOUT lines for the M-Modules. If any M-Module slot on the Carrier is empty, the Carrier bypasses that slot automatically for the daisy chain. Type B and C interrupts are converted to standard VXI ROAK (Release on Acknowledge) interrupt by default. Refer to Chapter 3 in this manual for more details.

Front Panel LEDs

The HP E2251A has seven LEDs visible from the front panel. Refer to Figure 1-2. The FAIL LED indicates a failure on the HP E2251A Carrier (and VXI mainframe SYSFAIL line is asserted); the Carrier must be serviced or replaced.

LEDs M0 through M5 light when the respective M-Module is addressed and is communicating with the system commander.

Cleaning

Clean the outer surfaces of the HP E2251A M-Module Carrier with only a soft cloth slightly dampened with water. Do not use harsh chemicals or cleaners. Do not attempt to clean the interior of the M-Module Carrier.

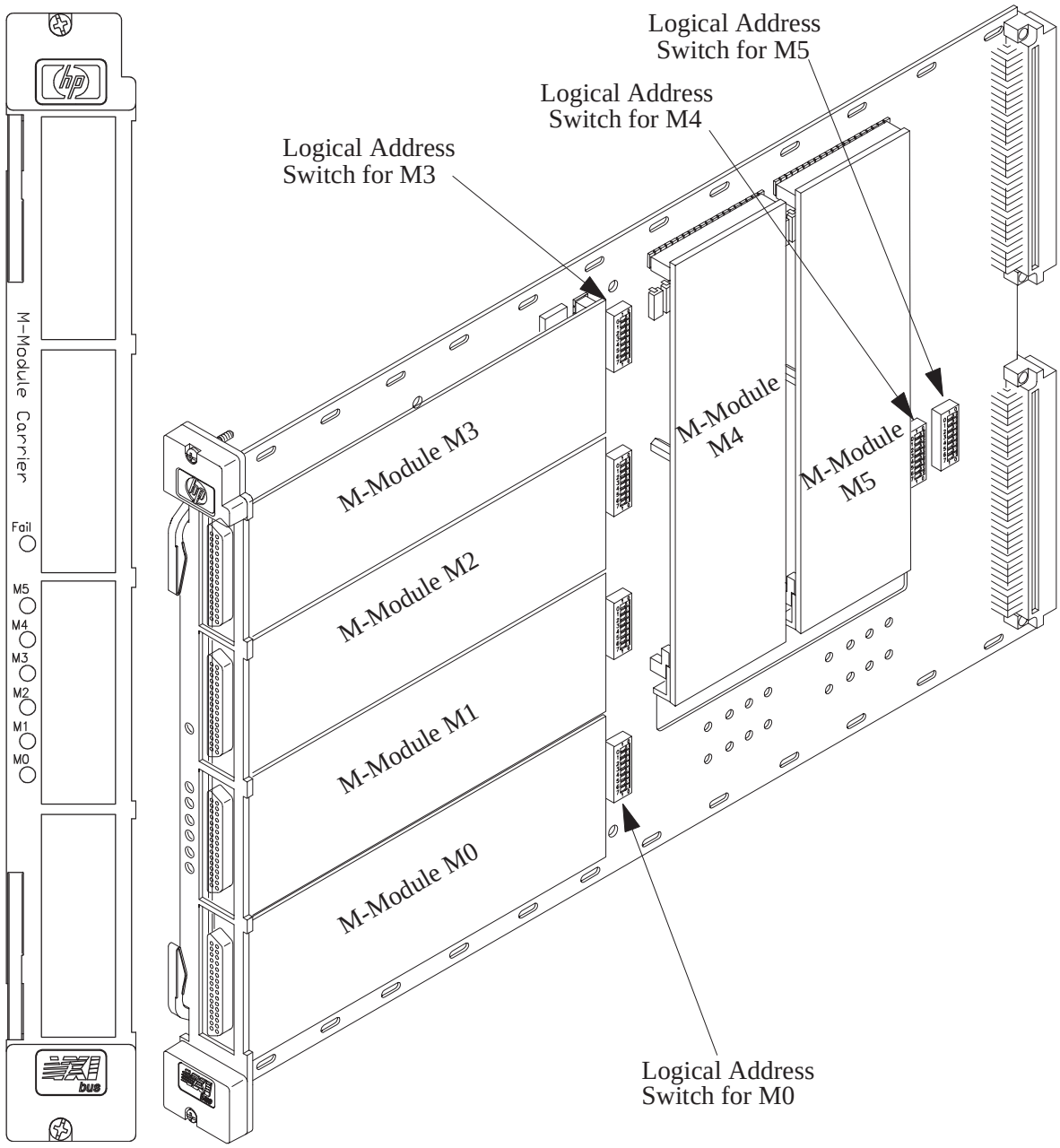


Figure 1-2. HP E2251A Front Panel and M-Module Layout

Connector Pinout

The control interface between the HP E2251A Carrier and each M-Module is via a 40-pin connector. Figure 1-3 shows the connector from the top-side of the HP E2251A Carrier. Table 1-1 lists the pin definitions.

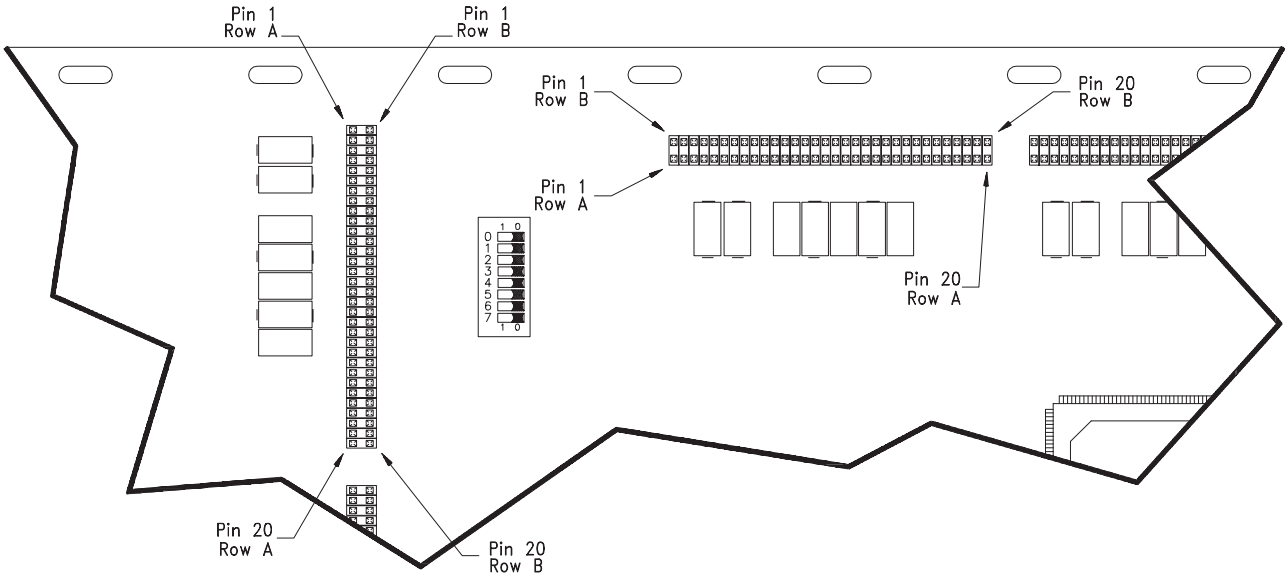


Figure 1-3. Control Interface Connector Pin Numbering

Table 1-1. Control Interface Connector Pin Definitions

Pin	Row A	Row B
1	CS*	Ground
2	A01	+5V
3	A02	+12V
4	A03	-12V
5	A04	Ground
6	A05	DREQ*
7	A06	DACK*
8	A07	Ground
9	D08	D00
10	D09	D01
11	D10	D02
12	D11	D03
13	D12	D04
14	D13	D05
15	D14	D06
16	D15	D07
17	DS1*	DS0*
18	DTACK*	WRITE*
19	IACK*	IRQ*
20	RESET*	SYSCLK

Chapter 2
SCPI Programming

Most HP M-Modules are provided with a downloadable SCPI driver. This chapter provides a brief overview of the Standard Commands for Programmable Instruments (SCPI) and summarizes IEEE 488.2 Common (*) Commands applicable to the HP M-Modules. This chapter also provides generic programming information for controlling HP M-Modules installed on an HP E2251A Carrier.

Note Do not do register writes if you are also controlling the modules by the downloaded SCPI driver. This is because the SCPI driver will not know the instrument state and an interrupt may occur causing the driver and/or command module to fail.

Command Types

Commands are separated into two types: IEEE 488.2 Common Commands and SCPI Commands.

Common Command Format

The IEEE 488.2 standard defines the common commands that perform functions such as reset, self-test, status byte query, and so on. Common commands are four or five characters in length, always begin with the asterisk character (*), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of common commands are shown below:

```
*RST        *ESR 32        *STB?
```

SCPI Command Format

The SCPI commands perform functions like closing switches, making measurements, and querying instrument states or retrieving data. A subsystem command structure is a hierarchical structure that usually consists of a top level (or root) command, one or more lower level commands, and their parameters. The following example shows part of a typical subsystem:

```
[ROUTe:]
    CLOSe <channel_list>
    CLOSe? <channel_list>
    OPEN <channel_list>
    OPEN? <channel_list>
```

[ROUTe:] is the root command, CLOSe, CLOSE?, OPEN, and OPEN? are second level commands with parameters.

Command Separator

A colon (**:**) always separates one command from the next lower level command as shown below:

```
STATus:OPERation:CONDition?
```

Colons separate the root command from the second level command (STATus:OPERation) and the second level from the third level (OPERation:CONDition?).

Abbreviated Commands

The command syntax shows most commands as a mixture of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send the abbreviated form. For better program readability, you may send the entire command. The instrument will accept either the abbreviated form or the entire command. For example, if the command syntax shows `MEASure`, then `MEAS` and `MEASURE` are both acceptable forms. Other forms of `MEASure`, such as `MEASU` or `MEASUR` will generate an error. You may use upper or lower case letters. Therefore, `MEASURE`, `measure`, and `MeAsUrE` are all acceptable.

Implied Commands

Implied commands are those which appear in square brackets ([]) in the command syntax. (Note that the brackets are not part of the command and are not sent to the instrument.) Suppose you send a second level command but do not send the preceding implied command. In this case, the instrument assumes you intend to use the implied command and it responds as if you had sent it. Examine the partial [ROUTe:] subsystem shown below:

```
[ROUTe:]
    CLOSe <channel_list>
    CLOSe? <channel_list>
    OPEN <channel_list>
    OPEN? <channel_list>
```

The root command [ROUTe:] is an implied command. To close relays in a channel list, you can send either of the following command statements:

```
ROUT:CLOS (@02:05,07,09:11)
or
CLOS (@02:05,07,09:11)
```

These commands function the same - closing channels 002 throough 05, 07, and 09 through 11.

Note A range of channels (*@nn:nn*) must be specified in ascending order, that is lower channel number on hte left, higher channel number on the right.

Command Parameters

The following table contains explanations and examples of parameter types you might see later in this chapter.

Parameter Type	Explanations and Examples
Numeric	Accepts all commonly used decimal representations of number including optional signs; decimal points; and scientific notation. 123; 123E2; -123; -1.23E2; .123; 1.23E-2; 1.23000E-01. Special cases include MINimum; MAXimum; and DEFault.
Boolean	Represents a single binary condition that is either true or false. ON; OFF; 1; 0
Discrete	Selects from a finite number of values. These parameters use mnemonics to represent each valid setting. An example is the TRIGger:SOURce <source> command where <source> can be BUS; EXT; HOLD; or IMM.

Optional Parameters. Parameters shown within square brackets ([ ]) are optional parameters. (Note that the brackets are not part of the command and are not sent to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value. For example, consider the `ARM:COUNt? [ MIN | MAX ]` command. If you send the command without specifying a parameter, the present `ARM:COUNt` setting is returned. If you send the `MIN` parameter, the command returns the minimum count available. If you send the `MAX` parameter, the command returns the maximum count available. Be sure to place a space between the command and the parameter.

Linking Commands

Linking IEEE 488.2 Common Commands with SCPI Commands. Use a semicolon between the commands. For example:

```
*RST;OUTP ON
```

or

```
TRIG:SOUR HOLD;*TRG
```

Linking Multiple SCPI Commands. Use both a semicolon and a colon between the commands. For example:

```
ARM:COUN 1;:TRIG:SOUR EXT
```

Example Programs

This section provides examples demonstrating SCPI programming M-Modules installed in the Carrier.

Note Do not do register writes if you are also controlling the modules by a high level language such as SCPI. This is because the high level language driver will not know the instrument state and an interrupt may occur causing the driver and/or command module to fail.

Initial Operation

The following example reads the module ID string, performs module self-test, and displays the results. This program can be used on any of the HP M-Modules.

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

    /* Interface address is 9, M-Module secondary address is 3*/
#define INSTR_ADDR "GPIB0::9::3::INSTR"

int main()
{
   ViStatus errStatus;                  /*Status from each VISA call*/
   ViSession viRM;                      /*Resource mgr. session */
   ViSession m_mod;                     /* M-module session  */
   char id_string[256];                 /*ID string*/
   char selftst_string[256];            /*self-test string*/

    /* 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;}

    /* Reset the M-Module */
   errStatus = viPrintf(m_mod, "*RST\n");
   if(VI_SUCCESS > errStatus){
      printf("ERROR: viviPrintf() returned 0x%x\n",errStatus);
      return errStatus;}
```

```
    /* Perform M-Module Self-Test */
    errStatus = viQueryf(m_mod,"*TST?\n","%t",selftst_string);
    if (VI_SUCCESS > errStatus) {
        printf("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}
    printf("Self Test Result is %s\n",selftst_string);

    /* Query the M-Module ID string */
    errStatus = viQueryf(m_mod,"*IDN?\n","%t",id_string);
    if (VI_SUCCESS > errStatus) {
        printf("ERROR: viPrintf() returned 0x%x\n",errStatus);
        return errStatus;}
    printf("ID is %s\n",id_string);

    /* 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;
}
```

Example: Closing Multiple Channels

The following example closes channels 01 and 10 through 13 on either the HP E2270A, E2271A, E2272A Switch M-Modules. The program then opens channels 01 and 11. The program assumes an M-Module secondary address of 3 and an interface address of 9.

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

    /* Interface address is 9, M-Module secondary address is 3*/
#define INSTR_ADDR "GPIB0::9::3::INSTR"

int main()
{
   ViStatus errStatus;                  /*Status from each VISA call*/
   ViSession viRM;                      /*Resource mgr. session */
   ViSession m_mod;                     /* M-module 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;}

    /* Reset the M-Module */
   errStatus = viPrintf(m_mod, "*RST\n");
   if(VI_SUCCESS > errStatus){
      printf("ERROR: viviPrintf() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Close channels 1 and 10 through 13 on the M-Module */
   errStatus = viPrintf(m_mod,"ROUT:CLOS (@01,10:13)\n");
   if (VI_SUCCESS > errStatus) {
      printf("ERROR: viPrintf() returned 0x%x\n",errStatus);
      return errStatus;}

    /* Open channels 1 and 11 on the M-Module */
   errStatus = viPrintf(m_mod,"ROUT:OPEN (@01,11)\n");
   if (VI_SUCCESS > errStatus) {
      printf("ERROR: viPrintf() returned 0x%x\n",errStatus);
      return errStatus;}
```

```
    /* 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;
}
```

IEEE 488.2 Common Command Reference

The following table lists the IEEE 488.2 Common (*) Commands accepted by the HP M-Modules. For more information on Common Commands, refer to the *HP 75000 Series C Mainframe User's Manual* or the ANSI/IEEE Standard 488.2-1987.

Command	Command Description
*CLS	Clears all status registers and clears the error queue.
*ESE *<register value>*	Enable Standard Event.
*ESE?	Enable Standard Event Query.
*ESR?	Standard Event Register Query.
*IDN?	Instrument ID Query; returns identification string of the module.
*OPC	Operation Complete.
*OPC?	Operation Complete Query.
*RCL *<numeric state>*	Recalls the instrument state saved by *SAV.
*RST	Resets the module to its power-on/reset state.
*SAV *<numeric state>*	Stores the instrument state.
*SRE *<register value>*	Service request enable, enables status register bits.
*SRE?	Service request enable query.
*STB?	Read status byte query.
*TRG	Triggers the module (if module is equipped for triggers). **The HP E2251A does not support trigger capabilities.**
*TST?	Self-test. Executes an internal self-test and returns only the first error encountered. Does not return multiple errors.
*WAI	Wait to Complete.

Note These commands apply to VXI instruments as well as M-Modules. See the *HP 75000 Series C E1400/E1401 Mainframe User's Manual* or the ANSI/IEEE Standard 488.2-1987 for more information about these commands. Refer to the individual M-Module or VXI intrument user manual for details on IEEE 488.2 Common Command actions and results.

Chapter 3
M-Module Register-Based Programming

To fully understand the operation of M-Modules installed on the HP E2251A M-Module Carrier, you need to know how the Carrier operates and how M-Module Registers are mapped onto the Carrier. This chapter presents a basic tutorial for M-Module register-based programming.

Note If you are using a high-level driver, such as SCPI, you do not need to know the register-based information contained in this chapter. Do not do register writes if you are also controlling the modules by the SCPI drivers. This is because the high level driver will not know the instrument state and an interrupt may occur causing the driver and/or command module to fail.

HP E2251A Block Diagram

Figure 3-1 shows the overall block diagram of the HP E2251A M-Module Carrier. Refer to the following discussion to understand carrier operation.

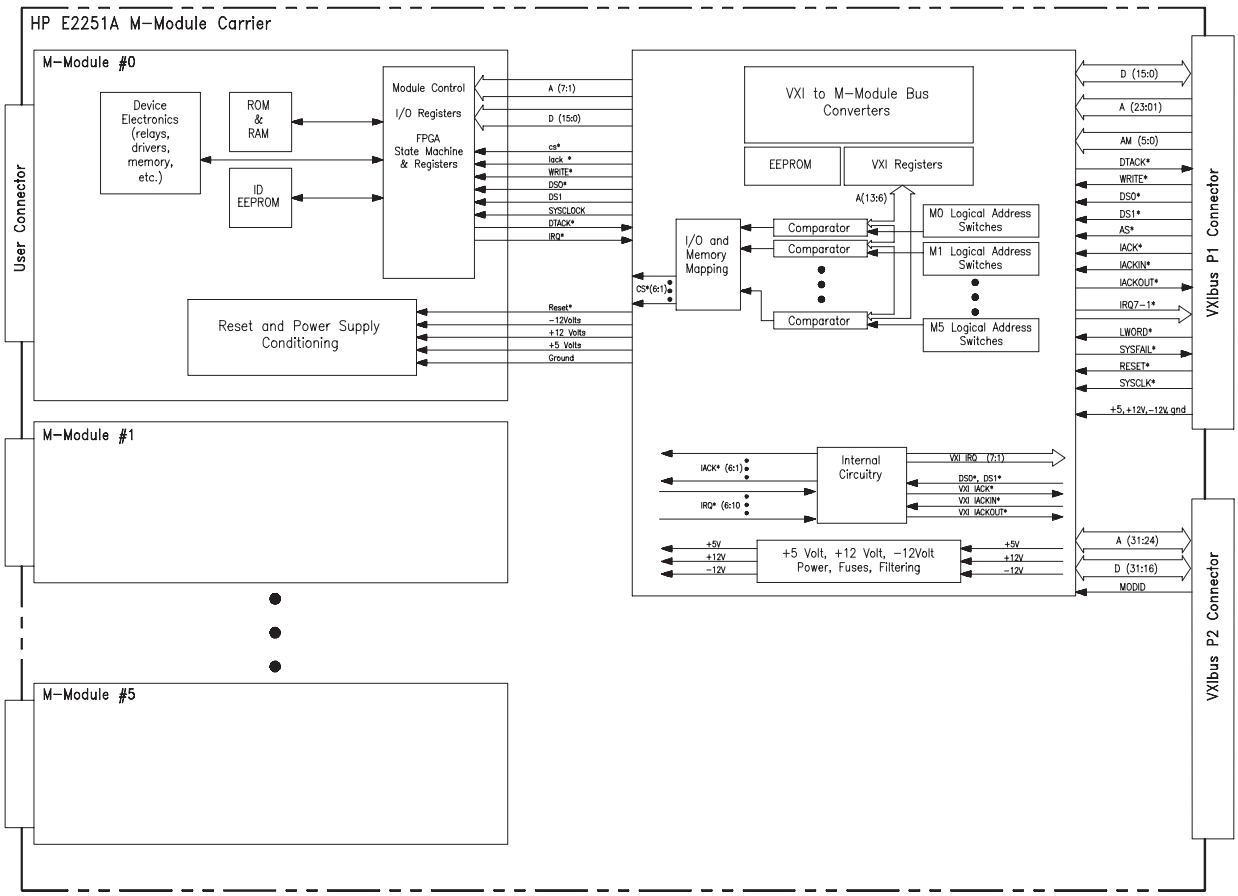


Figure 3-1. HP E2251A Simplified Block Diagram

Logical Address

The VMEbus International Trade Association (VITA) Mezzanine Concept M-Module Specification does not define logical address for M-Modules. To make M-Modules work as standard VXIbus devices, the HP E2251A Carrier provides a logical address for each M-Module. The HP E2251A also provides standard VXIbus registers (Manufacturer ID, Device Type, Status/Control, etc.) for each M-Module. These standard registers are located in the lowest 64-byte space for register-based VXIbus instruments.

Note Logical Addresses set on the HP E2251A Carrier cannot be used anywhere else in the VXI system. Unused slots in the Carrier should have their logical address switches set to 0's to be ignored by the Carrier and VXI system. Refer to the *Hp E2251a Installation And Wiring Manual* for information on setting the logical addresses.

M-Module Register Mapping

Based on the M-module specification, each M-Module has an optional 16 word deep (32 bytes) Identification EEPROM which stores the characteristics of that module. Note: EEPROMs on all HP M-Modules are (at least) 64 words deep. The HP E2251A Carrier copies these extra words into its memory and then the VXI System Resource Manager maps them into A16 and A24 VXI memory registers. Table 3-1 lists the words on HP M-Modules.*

Table 3-1. M-Module EEPROM Words

	Word #	Description	Default Value
Standard M-Module Words	0	Sync Code	5346_h
	1	Module Number (see Table 3-2)	(M-Module Dependent)
	2	Revision Number	(M-Module Dependent)
	3	Module Characteristics (refer to M-Module specifications for bit definitions)	(M-Module Dependent)
	4 - 7	Reserved	
	8-15	M-Modules Specific	(Module Dependent)
Words used on HP M-Modules for VXI use	16	VXI Sync Code	$ACBA_h$ (2's complement of 5346_h)
	17	VXI ID	(M-Module Dependent)
	18	Model Code (see Table 3-1)	(M-Module Dependent)
	19-31	Reserved	
	32-63	M-Modules Specific	(M-Module Dependent)

* Numbers with a subscripted h, such as $1F0000_h$, are shown in hexadecimal format. Numbers without the subscripted h are decimal.

VXI specifications provide for only 64 bytes of I/O space in A16 memory. However, the M-Module specifications provide for multiple 256 bytes of I/O space. To resolve this conflict, the HP E2251A Carrier provides two memory segments for each of the installed M-Modules: the first segment is in VXI A16 memory space and provides standard VXI registers (The M-module's VXI ID word is mapped into the VXI A16 memory as the ID Register at address 00_h and the M-Module's VXI Device Type word is mapped into the VXI Device Type Register at 02_h, a VXI-A24 Offset register at 06_h, and optionally other module-specific registers. The second memory segment is in A24 memory space for all of the M-Module registers.

These registers may have different bit descriptions than standard VXI registers; refer to the register descriptions later in this chapter and in the individual M-Module user manuals for detailed register information.

Table 3-2. HP M-Module Model Codes

HP M-Module Model Number	M-Module Number (M-Module Specification)	M-Module Model Code (VXI Specification)
HP E2251A Empty Slot (but with Logical Address switches set to other than 0)	1651	$F257_h$
HP E2259A Double-Wide Breadboard M-Module	1659	$F258_h$
HP E2261A Quad RS-232 Interface M-Module	1661	$F25A_h$
HP E2270A 16-Channel Form A Switch M-Module	1670	$F25B_h$
HP E2271A 4x4 Matrix Switch M-Module	1671	$F25C_h$
HP E2272A Dual 8-to-1 Relay Multiplexer M-Module	1672	$F25D_h$
HP E2273A 8-Channel Form C Switch M-Module	1673	$F25E_h$
HP E2274A 4-Channel Form C Power Relay M-Module	1674	$F25F_h$
HP E2290A 16-Bit Digital I/O M-Module	1690	$F260_h$
HP E2291A 16-Channel Isolated Digital Output M-Module	1691	$F261_h$

At Power-on

When power is applied to the VXI mainframe (and hence to the Carrier and M-Modules) the HP E2251A Carrier reads the EEPROM of each installed M-Module and checks each M-Module's Sync Code word (5346_h). If the sync code can be read, the Carrier enables the interrupt handling circuitry for that M-Module. If the sync code cannot be read from the M-Module (either the Carrier slot does not have an M-Module installed, the M-Module does not have an EEPROM, or the M-Module is non-HP), the Carrier establishes IACKIN as pass-through (much the same as a VXI mainframe "passes-through" IACKIN for empty slots) for that slot.

Addressing the Registers

Register addresses for the M-Modules (and VXI modules) are found in address space known as A16 and A24. The exact location of the A16 address space within a VXIbus master's memory map depends on the design of the VXIbus master you are using; for example, in the HP E1406 Command Module, the A16 memory space starts at address $1F0000_h$.

The A16 space is further divided so that the modules are addressed only at locations above $1FC000_h$ within A16. Further, every module is allocated 64 register addresses (40_h). The address of a module is determined by its logical address (set by the address switches on the Carrier for each M-Module) times 64 (40_h). For example, if an M-Module's logical address is 120 (78_h), the register addresses for that module start at $1FDE00_h$:

$$1FC000_h + (78_h * 40_h) = 1FDE00_h$$

A16 Register Space

The base address used in register base programming depends on whether the A16 address space is inside or outside the HP E1406 Command Module.

A16 Address Space Inside an HP Command Module

When the A16 address space is inside a Command Module (see Figure 3-2), the module's base address is computed as:

$$1FC000 + (LADDR_h * 40_h)$$

or (decimal)

$$2{,}080{,}768 + (LADDR * 64)$$

where $1FC000_h$ (2,080,768) is the starting location of the VXI A16 addresses, LADDR is the module's logical address, and 64 is the number of address bytes per register-based device. Again, this example uses an M-Module's logical address of 120. If this address is not changed, the modules will have a base address of:

$$1FC000_h + (78_h * 40_h) = 1FC000_h + 1E00_h = 1FDE00_h$$

or (decimal)

$$2{,}080{,}768 + (120*64) = 2{,}080{,}768 + 7680 = 2{,}088{,}448$$

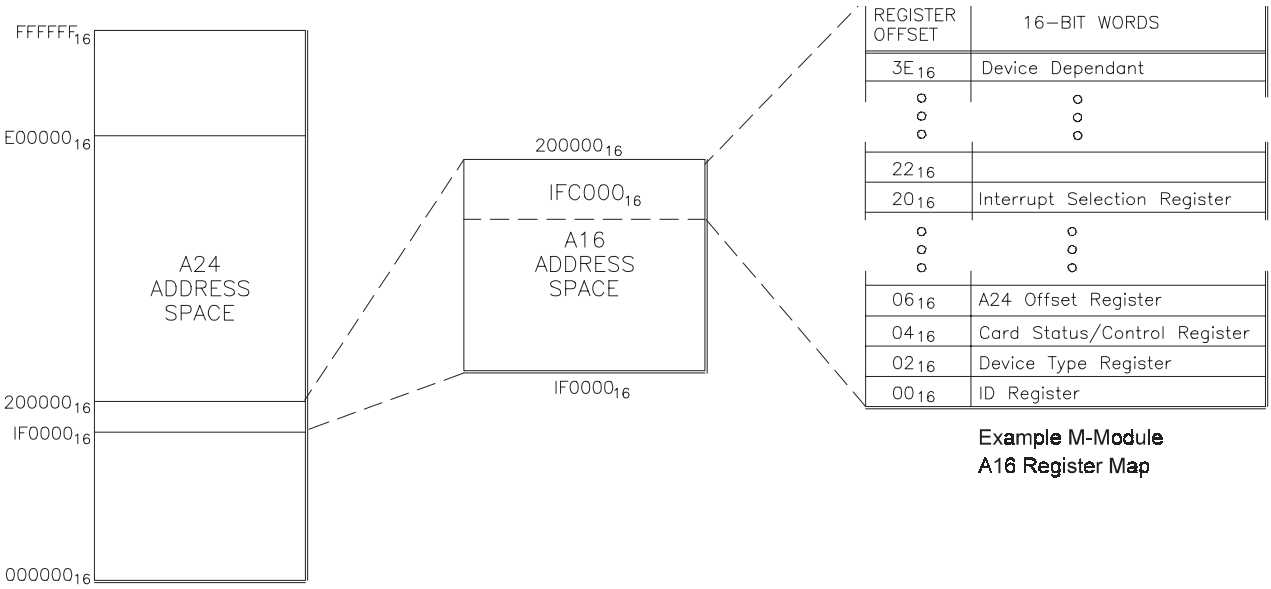


Figure 3-2. Register Address Space in an HP Command Module (such as HP E1406)

A16 Address Space Outside a Command Module

When an HP Command Module is not a part of your VXIbus system (see Figure 3-3), the M-Module's base address is computed as:

$$A16_{base} + C000_h + (LADDR_h * 40_h)$$

or (decimal)

$$A16_{base} + 49{,}152 + (LADDR * 64)$$

where $C000_h$ (49,152) is the starting location of the register addresses, LADDR is the module's logical address, and 64 is the number of address bytes per VXI device. For example, if an M-Module's logical address (LADDR) is 120 (78_h); it will have a base address of:

$$A16_{base} + C000_h + (78_h * 40_h) = C000_h + 1E00_h = DE00_h$$

or (decimal)

$$A16_{base} + 49{,}152 + (120 * 64) = 49{,}152 + 7{,}680 = 56832$$

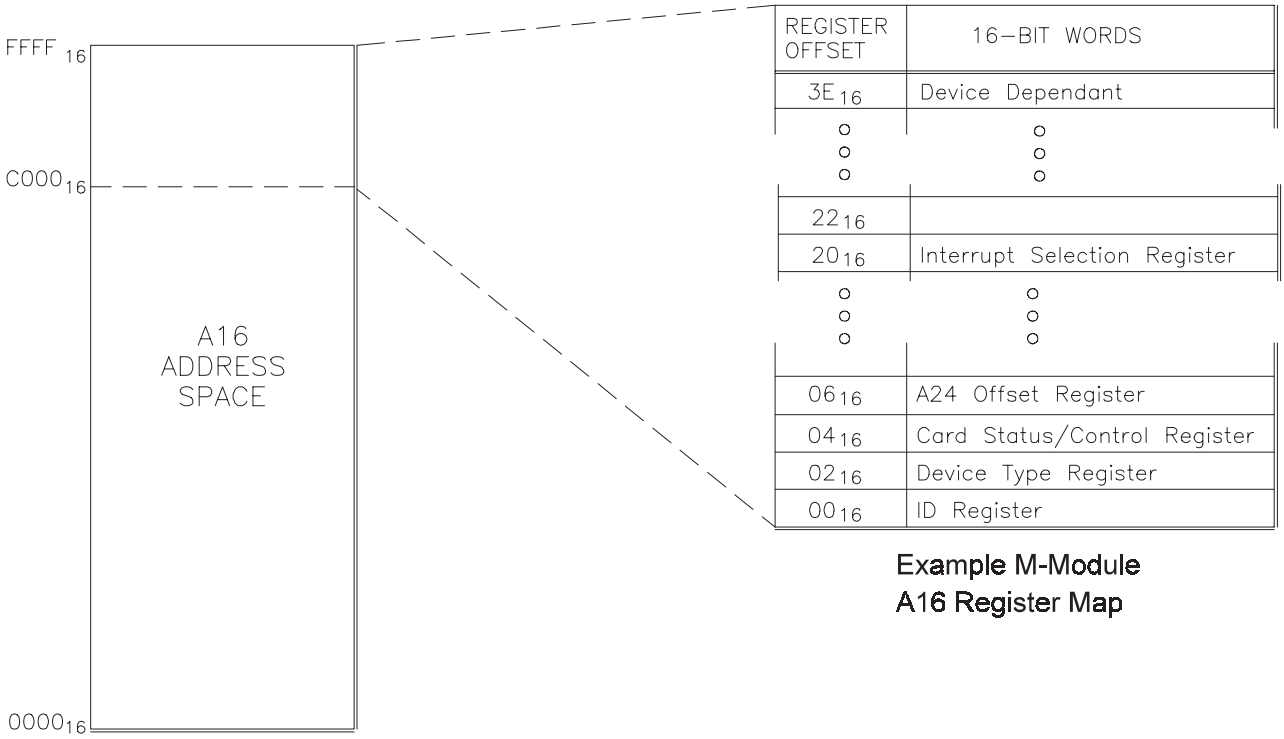


Figure 3-3. Register Address Location in A16 Memory Space

A24 Register Space

Most M-Modules use A24 memory for specific, unique purposes. If an M-Module uses A24 memory space, the VXI Resource manager performs the following steps:

1. Reads the M-Module's ID Register Address Space field (bits 12 and 13) to determine if the module uses A24.

2. Reads the M-Module's Device Type Register Required Memory field (bits 12 - 15) to determine the size of the A24 memory space needed by the M-Module.

3. Assigns the A24 offset for the M-Module and ensures no overlap.

4. Assigns the A24 base address and writes it to the Offset Register (06_h in A16 memory space).

5. Enables the M-Module's A24 registers by writing a '1' to the A24 Enable bit (bit 15) of the Status/Control Register.

To access the registers in A24 memory you must:

1. Verify that the registers are enabled by reading the A24 Enable bit (bit 15) of the VXI Status/Control Register and verifying that it is a '1'.

2. Obtain the A24 Base Address by reading the VXI Offset Register (06_h) in the A16 memory space.

3. Use this A24 Base Address plus the specific register offset specified for the M-Module. Refer to the individual M-Module's User Manual for detailed register descriptions.

Figure 3-4 shows a typical M-Module register mapping in A24 Address Space

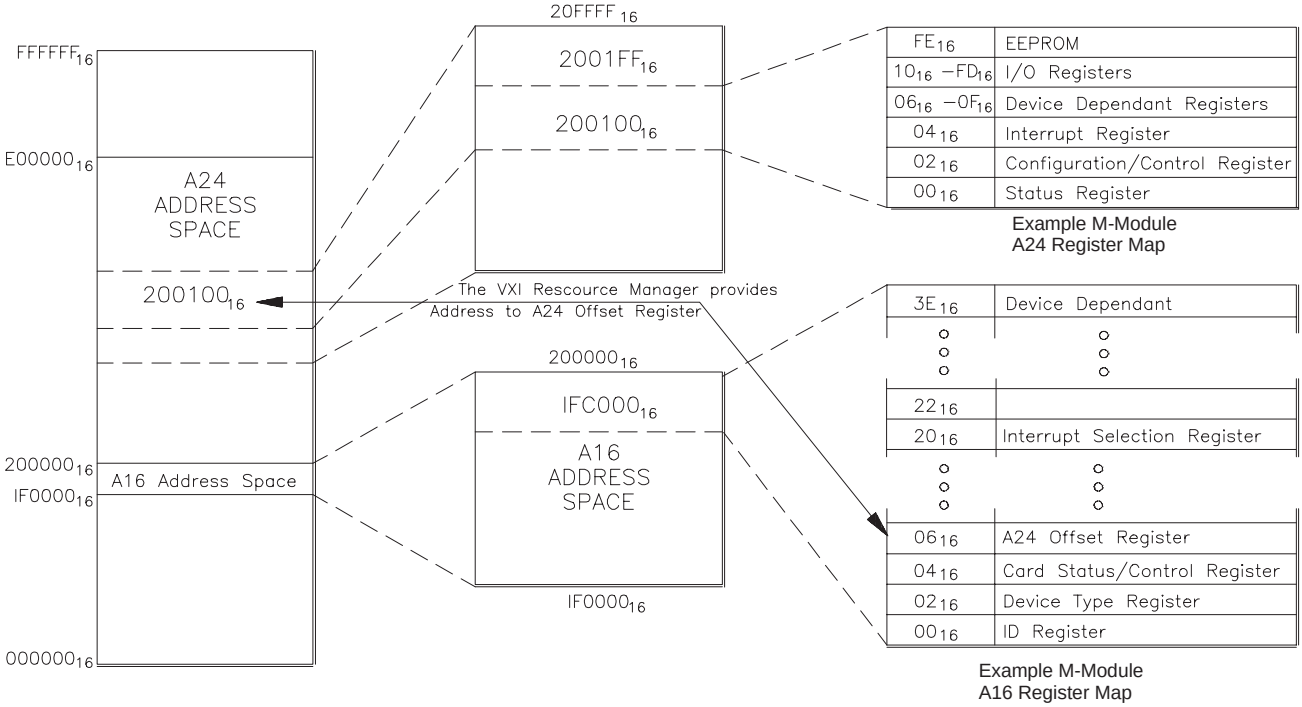


Figure 3-4. Register Mapping in A16 and A24 Address Space

Accessing the Registers

An M-Module's VXI register address consists of a base address (either A16 or A24) plus an offset. For example, an M-Module's Status/Control Register has an offset of 04_h and is in A16 address space. When you write to or read from this register, the offset is added to the base address to form the register address. Similarly, to access a register in A24 memory, you may need to read the A24 Offset Register (06_h in A16 memory) to obtain the A24 Base Address.

Example

Figure 3-5 shows an example register map for six M-Modules installed on the Carrier. The VXI Resource Manager (in this example, an HP E1406 Command Module is the Resource Manager) allocates both A16 and A24 memory space to each M-Module. An HP E2273A is installed on the Carrier and has its logical address switches set to 24 (or 36_h). The base address for this module is:

$$1FC000_h + (36_h * 40_h) = 1FC600_h$$

The Device Type register for this module has an offset of 02_h. Therefore to read this register, use the address 1FC602:

$$1FC600_h + 02_h = 1FC602_h$$

Reading this register returns the value $F25E_h$ identifying it as an HP E2273A.

Similarly, use the address $1FC606_h$ to read the A24 Offset register:

$$1FC600_h + 06_h = 1FC606_h$$

Reading this register returns the value 2600_h which is the most significant 16 bits of the A24 address; the remaining eight bits are always 00_h. Therefore, to access to read or write any of the registers in A24 memory, use the base address 260000_h plus the register offset.

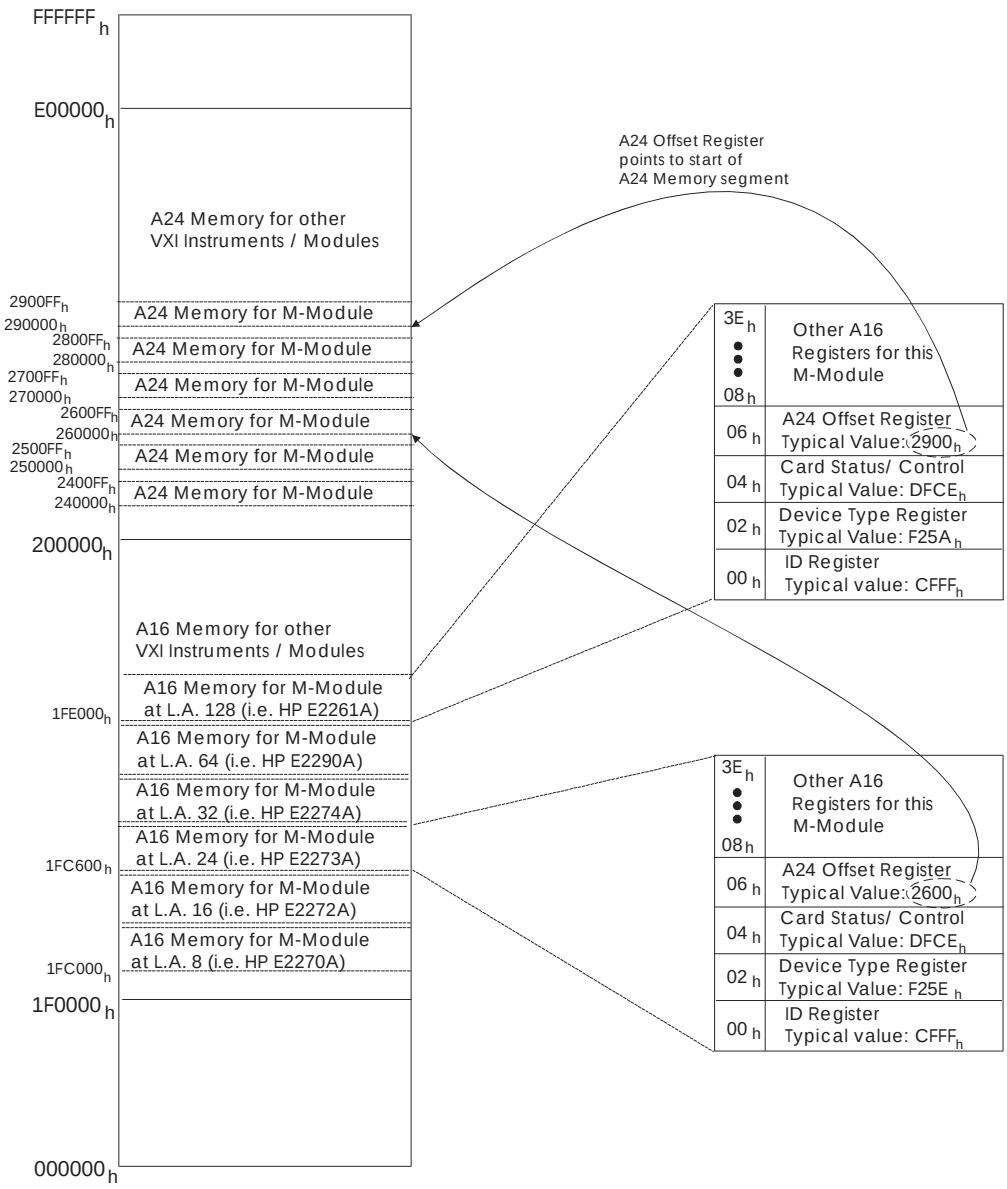


Figure 3-5. Example Register Addressing for M-Modules on the HP E2251A Carrier

VXI A16 Register Descriptions

All HP M-Module's installed in the HP E2251A Carrier have the following A16 registers. The register bit definitions may be different than standard VXI bit definitions.

Note

M-Modules may use A16 space registers in addition to the ones describe here. They may also have memory space mapped into A24 Memory Space. Refer to the individual M-Module user manual for details on specific registers, their addressing, and use.

VXI ID Register

The ID Register is a read only register at address 00_h (MSB) and 01_h (LSB).

$b+00_h$	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** indicates the classification of the VXI bus device. 00 is for memory, 01 is extended memory, 10 is for Message-based devices, and 11 is for a register-based module. The HP E2251A Carrier supports only register-based M-Modules and this field should always be 11.
- **Address Space** indicates the addressing modes of the module's operational registers.

Bit Pattern	Addressing Mode
00	A16/A24 (Since all HP M-Modules are A16/A24, this field should always be 00)
01	A16/A32
10	reserved
11	A16 devices only

- **Manufacturer ID** = 4095 (decimal) for Hewlett-Packard M-Modules

VXI Device Type Register

The Device Type Register is a read only register at address 02_h (MSB) and 03_h (LSB). Reading this register returns a unique identifier for each M-Module. Refer to the individual M-Module's User Manual for Device Type Register details.

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** specifies the memory required by the M-Module.

Bits 15 - 12	Memory Required		Bits 15 - 12	Memory Required
1111	256 Bytes		0111	64 kBytes
1110	512 Bytes		0110	128 kBytes
1101	1 kBytes		0101	256 kBytes
1100	2 kBytes		0100	512 kBytes
1011	4 kBytes		0011	1 MBytes
1010	8 kBytes		0010	2 MBytes
1001	16 kBytes		0001	4 MBytes
1000	32 kBytes		0000	8 MBytes

- M-Module Model Code is a unique identifier for that M-Module. For example, F25E$_h$ for HP E2273A 8-Channel Form C Switch M-Module or F25F$_h$ for HP E2274A 4-Channel Form C Power Relay M-Module. If an M-Module is installed that does not have an EEPROM, the default is F257$_h$. The following lists the valid Model Codes for the current HP M-Modules.

HP M-Module Model Number	M-Module Model Code
HP E2251A Empty Slot (but with Logical Address switches set to other than 0)	F257$_h$
HP E2259A Double-Wide Breadboard M-Module	F258$_h$
HP E2261A Quad RS-232 Interface M-Module	F25A$_h$
HP E2270A 16-Channel Form A Switch M-Module	F25B$_h$
HP E2271A 4x4 Matrix Switch M-Module	F25C$_h$
HP E2272A Dual 8-to-1 Relay Multiplexer M-Module	F25D$_h$
HP E2273A 8-Channel Form C Switch M-Module	F25E$_h$
HP E2274A 4-Channel Form C Power Relay M-Module	F25F$_h$
HP E2290A 16-Bit Digital I/O M-Module	F260$_h$
HP E2291A 16-Channel Isolated Digital Output M-Module	F261$_h$

VXI Status/Control Register

The Status/Control Register is a read/write register (address 04_h and 05_h) that controls the module and indicates its status. Refer to the individual M-Module's User Manual for Status/Control Register details.

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

VXI Offset Register

The Offset Register (address 06_h and 07_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	A24 Space Base address for those M-Modules needing A24 memory															
Read	A24 Space Base address for those M-Modules needing A24 memory															

Interrupt Selection Register

The Interrupt Selection Register (base + 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 HP Command 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. HP Command 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. HP Command 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 (default value)	Reserved												INT 1	VXI I nterrupt 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 HP E2251A 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 (D0 - D7) 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 (D0 - D7). A cause/status byte is also placed on the data bus (D15 - D8)

Additional Registers

M-Modules may use additional A16 space registers. They may also have memory space mapped into A24 Memory Space. Refer to the individual M-Module User Manual for details on register addressing and use.

Register-Based Programming Examples

This section provides several examples demonstrating programming M-Modules installed in the Carrier.

Note Do not do register writes if you are also controlling the modules by a high level language such as SCPI. This is because the high level language driver will not know the instrument state and an interrupt may occur causing the driver and/or command module to fail.

Example 1: Reading the M-Module ID Registers

The following example program reads and prints the four VXI registers (ID, Device Type, Status, A24 Offset) of an M-Module. The program was developed with the ANSI C language using the HP 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 HP SICL Library, the HP VISA extensions, and an HP 82340 or 82341 HP-IB module installed and properly configured in your PC. An HP E1406 Command Module provides direct access to the VXI backplane.

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

#define INSTR_ADDR "VXI::24::INSTR"

void main()
{
   ViStatus errStatus;                    /*Status from each VISA call*/
   ViSession viRM;                        /*Resource mgr. session */
   ViSession m_mod                        /* M-module 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);

/* 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;
   }
```

Example 2: Closing a Channel Relay

The following example program closes channels 1 and 3 and then opens (resets) the channels on an HP E2272A Dual 8-to1 Relay Multiplexer M-Module. The program was developed with the ANSI C language using the HP 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 HP SICL Library, the HP VISA extensions, and an HP 82340 or 82341 HP-IB module installed and properly configured in your PC. An HP E1406 Command Module provides direct access to the VXI backplane.

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

#define INSTR_ADDR "VXI::24::INSTR"

void main()
```

```
{
   ViStatus errStatus;                    /*Status from each VISA call*/
   ViSession viRM;                        /*Resource mgr. session */
   ViSession m_module                     /* M-module 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;}

   /* set the Driver Power Enable Bit (bit 3) in the Card Control Register */
   errStatus = viOut16 (m_mod,VI_A24_SPACE,0x02,0x08);
   if(VI_SUCCESS > errStatus){
      printf("ERROR: viOut16() returned 0x%x\n",errStatus);
      return errStatus;}

   /* Close Channels 0 and 2 (bit values 1 and 4 respectively) */
   errStatus = viOut16 (m_mod,VI_A24_SPACE,0x10,0x05);
   if(VI_SUCCESS > errStatus){
      printf("ERROR: viOut16() returned 0x%x\n",errStatus);
      return errStatus;}

   /* Open (Reset) channels in Row 0) */
   errStatus = viOut16 (m_mod,VI_A24_SPACE,0x12,0x00);
   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;
   }
```

Appendix A
HP E2251A M-Module Carrier Specifications

M-Module Standard Compliance

A08, A16, D08, D16, INTA, INTB, INTC

Allows standard M-Modules to operate in a VXI system as Register-based devices.

General Capabilities

Size:

- VXI C-Size

Slots required in VXI Mainframe:

- 1

VXI Connectors:

- P1, P2

Number of M-Modules:

- Up to 6 (with field wiring to the two internal modules M4 and M5, only five M-Modules can be installed on the Carrier.)

Watts/Slot:

- 50W maximum.

Caution **When calculating total power for the Carrier (for cooling), you must consider the power dissipated from each switching M-Module. For example, the relay contacts on the HP E2274A 4-Channel Power Relay M-Module are rated for a maximum of 5.0 Amps DC. With an assumed contact resistance of 0.25Ω, each relay can dissipate a maximum of 6.25 Watts ($I^2R = 6.25$), or a total of 25 Watts per module. The HP E2251A can accommodate a maximum of 50 Watts, so only two HP E2274A Modules can be used under those conditions. More modules can be installed if fewer relays are closed at one time, less current is switched, etc. Do not exceed the maximum cooling capacity of the HP E2251A.**

Cooling:

- 0.2 ΔP mm H_2O

Air Flow:

- 4.0 Liter/Sec

Current Required by Carrier:

- 5Vdc @ 0.3A (does not include power required by M-Modules)
  ±12Vdc @ 0.0A (does not include power required by M-Modules)

Symbols

A

B

C

D

E

F

G

H

W