Model 3000-155(A)
(4x4) Coaxial Matrix (155)
Dual (4x4) Coaxial Matrix (155A)
90401650

All technical data and specifications in this publication are subject to change without prior notice and do not represent a commitment on the part of Giga-tronics, Incorporated.

Warranty

Giga-tronics Series 3000 Switching Modules are warranted against defective materials and workmanship for three years from date of shipment, or as detailed in the warranty section of this manual. Giga-tronics will, at its option, repair or replace products that are proven defective during the warranty period. This warranty DOES NOT cover damage resulting from improper use, nor workmanship other than Giga-tronics service. There is no implied warranty of fitness for a particular purpose, nor is Giga-tronics liable for any consequential damages. Specification and price change privileges are reserved by Giga-tronics.

CONTACT INFORMATION

Giga-tronics, Incorporated
4650 Norris Canyon Road
San Ramon, California 94583
Telephone: 800.726.4442 (only within the United States)
925.328.4650
Fax: 925.328.4700
On the Internet: www.gigatronics.com

Regulatory compliance information

This product complies with the essential requirements of the following applicable European Directives, and carries the CE mark accordingly.

89/336/EEC and 73/23/EEC	EMC Directive and Low Voltage Directive
EN61010-1 (1993)	Electrical Safety
EN61326-1 (1997)	EMC – Emissions and Immunity
Manufacturer's Name:	**Manufacturer's Address**
Giga-tronics, Incorporated	4650 Norris Canyon Road San Ramon, California 94583 U.S.A.
Type of Equipment:	**Model Series Number**
Switching Module	3000-155(A)

Declaration of Conformity on file. Contact Giga-tronics at the following;

Giga-tronics, Incorporated
4650 Norris Canyon Road
San Ramon, California 94583
Telephone: 800.726.4442 (only within the United States)
925.328.4650
Fax: 925.328.4700

Record of Changes to This Manual

Use the table below to maintain a permanent record of changes to this document. Corrected replacement pages are issued as Technical Publication Change Instructions (TPCI). When you are issued a TPCI, do the following:

1. Insert the TPCI at the front of the manual binder.
2. Remove the pages from the manual binder that are noted in the TPCI.
3. Replace the page(s) removed in the previous step with the corrected page(s).
4. Record the changes in the table below.

TPCI Number	TPCI Issue Date	Date Entered	Comments

Revision History			
Revision	**Description of Change**	**Chg Order #**	**Approved By**
A	Initial Release		
B	Updated		
C	Updated 8/10		DT
D	Reformatted 3/12		RCW

Contents

Chapter 1 **Introduction**

1.1 Safety and Manual Conventions

This manual contains conventions regarding safety and equipment usage as described below.

1.1.1 Product Reference

Throughout this manual, the term “Common Core Switching Platform, Series 8800” refers to all models of within the series, unless otherwise specified.

1.1.2 Personal Safety Alert

WARNING: Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

1.1.3 Equipment Safety Alert

CAUTION: Indicates a situation which can damage or adversely affect the product or associated equipment.

1.1.4 Notes

Notes are denoted and used as follows:

NOTE: Highlights or amplifies an essential operating or maintenance procedure, practice, condition or statement.

1.1.5 Electrical Safety Precautions

Any servicing instructions are for use by service-trained personnel only. To avoid personal injury, do not perform any service unless you are qualified to do so.

For continued protections against fire hazard, replace the AC line fuse only with a fuse of the same current rating and type. Do not use repaired fuses or short circuited fuse holders.

Chapter 2 **Configuration Table**

Model 3000-155(A) (with **SMB** connectors)

90401650-001	Top Assembly
PL90401650-001	Parts List , Top Assembly
85005200-002	PWA
PL85005200-002	Parts List, PWA
SCH85005200	Schematic, PWA

Model 3000-155(A) (with **SMC** connectors)

90401650-001	Top Assembly
PL90401650-001	Parts List , Top Assembly
85005200-004	PWA
PL85005200-004	Parts List, PWA
SCH85005200	Schematic, PWA

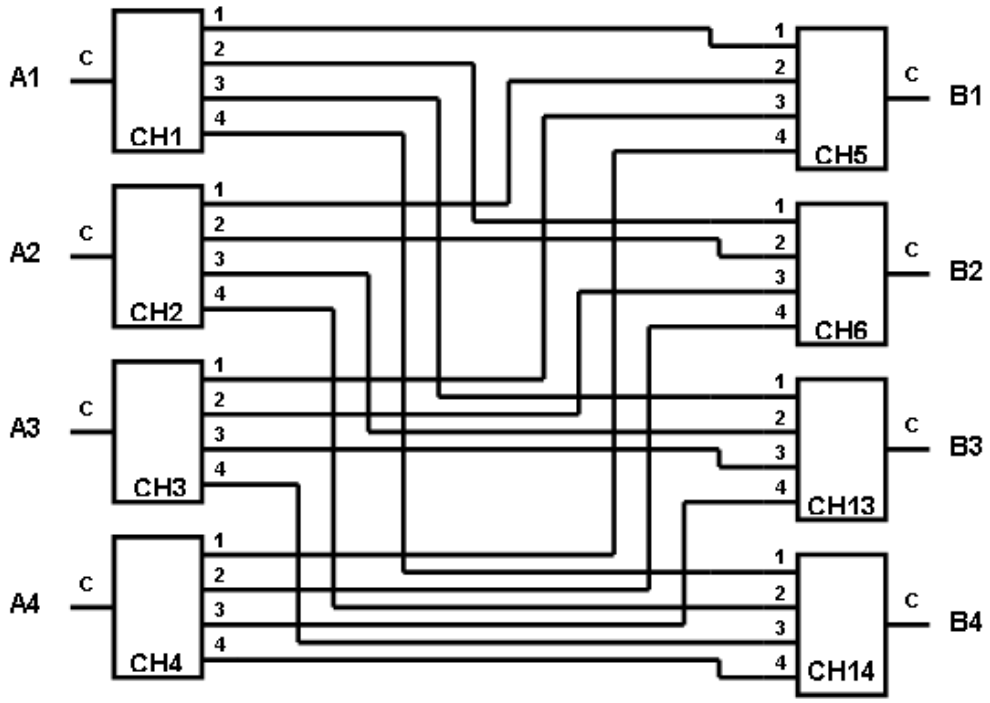
Chapter 3 Functional Description

3.1 Introduction

This manual provides the necessary information for the operation and maintenance of the Model 3000-155(A) Single 4X4 Matrix and Dual 4X4 (Option -01) Matrix VXI Module.

3.2 General Description

The 3000-155(A) is a single-wide VXI Module which provides one or two (2) completely independent 4X4 Matrices. Each Matrix is composed of eight (8) 1X4 Switch Channels which are interconnected as shown in figure 1. Each 1X4 Switch Channel is bi-directional and can have it's Common Path connected to only one of it's four associated Paths (1-4) at a time. This configuration allows any one of the four "A" Channels to be connected to any one of the four "B" Channels and vice versa. The same is true for the "C" and "D" Channels of the Dual 4X4 Matrix. From one (1) up to four (4) Paths can be completed at once in a 4X4 Matrix. For example you could complete Paths: A1<->B2, A2<->B3, A3<->B4, and A4<->B1 simultaneously.



The 3000-155(A) Single/Dual 4X4 Matrix is a single Motherboard VXI Module which provides a VXI register based interface as well as Single or Dual 4X4 Matrices.

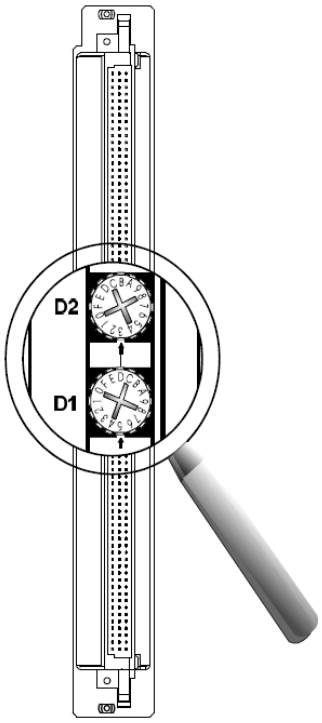
The 3000-155(A) is a register based device and supports VXIbus register maps. All controls to the 3000-155(A) are done through registers. All registers can be accessed with the use of slot 0 computers, host computers with VXI-MXI, or host computers with GPIB and GPIB-VXI slot 0 controllers. The 3000-155(A) Single/Dual 4X4 Matrix is not a message based device and does not support VXIbus communication protocols.

Chapter 4 Controls and Indicators

The following controls and indicators are provided to select and display the functions of the ASCOR 3000-155(A) Module's operating environment.

4.1 VXI Logical Address

The Logical Address Switch is dual circular switches, D1 and D2 which are located at the rear of the module. The address can be set to any value between 1 and 255 (decimal) or 1 and FF (hexadecimal), (address 0 is reserved for the resource manager). However, the Module fully supports Dynamic Configuration as defined in ***Section F of the VXI specification,*** address 255 (FF) should be selected only if the Resource Manager also supports Dynamic Configuration.



4.2 LEDs

The following LEDs are visible at the Module's front panel to indicate the

status of the module's operation:

4.2.1 "BUS" LED

This green color LED is normally off and will flash on when the module is addressed by the system.

4.2.2 "PWR" LED

This red color LED is normally on when the Module is Powered up.

Chapter 5 Internal Settings

The following items are inside the module and can be reached by removing the side cover.

5.1 Fuse

The ASCOR VXI 3000-155(A) uses a 10 Amp fuse in the +5 Volt line and is located on the Mother Board (MB) assembly.

5.2 VXI_{bus} Interrupt Level Selection

The VXIbus interrupt level is set with three bits in the “3Eh” register.

See the section on “A16 ADDRESS SPACE REGISTER DESCRIPTION”.

The interrupt level is factory set to “no interrupt”.

Chapter 6 **Specifications**

Electrical Specifications

Power

+5V (*0.85) amp (quiescent), 2.25 Amp (Full Load)*

Relay

Switching Voltage	24 Vdc Maximum
Switching Current	1.0 Amps Maximum
Power Rating	10 Watts Maximum
Life Expectancy (24V @1A)	10^5 Operations
Static Contact Resistance	100mΩ
Operating Time (Including Bounce)	12 mSec Maximum
Release Time	6.5 mSec Maximum

Signal

Insertion Loss	
<100 Mhz,	0.5 dB
100 - 500 Mhz	<1.2 dB
0.5 - 1.0 GHz	<1.8 dB
1.0 - 1.3 GHz	<2.8 dB
Crosstalk (Adjacent Paths)	
<100 Mhz	<-70 dB
100 - 500 Mhz	<-50 dB
0.5 - 1.3 GHz	<-37 dB
VSWR	
<100 Mhz	<1.15:1 dB
100 - 500 Mhz	<1.40:1 dB
0.5 - 1.3 GHz	<1.50:1 dB

Mechanical Specifications

Weight: 3 lbs. 4oz.

Dimensions: Single-wide “C” size VXI (13.4" x 9.2")

Connectors:

Single/Dual 4X4 Matrix (A1 - D4)	SMB Straight Male Bulkhead Jack, RG-316 Flexible Cable Crimp type, Nickel Plated, AEP 2003-7571-003.
Mate:	SMB Straight Female Plug, RG-316 Flexible Cable Crimp type, Nickel Plated, AEP 2002-7571-003.

Installation Kits:

Single 4X4 Matrix Installation Kit:	Installation Kit, 89800370-001, Available on request (Optional). Includes a full complement of mating connectors.
Dual 4X4 Matrix Installation Kit:	Installation Kit, 89800370-002, Available on request (Optional). Includes a full complement of mating connectors.

Environmental Specifications

Temperature:	
Operating:	0º to 55ºC
Storage:	- 40º to 75ºC
Relative Humidity:	
Operating:	0 to 90% non-condensing
Storage:	0 to 95% non-condensing

Chapter 8 **Register Map**

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix Module has VXI Device Registers located in the A16 Address Space. The VXI Specification defines 32 VXI Device Registers and they are all 16 bits wide. The first 4 registers are VXI Configuration Registers. The next 12 registers are VXI Device Class Dependent Registers. The last 16 registers are VXI Device Dependent Registers.

The 3000-155(A) supports 5 of the 32 VXI Device Registers, four in VXI Configuration Registers and one in VXI Device Dependent Register. All other registers are not supported. Some Device Registers must be written only by a Resource Manager. Table 2 lists the VXI Device Registers.

General Information

Register-based Architecture:

Allows VXI bus speeds to control I/O actions, resulting in much improved operating speeds when compared to message-based modules.

A16/A24 Address Space:

This module utilizes both the A16 and A24 Address space for mapping I/O functions. The A16 Address space contains the VXIbus configuration registers which are defined by the VXIbus specification for all VXIbus devices. The A24 Address space contains the 1.3Ghz 1X4 Switch Relay Registers.

16/32 Data Bus operation:

ASCOR's VXIMAX 16/32 VXIbus Interface allows this module to operate in either 16-bit or 32-bit data bus modes. The A16 Address Space is addressed in 16-bit mode only. The A24 Address Space can be addressed in 16 or 32-bit mode.

Static and Dynamic:

The 3000-155(A) supports both Static Configuration and Dynamic Configuration: Configuration of the Logical Address. In Static Configuration mode the Logical Address of the Module is manually set, and cannot be changed by the Resource Manager. In Dynamic Configuration mode the Logical Address is determined by the Resource Manager based on other devices in the system.

Register-Based Architecture

- A16 / A24 Addressing
- 16 / 32-Bit Data Bus Operation
- Static / Dynamic Logical Address Configuration

Memory Map

A16 Space

Offset (hex)	This offset is added to the A16 Base Address of the module. The A16 Base Address for the 3000-155(A) 1.3Ghz 1X4 Switch is
equal	to the VXIbus logical address assigned to the 3000-155(A) shifted left six times and ORed with hex C000. These registers reside in the VXI Interface circuitry on the Motherboard (85002250).
00	VXIbus ID Register
02	Device Type Register
04	VXIbus Status/Control Register
06	Offset Register
3E	ASCOR Relay Control Register

A24 Space

Offset (hex)	This offset is added to the A24 Base Address of the module. The A24 Base Address can be derived from the value stored in the Offset Register (A16 Address Space, 06h). Take the 8 Most Significant Bits of the Offset Register and map them into the 8 Most Significant bits of the A24 Base Address. The 8 Least Significant Bits are set to zeros.
8000	Mother Board – 4X4 Matrix (85002250)

VXI Device Register Description

The ASCOR 3000-155(A) Twelve (12) Channel 1.3 Ghz 1x4 Switch Module has VXI Device Registers located in the A16 address space. The VXI Specification defines 32 VXI Device Registers and they are all 16 bits wide. The first 4 registers are VXI Configuration Registers. The next 12 registers are VXI Device Class Dependent Registers. The last 16 registers are VXI Device Dependent Registers. The 3000-155(A) supports 5 of the 32 VXI Device Registers: four in VXI Configuration Registers, and one in VXI Device Dependent Register. All other registers are not supported.

VXI Device Registers for ASCOR 3000-155 (A)	
VXI Device Dependent Registers	
3Eh	ASCOR Control Register
3Ch	Register Not Used
3Ah	Register Not Used
38h	Register Not Used
36h	Register Not Used
34h	Register Not Used
32h	Register Not Used
2Eh	Register Not Used
2Ch	Register Not Used
2Ah	Register Not Used
28h	Register Not Used
26h	Register Not Used
24h	Register Not Used
22h	Register Not Used
20h	Register Not Used
VXI Device Class Dependent Registers	
1Eh	Register Not Used
1Ch	Register Not Used
1Ah	Register Not Used
18h	Register Not Used
16h	Register Not Used
14h	Register Not Used
12h	Register Not Used
10h	Register Not Used
0Eh	Register Not Used
0Ch	Register Not Used
0Ah	Register Not Used
08h	Register Not Used
VXI Configuration Registers	
06h	Offset Register
04h	Status / Control Register
02h	Device Type Register
00h	ID / Logical Address Register

VXI Device Register Description (Continued)

VXI Configuration Registers

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Device Class		Address Space		Manufacturer ID											

Offset Description 00h ID Register (read) / Logical Address Register (write)

A read of this 16-bit register provides information about the 3000-155(A) Module's configuration.

(Bits 15-14) Device Class: This field indicates the classification of the VXIbus device.
00b = Memory
01b = Extended
10b = Message Based
11b = Register Based (ASCOR VXI Module)

(Bits 13-12) Address Space: This field indicates the addressing mode(s) of the device's operational registers.
00b = A16/A24
01b = A16/A32
10b = RESERVED
11b = A16 Only

(Bits 11-0) Manufacturer ID: This field uniquely identifies the manufacturer of the device.
FB5h = ASCOR

For the 3000-155(A) Module, the register should read back a value of CFB5h.

A write to this 16-bit register is provided for Dynamic Configuration protocol. This register should only be written to by a resource manager. ***Do not write to this register***.

VXI Device Register Description (Continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Required Memory				Model Code											

Offset Description 02h Device Type (read/write)

A read of this 16-bit register provides information about the 3000-155(A) Module's Device Type. This register indicates how much VMEbus memory is required by the VXI module, as well as the Manufacture's unique model code.

(Bits 15-12) Required Memory: This field contains the value used for determining the A24 or A32 memory space resident on the device

7h = 64K bytes in A24 Address Space

Fh = 64K bytes in A32 Address Space

(Bits 11-0) Model Code: This field contains the manufacturer's unique module identifier.

D10h = The ASCOR Model Code for the 3000-155(A)

For the 3000-155(A) Module, the register should read back a value of 7D10h. A write to this 16-bit register is provided for VXIbus definition. ***Do not write to this register.***

VXI Device Register Description (Continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A24/ A32	MOD ID	Device Dependent (Not used)										Ready	Passed	Sys fail	Dev. Rst

Offset **Description**

04h **Status Register (read) / Control Register (write)**

A read of this 16-bit register provides information about the 3000-155(A) status.

(Bit 15) A24/A32 Active: This bit indicates the accessibility of A24 or A32 registers.
1b = A24 or A32 Address Space Active
0b = A24 or A32 Address Space not Active (for A16 only devices)

(Bit 14) MODID: This bit indicates if the device is selected via the P2 MODID line.
1b = Device is not selected via the P2 MODID line. Used by the resource manager during Dynamic Configuration.

(Bits 13-4) Device Dependent (Not used)

(Bit 3) Ready: This bit indicates if the device is ready to accept operational commands.
1b = Device is ready after power-on initialization sequence

(Bit 2) Passed: This bit indicates if the power-on self test has successfully completed.
1b = Device does not support power-on self test (always pass)

(Bits 1-0) Device Dependent
00b = State of the corresponding bits of the Control register

For the 3000-155(A) Module, the register should read back a value of FFFCh.

A write to this 16-bit register causes specific actions to be executed.

(Bit 15) A24/A32 Enable: This bit enables or disables A24 / A32 VMEbus registers.
1b = Enables A24 or A32 VMEbus registers. This bit must always remain a one after being set to one by the resource manager.
0b = Disables A24 or A32 VMEbus registers. This bit must always remain a zero after being cleared to zero by the resource manager. (for A16 only devices)

(Bits 14-2) Device Dependent (Not used)

(Bit 1) Sysfail Inhibit: This bit controls the device's ability to drive the SYSFAIL line.
0b = Always set to zero (Sysfail not inhibited)

(Bit 0) Device Reset: This bit controls the state of the device.
1b = Reset the device to power-on state.
0b = Normal operational mode

VXI Device Register Description (Continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset Value								Not Used							

06h **Offset Register (read/write)**

A read of the 16-bit register provides information for calculating the base address of the 3000-155(A) A24 operational registers.

(Bits 15-0) Offset Value: This field is used for calculating the A24 Base Address.

To obtain the A24 base address for the 3000-155(A), take the 8 Most Significant Bits of the Offset register and map them to the 8 most significant bits of the A24 Base Address. All other bits in the A24 Base Address are set to zeroes. For more detail refer to (Section 3, Page 16) ***"Miscellaneous Questions and Answers"*** in the Programming Guide.

A write to this 16-bit register is provided for Dynamic Configuration protocol. This register should only be written to by a resource manager. ***Do not write to this register***.

VXI Device Register Description (Continued)

VXI Device Class Dependent Registers

The ASCOR 3000-155(A) does not use nor provide any of the 12 Device Class Dependent Registers.

Offset	Description
08h	Not Used
0Ah	Not Used
0Ch	Not Used
0Eh	Not Used
10h	Not Used
12h	Not Used
14h	Not Used
16h	Not Used
18h	Not Used
1Ah	Not Used
1Ch	Not Used
1Eh	Not Used

VXI Device Register Description (Continued)

VXI Device Dependent Registers for the 3000-155(A)

The VXI Specification defines 16 Device Dependent Registers in the A16 address space following the Device Class Dependent Register space. Each register is 16 bits wide. The first 15 registers are not used nor provided by the ASCOR 3000-155(A) Module, only the last register is used.

Offset	Description
20h	Not Used
22h	Not Used
24h	Not Used
26h	Not Used
28h	Not Used
2Ah	Not Used
2Ch	Not Used
2Eh	Not Used
30h	Not Used
32h	Not Used
34h	Not Used
36h	Not Used
38h	Not Used
3Ah	Not Used
3Ch	Not Used

VXI Device Register Description (Continued)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved														Reg Bit	Coil Ena

3Eh **ASCOR Control Register (read/write)**

This 16-bit register provides the module control status.

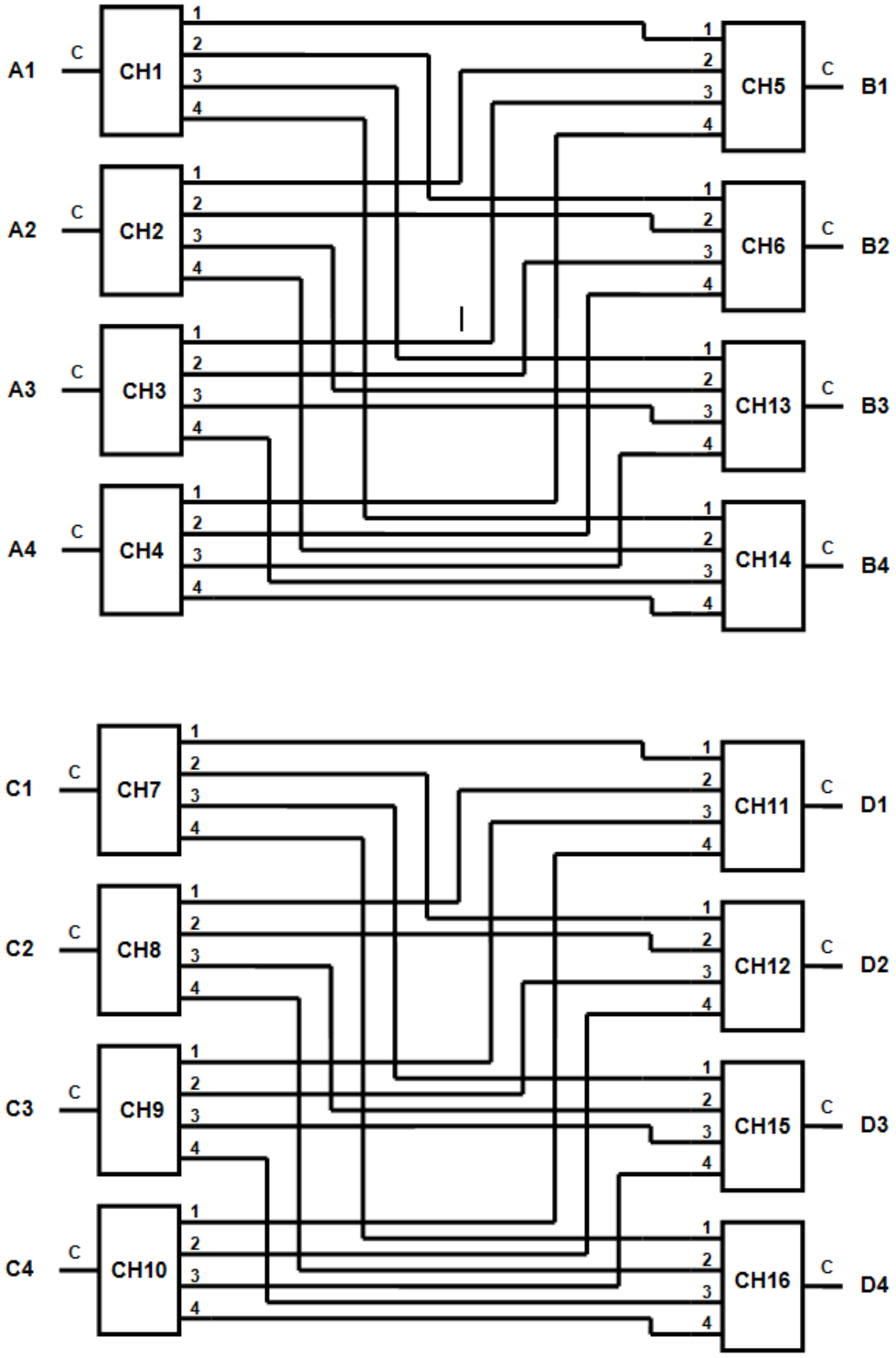
(Bits 15-2) Reserved
0h = Should always be set to zeroes

(Bit 1) Reg Bit: This bit indicates the device's readback mode.
0b = Relay coil state readback is enabled
1b = Data register state readback is enabled

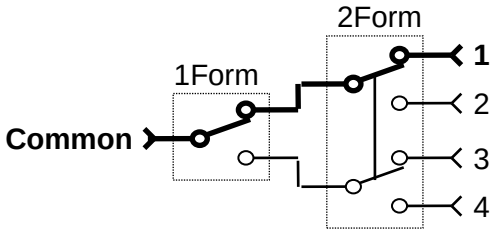
(Bit 0) Coil Enable This bit indicates the device's coil driver state.

0b = Relay coil driver is enabled
1b = Relay coil driver is disabled

Circuit Description



The 3000-155(A) provides one or two (2) completely independent 4X4 Matrices. Each Matrix is composed of eight (8) 1X4 Switch Channels which are interconnected as shown in figure 1. Each 1X4 Switch Channel is bi-directional and can have it's Common Path connected to only one of it's four associated Paths (1-4) at a time. This configuration allows any one of the four "A" Channels to be connected to any one of the four "B" Channels and vice versa. The same is true for the "C" and "D" Channels of the Dual 4X4 Matrix. From one (1) up to four (4) Paths can be completed at once in a 4X4 Matrix. For example you could complete Paths: A1<->B2, A2<->B3, A3<->B4, and A4<->B1 simultaneously.



1X4 Switch Channel with Path #1 Connected to Common.

A path is configured by energizing or de-energizing the Relays. For instance to complete the connection for a 1X4 Switch Channel Path #2 to the Common, the 2FormC Relay must be energized, while keeping the 1FormC Relay de-energized as shown.

2Form
1Form
Common
1
2
3
4

1X4 Switch Channel with Path #2 Connected to Common.

To complete the connection from Path #3 to the Common, the 1FormC Relay must be energized, while keeping the 2FormC Relay de-energized as shown.

2Form
1Form
Common
1
2
3
4

1X4 Switch Channel with Path #3 Connected to Common.

Finally, to complete the connection from Path #4 to the Common, both the 2FormC and 1FormC Relay must be energized as shown in figure 6.

2Form
1Form
Common
1
2
3
4

1X4 Switch Channel with Path #4 Connected to Common.

Relay energizing and de-energizing is accomplished by setting or clearing the associated bits in the Relay Control Registers (0=Non-Energized, 1=Energized). There are two (2) 16-bit Relay Control Registers which are located in the A24 Address space.

16-bit Offset: 0h 32-bit
Offset: 0h

CHANNEL 8 (C2)		CHANNEL 7 (C1)		CHANNEL 6 (B2)		CHANNEL 5 (B1)		CHANNEL 4 (A4)		CHANNEL 3 (A3)		CHANNEL 2 (A2)		CHANNEL 1 (A1)	
1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

MSB LSB

16-bit Relay Control Register for 1X4 Channels 1 – 8

16-bit Offset: 2h 32-bit
Offset: 0h

CHANNEL 16 (D4)		CHANNEL 15 (D3)		CHANNEL 14 (B4)		CHANNEL 13 (B3)		CHANNEL 12 (D2)		CHANNEL 11 (D1)		CHANNEL 10 (C4)		CHANNEL 9 (C3)	
1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C	1 Form C	2 Form C
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16

MSB LSB

16-bit Relay Control Register for 1X4 Channels 9 -16.

From one to eight 1x4 Channels can be configured with a single 16-bit write operation to either register. From one to all sixteen 1x4 Channels can be configured with a single 32-bit write operation to both registers. Care must be taken when values are written in order to prevent unintended Relay energizing or de-energizing. In order to preserve the states of the 4X4 Matrices that you do not want to alter, perform the following sequence of operations:

1. Read the Relay Control Register and save the register contents.
2. Clear the desired Relay bits you intend to program by ANDing the saved copy of the register contents with the associated Channel Clear Masks from Table 2.
3. Set the desired Relay bits you intent to program by ORing the cleared register contents with the associated Channel Set Masks from Table 2.

4. Write the new value back to the Relay Control Register.

Using this technique, each 1X4 Switch Channel can be independently programmed to one of the four different path configurations discussed above. Table 2 lists the 16 and 32-bit Address Offsets As well as the Clear and Set Mask data for each Channel.

PATH	ENERGIZED RELAYS							
A1 <-> B1	-		-		-		-	
A1 <-> B2	Ch 1	2Form C	-		-		-	
A1 <-> B3	-		Ch 1	1Form C	-		-	
A1 <-> B4	Ch 1	2Form C	Ch 1	1Form C	-		-	
A2 <-> B1	-		-		Ch 5	2Form C	-	
A2 <-> B2	Ch 2	2Form C	-		Ch 6	2Form C	-	
A2 <-> B3	-		Ch 2	1Form C	Ch 13	2Form C	-	
A2 <-> B4	Ch 2	2Form C	Ch 2	1Form C	Ch 14	2Form C	-	
A3 <-> B1	-		-		-		Ch 5	1Form C
A3 <-> B2	Ch 3	2Form C	-		-		Ch 6	1Form C
A3 <-> B3	-		Ch 3	1Form C	-		Ch 13	1Form C
A3 <-> B4	Ch 3	2Form C	Ch 3	1Form C	-		Ch 14	1Form C
A4 <-> B1	-		-		Ch 5	2Form C	Ch 5	1Form C
A4 <-> B2	Ch 4	2Form C	-		Ch 6	2Form C	Ch 6	1Form C
A4 <-> B3	-		Ch 4	1Form C	Ch 13	2Form C	Ch 13	1Form C
A4 <-> B4	Ch 4	2Form C	Ch 4	1Form C	Ch 14	2Form C	Ch 14	1Form C

4X4 Matrix Relay Configuration.

CHANNEL	PATH TO COMMON	16-bit ADDRESS OFFSET	16-bit Clear Mask	16-bit Set Mask	32-bit ADDRESS OFFSET	32-bit Clear Mask	32-bit Set Mask
1	1	8000h	FFFCh	0000h	8000h	FFFFFFFCh	00000000h
1	2	8000h	FFFCh	0001h	8000h	FFFFFFFCh	00000001h
1	3	8000h	FFFCh	0002h	8000h	FFFFFFFCh	00000002h
1	4	8000h	FFFCh	0003h	8000h	FFFFFFFCh	00000003h
2	1	8000h	FFF3h	0000h	8000h	FFFFFFF3h	00000000h
2	2	8000h	FFF3h	0004h	8000h	FFFFFFF3h	00000004h
2	3	8000h	FFF3h	0008h	8000h	FFFFFFF3h	00000008h
2	4	8000h	FFF3h	000Ch	8000h	FFFFFFF3h	0000000Ch
3	1	8000h	FFCFh	0000h	8000h	FFFFFFCFh	00000000h
3	2	8000h	FFCFh	0010h	8000h	FFFFFFCFh	00000010h
3	3	8000h	FFCFh	0020h	8000h	FFFFFFCFh	00000020h
3	4	8000h	FFCFh	0030h	8000h	FFFFFFCFh	00000030h
4	1	8000h	FF3Fh	0000h	8000h	FFFFFF3Fh	00000000h
4	2	8000h	FF3Fh	0040h	8000h	FFFFFF3Fh	00000040h
4	3	8000h	FF3Fh	0080h	8000h	FFFFFF3Fh	00000080h
4	4	8000h	FF3Fh	00C0h	8000h	FFFFFF3Fh	000000C0h
5	1	8000h	FCFFh	0000h	8000h	FFFFFCFFh	00000000h
5	2	8000h	FCFFh	0100h	8000h	FFFFFCFFh	00000100h
5	3	8000h	FCFFh	0200h	8000h	FFFFFCFFh	00000200h
5	4	8000h	FCFFh	0300h	8000h	FFFFFCFFh	00000300h
6	1	8000h	F3FFh	0000h	8000h	FFFFF3FFh	00000000h
6	2	8000h	F3FFh	0400h	8000h	FFFFF3FFh	00000400h
6	3	8000h	F3FFh	0800h	8000h	FFFFF3FFh	00000800h
6	4	8000h	F3FFh	0C00h	8000h	FFFFF3FFh	00000C00h
7	1	8000h	CFFFh	0000h	8000h	FFFFCFFFh	00000000h
7	2	8000h	CFFFh	1000h	8000h	FFFFCFFFh	00001000h
7	3	8000h	CFFFh	2000h	8000h	FFFFCFFFh	00002000h
7	4	8000h	CFFFh	3000h	8000h	FFFFCFFFh	00003000h
8	1	8000h	3FFFh	0000h	8000h	FFFF3FFFh	00000000h
8	2	8000h	3FFFh	4000h	8000h	FFFF3FFFh	00004000h
8	3	8000h	3FFFh	8000h	8000h	FFFF3FFFh	00008000h
8	4	8000h	3FFFh	C000h	8000h	FFFF3FFFh	0000C000h
9	1	8002h	FFFCh	0000h	8000h	FFFCFFFFh	00000000h
9	2	8002h	FFFCh	0001h	8000h	FFFCFFFFh	00010000h
9	3	8002h	FFFCh	0002h	8000h	FFFCFFFFh	00020000h
9	4	8002h	FFFCh	0003h	8000h	FFFCFFFFh	00030000h
10	1	8002h	FFF3h	0000h	8000h	FFF3FFFFh	00000000h
10	2	8002h	FFF3h	0004h	8000h	FFF3FFFFh	00040000h
10	3	8002h	FFF3h	0008h	8000h	FFF3FFFFh	00080000h
10	4	8002h	FFF3h	000Ch	8000h	FFF3FFFFh	000C0000h
11	1	8002h	FFCFh	0000h	8000h	FFCFFFFFh	00000000h
11	2	8002h	FFCFh	0010h	8000h	FFCFFFFFh	00100000h
11	3	8002h	FFCFh	0020h	8000h	FFCFFFFFh	00200000h
11	4	8002h	FFCFh	0030h	8000h	FFCFFFFFh	00300000h
12	1	8002h	FF3Fh	0000h	8000h	FF3FFFFFh	00000000h
12	2	8002h	FF3Fh	0040h	8000h	FF3FFFFFh	00400000h
12	3	8002h	FF3Fh	0080h	8000h	FF3FFFFFh	00800000h
12	4	8002h	FF3Fh	00C0h	8000h	FF3FFFFFh	00C00000h
13	1	8002h	FCFFh	0000h	8000h	FCFFFFFFh	00000000h
13	2	8002h	FCFFh	0100h	8000h	FCFFFFFFh	01000000h
13	3	8002h	FCFFh	0200h	8000h	FCFFFFFFh	02000000h
13	4	8002h	FCFFh	0300h	8000h	FCFFFFFFh	03000000h
14	1	8002h	F3FFh	0000h	8000h	F3FFFFFFh	00000000h
14	2	8002h	F3FFh	0400h	8000h	F3FFFFFFh	04000000h
14	3	8002h	F3FFh	0800h	8000h	F3FFFFFFh	08000000h
14	4	8002h	F3FFh	0C00h	8000h	F3FFFFFFh	0C000000h
15	1	8002h	CFFFh	0000h	8000h	CFFFFFFFh	00000000h
15	2	8002h	CFFFh	1000h	8000h	CFFFFFFFh	10000000h
15	3	8002h	CFFFh	2000h	8000h	CFFFFFFFh	20000000h
15	4	8002h	CFFFh	3000h	8000h	CFFFFFFFh	30000000h
16	1	8002h	3FFFh	0000h	8000h	3FFFFFFFh	00000000h
16	2	8002h	3FFFh	4000h	8000h	3FFFFFFFh	40000000h

16	3	8002h	3FFFh	8000h	8000h	3FFFFFFFh	80000000h
16	4	8002h	3FFFh	C000h	8000h	3FFFFFFFh	C0000000h

32-bit Programming Description:

Example #1 As an 32-bit example, let's assume that the 3000-155(A) Single/Dual 4X4 Matrix Module A24 Base Address is 200000h. By adding the 3000-155(A)'s offset from the Base Address of 8000h, we can perform a 32-bit read of the Relay Control Registers by executing a 32-bit Read from 208000h.

OPERATION	32-bit Read
ADDRESS	0x208000h
DATA	0 0 0 0 0 0 0 0 h

If we have just powered up or reset the 3000-155(A) the contents of the Relay Control Registers will read back 00000000h, indicating that all of the Relays are de-energized. This means that all 1X4 Channels are configured with Path #1 connected to Common.

Example #2 Now let's configure Channel 4 so that Path #2 is connected to Common. You can change one Channel by modifying just the Relay Control bits associated with the desired Channel. First we perform a 32-bit read from 208000h and get back 00000000h. We'll call this value CH1_CH16 as it represents the current configuration of all 1X4 Channels. Then we clear the Channel 4 Relay bits by ANDing the Channel 4 Clear Mask value from Table 2 with the CH1_CH16 value.

OPERATION	32-bit AND
CH1_CH16	0 0 0 0 0 0 0 0 h
CLEAR MASK	F F F F F F 3 F h
RESULT	0 0 0 0 0 0 0 0 h

Next we prepare to energize the desired Channel 4 Relays by setting the appropriate Channel 4 Relay Control Register Bits. This can be accomplished by ORing the Channel 4 Path #2 Set Mask value from Table 2 with our CH1_CH16 value.

OPERATION	32-bit OR
CH1_CH16	0 0 0 0 0 0 0 0 h
CH4 PATH #2 SET MASK	0 0 0 0 0 0 4 0 h
RESULT	0 0 0 0 0 0 4 0 h

The result has only the 2FormC Relay bit set for Channel 4. Note that none of the other bits have been modified. Thus, when we perform a 32-bit write of this value to the Relay Control Register (208000h) we will energize only the 2FormC Relay of Channel 4 which will connect Path #2 to Common.

Example #3 Next Let's connect Path #3 on Channel 4 and Channel 11. First we perform a 32-bit read from 208000h and get back 00000040h. Again, we save this as our CH1_CH16 value. Then, starting with Channel 4, we clear the Relay bits by ANDing the Channel 4 Clear Mask Value with our CH1_CH16 value. We repeat this ANDing process for Channel 11, using the proper Table 2 value.

OPERATION	32-bit AND
CH1_CH16	0 0 0 0 0 0 4 0 h
CH4 CLEAR MASK	F F F F F F 3 F h
RESULT	0 0 0 0 0 0 0 0 h

↓

OPERATION	32-bit AND
CH1_CH16	0 0 0 0 0 0 0 0 h
CH11 CLEAR MASK	F F C F F F F F h
RESULT	0 0 0 0 0 0 0 0 h

32-bit Programming Description: (Continued)

Next, again starting with Channel 4 we prepare to energize the Relays by ORing the Channel 4 Set Mask value from Table 2 with our CH1_CH16 value. We then repeat this ORing process for Channel 11, using the proper Table 2 values.

OPERATION	32-bit OR
CH1_CH16	0 0 0 0 0 0 0 0 h
CH4 PATH #3 SET MASK	0 0 0 0 0 0 8 0 h
RESULT	0 0 0 0 0 0 8 0 h

↓

OPERATION	32-bit OR
CH1_CH16	0 0 0 0 0 0 8 0 h
CH11 PATH #3 SET MASK	0 0 2 0 0 0 0 0 h
RESULT	0 0 2 0 0 0 8 0 h

The result has the proper relay bits set for connecting Path #3 on Channels 4 and Channel 11. Finally, we energize the Relays by performing a 32-bit write of the resultant value to the Relay Control Register (208000h).

16-bit Programming Description:

The same results can be achieved using 16-bit read and write operations by using an Offset of 8000h for Channels 1-8 and an Offset of 8002h for Channels 9-16.

Example #4 Let's configure Channel 3 so that it has Path #4 connected to Common. Following our programming process, first we perform a 16-bit read from 208000h. We'll call this value CH1_CH8 as it represents the current configuration of 1X4 Channels 1-8.

OPERATION	16-bit Read
ADDRESS	0x208000h
DATA	0 0 8 0 h

Then we clear the Channel 3 Relay bits by ANDing the Channel 3 Clear Mask value from Table 2 with the CH1_CH8 value.

OPERATION	16-bit AND
CH1_CH8	0 0 8 0 h
CH3 CLEAR MASK	F F C F h
RESULT	0 0 8 0 h

Next, we prepare to energize the Channel 3 Relays ORing the Channel 3 Set Mask value from Table 2 with our CH1_CH8 value.

OPERATION	16-bit OR
CH1_CH8	0 0 8 0 h
CH3 PATH #4 SET MASK	0 0 3 0 h
RESULT	0 0 B 0 h

The result has both the 1FormC and 2FormC Relay bits set for Channel 3. Note that none of the other bits have been modified. Thus, when we perform a 16-bit write of this value to the Relay Control Register (208000h) we will energize both of the Channel 3 Relays and connect Path #4 to Common.

16-bit Programming Description: (Continued)

Example #5 Now Let's configure Channels 10 and 11 so that they both have Path #2 connected to Common. First, we perform a 16-bit read from 208002h. We'll call this value CH9_CH16 as it represents the current configuration of 1X4 Channels 9-16.

OPERATION	16-bit Read
ADDRESS	0x208002h
DATA	0 0 2 0 h

Then we clear the Channel 10 and 11 Relay bits by ANDing the Channel 10 and 11 Clear Mask values from Table 2 with the CH9_CH16 value.

OPERATION	16-bit AND
CH9_CH16	0 0 2 0 h
CH10 CLEAR MASK	F F F 3 h
RESULT	0 0 2 0 h

↓

OPERATION	16-bit AND
CH9_CH16	0 0 2 0 h
CH11 CLEAR MASK	F F C F h
RESULT	0 0 0 0 h

Next, we prepare to energize the Channel 10 and 12 Relays by ORing the proper Set Mask values from Table 2 with our CH9_CH16 value.

OPERATION	16-bit OR
CH9_CH16	0 0 0 0 h
CH10 PATH #2 SET MASK	0 0 0 4 h
RESULT	0 0 0 4 h

OPERATION	16-bit OR
CH9_CH16	0 0 0 4 h
CH11 PATH #2 SET MASK	0 0 1 0 h
RESULT	0 0 1 4 h

The result has the 2FormC Relay bit set for Channels 10 and 11. Note that none of the other Channel bits have been modified. Thus, when we perform a 16-bit write of this value to the Relay Control Register (208002h) we will connect Path #2 to Common for both Channels.

Chapter 9 Connector / Channel Assignments

Single/Dual 4X4 Matrix Channels
A1-B4 provide the first eight (8) 4X4 Matrix Channel inputs and outputs. C1-D4 provide the second eight (8) 4X4 Matrix Channel inputs and outputs.

Connector	1X4	Connector	1X4
A1	Channel 1	B1	Channel 5
A2	Channel 2	B2	Channel 6
A3	Channel 3	B3	Channel 13
A4	Channel 4	B4	Channel 14
C1	Channel 7	D1	Channel 11
C2	Channel 8	D2	Channel 12
C3	Channel 9	D3	Channel 15
C4	Channel 10	D4	Channel 16

Dual 4X4

Chapter 10 Register Programming

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix is a Register-Based VXI Module with registers in both the A16 and A24 Address Spaces. The A16 Address Space contains the VXI Device Registers, which are 16-bit registers defined by the VXI Specification. See pages 24-32 for Device Register programming Examples. The A24 Address Space contains the Single/Dual 4X4 Matrix Relay Registers. See page 31 for Single/Dual 4X4 Matrix Relay Register programming Examples.

Programming VXI Device Registers

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix VXI Device Registers are read / write registers located in the A16 Address Space. Some Device Registers must be written only by a Resource Manager. Table 4 lists the VXI Device Registers. Pages 7-14 describe the contents of these registers.

VXI Device Registers for ASCOR 3000-155(A)			
VXI Device Dependent Registers			
Address	**Description**	**Read**	**Write**
3Eh	ASCOR Control Register	Yes	Yes
3Ch	Register Not Used	-	-
3Ah	Register Not Used	-	-
38h	Register Not Used	-	-
36h	Register Not Used	-	-
34h	Register Not Used	-	-
32h	Register Not Used	-	-
30h	Register Not Used	-	-
2Eh	Register Not Used	-	-
2Ch	Register Not Used	-	-
2Ah	Register Not Used	-	-
28h	Register Not Used	-	-
26h	Register Not Used	-	-
24h	Register Not Used	-	-
22h	Register Not Used	-	-
20h	Register Not Used	-	-
VXI Device Class Dependent Registers			
Address	**Description**	**Read**	**Write**
1Eh	Register Not Used	-	-
1Ch	Register Not Used	-	-
1Ah	Register Not Used	-	-
18h	Register Not Used	-	-
16h	Register Not Used	-	-
14h	Register Not Used	-	-
12h	Register Not Used	-	-
10h	Register Not Used	-	-
0Eh	Register Not Used	-	-
0Ch	Register Not Used	-	-
0Ah	Register Not Used	-	-
08h	Register Not Used	-	-
VXI Configuration Registers			
Address	**Description**	**Read**	**Write**
06h	Offset Register	Yes	Res. Man. Only
04h	Status / Control Register	Yes	Yes
02h	Device Type Register	Yes	Res. Man. Only
00h	ID / Logical Address Register	Yes	Res. Man. Only

Programming VXI Device Registers (Continued)

Since all 16 bits of the Device Register are programmed with a single write operation, care must be taken when values are written to the Device Registers in order to prevent unintended function enabling or disabling. In order to preserve the states of the functions that you do not want to alter, perform the following sequence of operations:

1. Read the Device Register first,
2. Modify only the bits you intend to program using the copy of the Device Register,
3. Write the new value back to the Device Register.

Here are some example codes for reading the ID Register of the 3000-155(A) Module.

Example using National Instruments NI-VXI calls with the Logical Address of 5

```
/*  C code segment for reading the ID Register using VXIinReg call.  */

        int16   ret;
        uint16  la          = 5;   /* Logical Address */
        uint16  reg         = 0;   /* ID Register offset */
        uint16  value16;

        /* Read the ID Register */
        ret = VXIinReg (la, reg, &value16);

        /* Check for read error */
        if (ret < 0)
               /*  Error occurred during read.  */;

/*  C code segment for reading the ID Register using VXIin call.  */

        int16   ret;
        uint16  accessparms= 1;      /* A16, Nonprivileged data access, Motorola Byte Order */
        uint32  address    = 0xC140; /* LA * 0x40 + 0xC000 + ID Register offset  */
        uint16  width      = 2;      /* 16-bit word */
        uint16  value16;

        /* Read the ID Register */
        ret = VXIin (accessparms, address, width, &value16);

        /* Check for read error */
        if (ret < 0)
               /*  Error occurred during read.  */;
```

Example using VXI*plug&play* VISA calls

```
/*  C code segment for reading the ID Register using ViIn16 call.  */

        ViStatus            as3xxx_status;
        ViSession    vi;            /*  vi from previous call to as3xxx_init  */
        ViUInt16            space   = VI_A16_SPACE;
        ViBusAddress offset = 0x00; /*  Offset of the ID Register  */
        ViUInt16            val16;

        /* Read the ID Register */
        as3xxx_status = viIn16 (vi, space, offset, &val16);

        /* Check for read error */
        if (as3xxx_status < VI_SUCCESS)
               /*  Error occurred during read.  */;
```

Resetting ASCOR VXI Module

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix VXI Module can be reset to a power up state by setting the Device Reset bit in the Status / Control (04h) register in the VXI Configuration Registers. Care must be taken when writing to this register since all bits other than the Device Reset bit must not be changed. In order to preserve the states of all other bits, perform the following sequence of operations:

1. Read the Status / Control register,
2. Set only the Device Reset bit,
3. Write the modified word to the Status / Control register.

After the reset operation, the module must be brought back to a normal operational mode in order for the relays to close. The 3000-155(A) Module can be set back to the normal operation mode by clearing the Device Reset bit without modifying any other bits.

Here are some example codes for resetting the 3000-155(A) Module.

Example using National Instruments NI-VXI calls with the Logical Address of 5

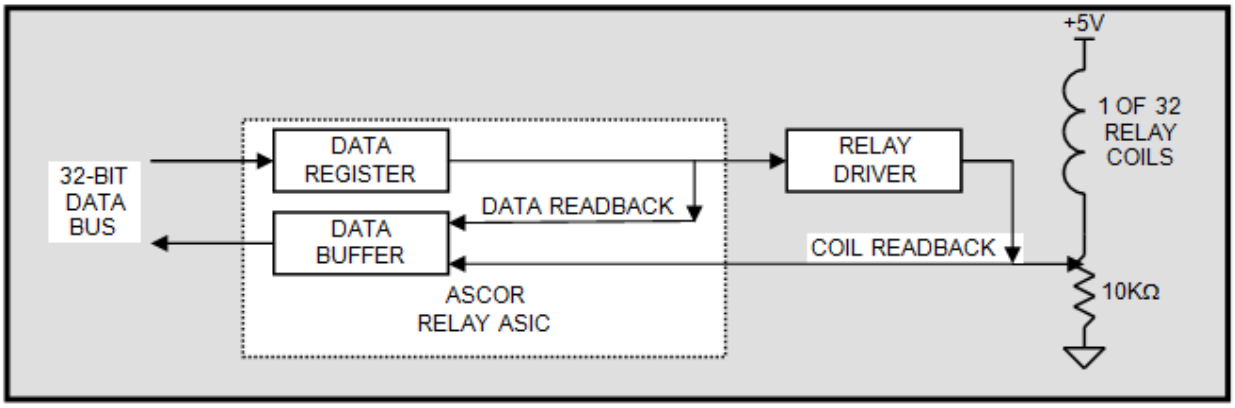
```
/*  C code segment for resetting the ASCOR VXI module using VXIinReg and VXIoutReg calls  */

        int16   ret;
        uint16  la          = 5;   /* Logical Address */
        uint16  reg         = 4;   /* Status / Control register offset */
        uint16  value16;

        /* Read the Status / Control Register */
        ret = VXIinReg (la, reg, &value16);

        /* Check for read error */
        if (ret < 0)
               /*  Error occurred during read.  */;


        /* Set the Device Reset bit in the copy of the Status / Control Register */
        value16 |= 0x0001;


        /* Write to the Status / Control Register */
        ret = VXIoutReg (la, reg, value16);

        /* Check for write error */
        if (ret < 0)
               /*  Error occurred during write.  */;


        /* Bring the module back to the normal operation by
           clearing the Device Reset bit in the copy of the Status / Control Register */
        value16 &= 0xFFFE;


        /* Write to the Status / Control Register */
        ret = VXIoutReg (la, reg, value16);

        /* Check for write error */
        if (ret < 0)
               /*  Error occurred during write.  */;
```

3000-155(A) Reset Programming Example using National Instruments NI-VXI Calls (Continued)

```
/*  C code segment for resetting the ASCOR VXI module using VXIin and VXIout calls  */

        int16   ret;
        uint16  accessparms= 1;      /* A16, Nonprivileged data access, Motorola Byte Order */
        uint32  address     = 0xC144; /* LA * 0x40 + 0xC000 + Control / Status Register */
        uint16  width       = 2;     /* Word */
        uint16  value16;
        uint32  value32;

        /* Read the Status / Control Register */
        ret = VXIin (accessparms, address, width, &value16);

        /* Check for read error */
        if (ret < 0)
               /*  Error occurred during read.  */;

        /* Set the Device Reset bit in the copy of the Status / Control Register */
        value32 = value16;
        value32 |= 0x0001;


        /* Write to the Status / Control Register */
        ret = VXIout (accessparms, address, width, value32);

        /* Check for write error */
        if (ret < 0)
               /*  Error occurred during write.  */;


        /* Bring the module back to the normal operation by
           clearing the Device Reset bit in the copy of the Status / Control Register */
        value32 &= 0xFFFE;


        /* Write to the Status / Control Register */
        ret = VXIout (accessparms, address, width, value32);

        /* Check for write error */
        if (ret < 0)
               /*  Error occurred during write.  */;
```

3000-155(A) Reset Programming Example using VXI*plug&play* VISA Calls

```
/*  C code segment for resetting the ASCOR VXI Module  */

        ViStatus            as3xxx_status;
        ViSession    vi;            /*  vi from previous call to as3xxx_init  */
        ViUInt16            space   = VI_A16_SPACE;
        ViBusAddress offset = 0x04; /*  Offset of the Status / Control Register  */
        ViUInt16            value16;

        /* Read the Status / Control Register */
        as3xxx_status = viIn16 (vi, space, offset, &value16);

        /* Check for read error */
        if (as3xxx_status < VI_SUCCESS)
               /*  Error occurred during read.  */;


        /* Set the Device Reset bit in the copy of the Status / Control Register */
        value16 |= 0x0001;


        /* Write to the Status / Control Register */
        as3xxx_status = viOut16 (vi, space, offset, value16);

        /* Check for write error */
        if (as3xxx_status < VI_SUCCESS)
               /*  Error occurred during write.  */;


        /* Bring the module back to the normal operation by
           clearing the Device Reset bit in the copy of the Status / Control Register */
        value16 &= 0xFFFE;


        /* Write to the Status / Control Register */
        as3xxx_status = viOut16 (vi, space, offset, value16);

        /* Check for write error */
        if (as3xxx_status < VI_SUCCESS)
               /*  Error occurred during write.  */;
```



Simplified Relay Circuit Diagram

Changing the 3000-155(A) Relay Register Read Mode

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix Relay Registers are read / write registers. When a relay register is read the states of the coils associated to that register are returned. Normally, the states of the coils should match the values which were written. They may not match when error conditions occur or when relay coil drivers are disabled.

The read mode of the Relay Registers can be switched between reading the states of the coils and reading data registers used for programming coils. The default read mode is reading the coil states. Any module reset brings the read mode back to reading coil states.

Change the mode by setting the Reg Bit in the Relay Control (3Eh) register in the VXI Device Dependent Registers. Care must be taken when writing to this register since all bits other than the Reg Bit must not be changed. In order to preserve the states of all other bits, perform the following sequence of operations:

1. Read the Relay Control register,
2. Set only the Reg Bit,
3. Write the modified word to the Relay Control register.

Subsequent reading of the Relay Registers will return the values of the data registers instead of the coil states. Reg Bit is cleared and the read mode is reset to reading the coil states when a module is reset.

Here are some example codes for changing the read mode of the 3000-155(A) Module.

Example using National Instruments NI-VXI calls with the Logical Address of 5

```
/*  C code segment for switching to data register readback using VXIinReg and VXIoutReg calls  */

        int16   ret;
        uint16  la          = 5;   /* Logical Address */
        uint16  reg         = 0x3E;   /* Relay Control register offset */
        uint16  value16;

        /* Read the Status / Control Register */
        ret = VXIinReg (la, reg, &value16);

        /* Check for read error */
        if (ret < 0)
                /*  Error occurred during read.  */;


        /* Set the Reg Bit in the copy of the Status / Control Register */
        value16 |= 0x0002;

        /* Write to the Status / Control Register */
        ret = VXIoutReg (la, reg, value16);

        /* Check for write error */
        if (ret < 0)
                /*  Error occurred during write.  */;
```

28

Changing the 3000-155(A) Relay Register Read Mode (Continued)

```
/*  C code segment for change to data register readback using VXIin and VXIout calls  */

        int16   ret;
        uint16  accessparms= 1;      /* A16, Nonprivileged data access, Motorola Byte Order */
        uint32  address     = 0xC17E; /* LA * 0x40 + 0xC000 + Relay Control Register  */
        uint16  width       = 2;      /* Word */
        uint16  value16;
        uint32  value32;

        /* Read the Status / Control Register */
```

```
ret = VXIin (accessparms, address, width, &value16);

/* Check for read error */
if (ret < 0)
        /*  Error occurred during read.  */;


/* Set the Reg Bit in the copy of the Status / Control Register */
value32 = value16;
value32 |= 0x0002;


/* Write to the Status / Control Register */
ret = VXIout (accessparms, address, width, value32);

/* Check for write error */
if (ret < 0)
        /*  Error occurred during write.  */;
```

Example using VXI*plug&play* VISA calls

```
/*  C code segment for change to data register readback using VISA calls  */

        ViStatus            as3xxx_status;
        ViSession    vi;            /*  vi from previous call to as3xxx_init  */
        ViUInt16            space   = VI_A16_SPACE;
        ViBusAddress offset = 0x3E; /*  Offset of the Relay Control Register  */
        ViUInt16            value16;

        /* Read the Status / Control Register */
        as3xxx_status = viIn16 (vi, space, offset, &value16);

        /* Check for read error */
        if (as3xxx_status < VI_SUCCESS)
                /*  Error occurred during read.  */;


        /* Set the Reg Bit in the copy of the Status / Control Register */
        value16 |= 0x0002;


        /* Write to the Status / Control Register */
        as3xxx_status = viOut16 (vi, space, offset, value16);

        /* Check for write error */
        if (as3xxx_status < VI_SUCCESS)
                /*  Error occurred during write.  */;
```

ASCOR 3000-155(A) Single/Dual 4X4 Matrix Relay Registers

The ASCOR 3000-155(A) Single/Dual 4X4 Matrix Relay Registers are located in the A24 address space and are accessed in 16 or 32-bit mode. The method of accessing the Relay Registers in the A24 address space is different from accessing the VXI Device Registers in the A16 address space. Therefore, care must be taken whenever accessing registers that are located in different address spaces.

A unique A24 base address is assigned by the Resource Manager to the A24 module in the system. The assignment of the base address is performed every time when the Resource Manager is executed. ASCOR 3000-155(A) Relay Registers start at an offset of 8000h from the module's assigned A24 base address. The sum of the two values, A24 base address and the register offset, gives the unique register address. Some interface library calls require the A24 register address. VXI*plug&play* library calls require only the offset of the register from the base address. The A24 base address is added to the offset internally.

Here are some example codes for writing to the 3000-155(A) registers.

Example using National Instruments NI-VXI calls

```
/*  C code segment for writing the value 0x1000 to the first custom register,
    assume A24 Base Address of 200000h */

       int16      ret;
       uint16     accessparms      = 2;    /* A24, Nonprivileged data access, Motorola
Byte Order */
       uint32     address;
       uint16     width  = 2;      /* Word */
       uint32     value32;

       address    = 0x208000;      /*  A24 Base Address + offset of the first custom
register  */
       value32    = 0x1000; /*  Value to write to the first custom register  */

       /* Write to the first custom register */
       ret = VXIout (accessparms, address, width, value32);

       /* Check for write error */
       if (ret < 0)
                  /*  Error occurred during write.  */;
```

Example using VXI*plug&play* VISA calls

```
/*  C code segment for writing the value 0x1000 to the first custom register  */

       ViStatus             as3xxx_status;
       ViSession    vi;
       ViUInt16             space   = VI_A24_SPACE;
       ViBusAddress offset  = 0x8000;      /*  Offset of the first custom register  */
       ViUInt16             value16;

       value16      = 0x1000;      /*  Value to write to the first custom register  */

       /* Write to the first custom register */
       as3xxx_status = viOut16 (vi, space, offset, value16);

       /* Check for write error */
       if (as3xxx_status < VI_SUCCESS)
            /*  Error occurred during write.  */;
```

Chapter 11 Miscellaneous Questions and Answers

Chapter 12 *Q: How do I calculate the 3000-155(A) Module's A16 Base Address?*

A: The A16 Base Address of the 3000-155(A) is derived from the Logical Address. The formula for calculating the A16 Base Address is as follows:

A16 Base Address = C000h + LA x 40h

where LA is the Logical Address of a module

Logical Address	A16 Base Address
1	C040h
2	C080h
3	C0C0h
4	C100h
5	C140h
and so on . . .	

If the module's Logical Address is 5 then A16 Base Address is C140h and Device Register addresses are as follows:

Address	Device Registers
C140h	ID Register / Logical Address Register
C142h	Device Type Register
C144h	Status / Control Register
C146h	Offset Register
C17Eh	ASCOR Control Register

If the module's Logical Address is 8 then A16 Base Address is C200h and Device Register addresses are as follows:

Address	Device Registers
C200h	ID Register / Logical Address Register
C202h	Device Type Register
C204h	Status / Control Register
C206h	Offset Register
C23Eh	ASCOR Control Register

Q: How do I Get the Module's A24 Base Address

A: The A24 Base Address of a VXI module can be derived from the value stored in the Offset Register (06h). To obtain ASCOR VXI Module's A24 base address, take the 8-Most Significant Bits of the Offset register and map them to the 8-Most Significant Bits of the A24 Base Address. All other bits in the A24 Base Address are set to zeroes. This conversion works for the modules whose Required Memory in the Device Type Register (A16 Address Space, Offset 02h) is set to 7h (Bits 12-15).

Offset Register (06h)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	X	X	X	X	X	X	X	X

A24 Base Address

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Following are some examples of the Offset Register Values and the corresponding A24 Base Addresses.

Offset Register Values	Derived A24 Base Addresses
2000h	200000h
3000h	300000h
7000h	700000h

Alternatively, A24 Base Address of a device can be obtained by issuing a library call.

Example using National Instruments NI-VXI calls with the Logical Address of 5

```
/*  C code segment for obtaining the device's A24 Base Address  */

        int16   ret;
        uint16  la          = 5;   /* Logical Address */
        uint16  field       = 12;  /* Base of A24/A32 address space */
        uint32  baseA24;

        /* Get the A24 Base Address */
        ret = GetDevInfo (la, field, &baseA24);

        /* Check for function error */
        if (ret < 0)
               /*  Error occurred during GetDevInfo.  */;
```