ARTISAN®
TECHNOLOGY GROUP

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

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

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *Astronics / Giga-tronics / Ascor 3000-66* at our website: ***Click HERE***

Programming Guide
for
ASCOR VXI Modules

Manual Part No. 07500740

Rev: C

Date: August 30, 2000

ASCOR, Inc
4384 Enterprise Place.
Fremont, CA 94538

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Rev	Description	Date
Preliminary		February 7, 1997
A	Aligned the title to fit inside the cover window.	April 7, 1997
B	Updated per ECO 2059 L. Garth	November 25, 1998
C	Update Table of Contents, page 6, 20, 21, and Chapter 6 added	August 30, 2000

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Table of Contents

1. ASCOR VXI MODULES OVERVIEW 1

1.1 VXI Register Based Modules 1
1.2 ASCOR VXI Module Types 1
1.3 VXI Device Registers 1
1.4 ASCOR Relay Registers 2
1.5 ASCOR Custom Registers 2
1.6 Static and Dynamic Configurations 2

2. VXI DEVICE REGISTER DESCRIPTION 3

2.1 VXI Configuration Registers 4
2.2 VXI Device Class Dependent Registers 9
2.3 VXI Device Dependent Registers for Switch Modules 10
2.4 VXI Device Dependent Registers for ASCOR Non-Switch Modules 12

3. ASCOR REGISTER DESCRIPTION 13

3.1 ASCOR Relay Registers for Switch Modules 13
3.2 ASCOR Custom Registers for ASCOR A16/A24 Modules 14
3.3 ASCOR Custom Registers for ASCOR A16/A32 Modules 15

4. PROGRAMMING WITH REGISTERS 16

4.1 Programming VXI Device Registers 16
4.2 Resetting ASCOR VXI Module 17
4.3 Changing the Relay Register Read Mode for Switch Modules 20
4.4 Programming ASCOR Relay Registers for Switch Module 22
4.5 Programming ASCOR Custom Registers for A16/A24 Module 23
4.6 Programming ASCOR Custom Registers for A16/A32 Module 24

5. MISCELLANEOUS QUESTIONS AND ANSWERS 25

5.1 How to Get the Module's A16 Base Address 25
5.2 How to Get the Module's A24 Base Address 26
5.3 How to Get the Module's A32 Base Address 27
5.4 How to Change the Module's Logical Address 28

6. ASCOR VXIPlug&play DRIVERS QUESTIONS AND ANSWERS 29

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1. ASCOR VXI Modules Overview

1.1 VXI Register Based Modules

ASCOR VXI Modules are register based devices and support VXIbus register maps. All controls to ASCOR VXI Modules 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. ASCOR VXI Modules **are not** message based devices and **do not** support VXIbus communication protocols.

1.2 ASCOR VXI Module Types

ASCOR VXI Modules are single device modules. They can be classified into three different types of devices. The different types are based on the use of operational registers in the A16, A24, and A32 address spaces. The three types of devices are defined below.

VXI A16 Devices
VXI A16 Devices have registers in A16 address space only. These registers are VXI Device Registers and they are accessed as a 16-bit word.

ASCOR Model 3000-16 GUARDIAN - VXI Alert Status Monitor is an example of a VXI A16 Device.

VXI A16/A24 Devices
These devices have registers in A16 address space as well as in A24 address space. The registers located in the A16 address space are VXI Device Registers. They are accessed as a 16-bit word. The registers located in the A24 address space are ASCOR registers. They are accessed as a 16-bit word, but if the ASCOR VXI Module is equipped with VXIMAX™ 16/32 then these ASCOR registers can also be accessed as a 32-bit word.

All standard and custom ASCOR VXI Switch Modules are VXI A16/A24 devices. The ASCOR Relay Registers are located in the A24 address space.

There are other ASCOR VXI Modules that are VXI A16/A24 devices.

VXI A16/A32 Devices
These devices have registers in A16 address space as well as in A32 address space. The registers located in the A16 address space are VXI Device Registers. They are accessed as a 16-bit word. The registers located in the A32 address space are ASCOR registers. They are accessed as a 32-bit word.

Most custom ASCOR VXI Non-Switch Modules belong to this type.

1.3 VXI Device Registers

VXI Device Registers are located in the A16 address space, which can be accessed by A16 address mode. VXI Device Registers are present in all three types of VXI Devices. VXI Device Registers are separated into Configuration, Device Class Dependent, and Device Dependent registers. ASCOR VXI Switch Modules provide all 4 Configuration Registers and one Device Dependent registers. The rest of the A16 register space is not populated. ASCOR VXI Non-Switch Modules provide all 4 Configuration Registers. All the Device Dependent registers are customized for individual module. Device Class Dependent registers are not used in any ASCOR VXI Modules.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

1.4 ASCOR Relay Registers

ASCOR Relay Registers for VXI Switch Modules are located in the A24 address space and they can only be accessed with A24 address mode. The number of A24 registers is determined by each module's usage. The method of accessing ASCOR 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. Programming examples are shown in *Section 4.4: Programming ASCOR Relay Registers for Switch Module* using several popular interface libraries.

1.5 ASCOR Custom Registers

ASCOR Custom Registers for VXI Modules are located in one of either A24 or A32 address spaces. The registers in the A24 address space can only be accessed with A24 address mode. The registers in the A32 address space can only be accessed with A32 address mode. The number of these custom registers is determined by each module's usage. The method of accessing ASCOR Registers in the A24 and A32 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. Programming examples are shown in *Section 4.5: ASCOR Custom Registers for ASCOR A16/A24 Modules* and *Section 4.6: ASCOR Custom Registers for ASCOR A16/A32 Modules* using several popular interface libraries.

1.6 Static and Dynamic Configurations

ASCOR VXI Modules support both Static Configuration and Dynamic Configuration of Logical Address. In Static Configuration mode the Logical Address of the module is 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. Procedures for changing the Logical Addresses for ASCOR VXI Modules are discussed in *Section 5.4: VXI Device Register Description*.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2. VXI Device Register Description

ASCOR VXI Modules have VXI Device Registers located in the A16 address space. There are 32 VXI Device Registers and they are all 16 bits wide. The first four registers of the 32 VXI Device Registers are VXI Configuration Registers. The next 12 registers are VXI Device Class Dependent Registers. The last 16 registers are VXI Device Dependent Registers.

ASCOR VXI Switch Modules support 5 of the 32 VXI Device Registers, four in VXI Configuration Registers and one in VXI Device Dependent Registers. All other registers are not used nor supported.

VXI Device Registers for ASCOR Switch Modules	
VXI Device Dependent Registers	
3Eh	Relay Control Register
20h - 3Ch	15 Registers Not Used
VXI Device Class Dependent Registers	
08h - 1Eh	12 Registers Not Used
VXI Configuration Registers	
06h	Offset Register
04h	Status / Control Register
02h	Device Type Register
00h	ID / Logical Address Register

For ASCOR VXI Non-Switch Modules all 4 VXI Configuration Registers are supported, none of the VXI Device Class Dependent Registers are supported, and the number of VXI Device Dependent Registers supported varies from module to module. Consult the module's Technical Manual for detail.

VXI Device Registers for ASCOR Non-Switch Modules	
VXI Device Dependent Registers	
20h - 3Eh	16 Register for Custom Use
VXI Device Class Dependent Registers	
08h - 1Eh	12 Registers Not Used
VXI Configuration Registers	
06h	Offset Register
04h	Status / Control Register
02h	Device Type Register
00h	ID / Logical Address Register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

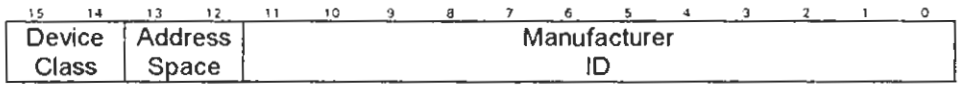
2.1 VXI Configuration Registers

The first four registers in the A16 address space are Configuration Registers. Each register is 16-bits wide. They are explained below.

Offset **Description**

00h **ID Register (read) / Logical Address Register (write)**

A read of this 16-bit register provides information about ASCOR VXI Module's configuration.



The breakdown of this value is defined below.

(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 ASCOR VXI A16 only Modules, the register should read back a value of FFB5h.
For ASCOR VXI A16/A24 Modules, the register should read back a value of CFB5h.
For ASCOR VXI A16/A32 Modules, the register should read back a value of DFB5h.

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.**

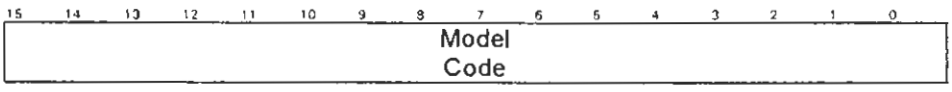
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Offset **Description**

02h **Device Type (read/write)**

A read of this 16-bit register provides information about ASCOR VXI 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.

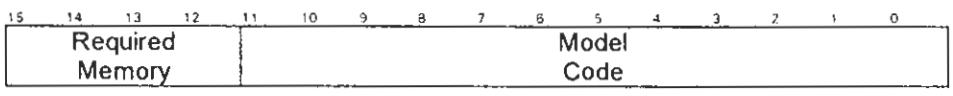
For ASCOR VXI A16 only Modules.



The breakdown of this value is defined below.

(Bits 15-0) Model Code: This field contains a unique card identifier.
xxxxh = One of ASCOR Model Codes such as 0F06h. This number is different from the ASCOR Model numbers such as 3000-16.

For ASCOR VXI A16/A24 and A16/A32 Modules.



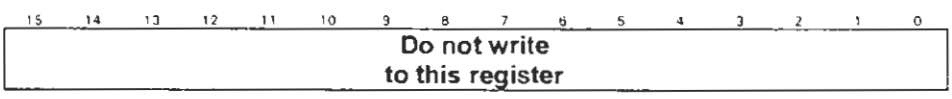
The breakdown of this value is defined below.

(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 a unique card identifier.
xxxh = One of ASCOR Model Codes such as F2Dh or FCCh. This number is different from the ASCOR Model numbers such as 3000-45 or 3000-266. Each ASCOR VXI Module has a unique Model Code.

For ASCOR VXI A16/A24 Modules, the register should read back a value of 7xxxh.
For ASCOR VXI A16/A32 Modules, the register should read back a value of Fxxxh.

A write to this 16-bit register is provided for VXIbus definition. **Do not write to this register.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Offset **Description**

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

A read of this 16-bit register provides information about the Device's status.

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

The breakdown of this value is defined below.

(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 ASCOR VXI A16 only Modules, the register should read back a value of 7FFCh.
For ASCOR VXI A16/A24 Modules, the register should read back a value of FFFCh.
For ASCOR VXI A16/A32 Modules, the register should read back a value of FFFCh.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

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

The breakdown of this value is defined below.

(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)

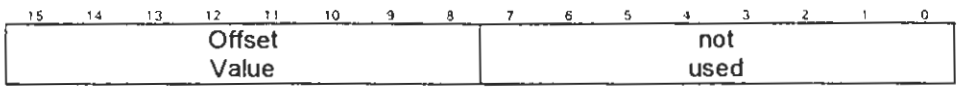
(Bit 0) Device Reset: This bit controls the state of the device.
1b = Reset the device to power-on state.
0b = Normal operational mode

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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

A read of the 16-bit register provides information for calculating the base address of this device's A24 or A32 operational registers.

For A24 Address Space



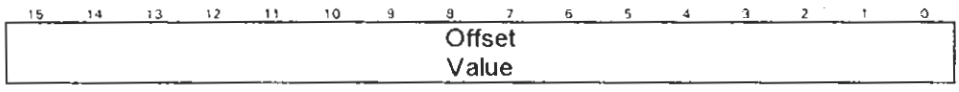
The breakdown of this value is defined below.

(Bits 15-8) Offset Value: This field is used for calculating the A24 Base Address.

(Bits 7-0) Don't Care bits (Not used)

To obtain the A24 base address for the device, 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 5.2: How to Get the Module's A24 Base Address.*

For A32 Address Space

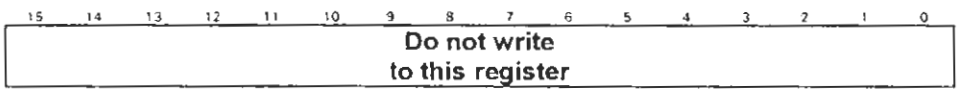


The breakdown of this value is defined below.

(Bits 15-0) Offset Value: This field is used for calculating the A32 Base Address

To obtain the A32 base address for the device, take all 16 bits of the Offset register and map them to the 16 most significant bits of the A32 Base Address. All other bits in the A32 Base Address are set to zeroes. For more detail refer to *Section 5.3: How to Get the Module's A32 Base Address.*

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.**



Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.2 VXI Device Class Dependent Registers

All ASCOR VXI Modules do 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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

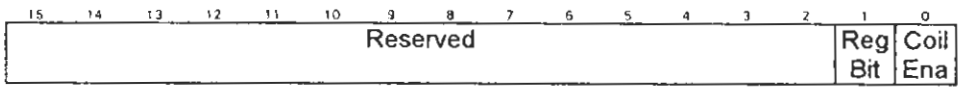
2.3 *VXI Device Dependent Registers for Switch Modules*

There are 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 ASCOR VXI Switch Modules, 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

3Eh Relay Control Register (read/write)

A read of the 16-bit register provides the relay control status.



The breakdown of this value is defined below.

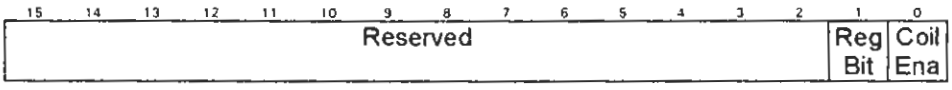
(Bits 15-2) Reserved
0h = Should always readback zeroes

(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

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

A write to this 16-bit register is sets the relay control.



The breakdown of this value is defined below.

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

(Bit 1) Reg Bit: This bit controls the device's readback mode.
0b = Enable relay coil state readback
1b = Enable data register state readback

(Bit 0) Coil Enable: This bit controls the device's relay coil driver.
0b = Enable relay coil driver
1b = Disable relay coil driver

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

2.4 VXI Device Dependent Registers for ASCOR Non-Switch Modules

There are 16 Device Dependent Registers in the A16 address space following the Device Class Dependent Register space. Each register is 16 bits wide. Refer to the Module's Technical Manual for detailed description for these Device Dependent Registers.

Offset	Description
20h	Custom Use
22h	Custom Use
24h	Custom Use
26h	Custom Use
28h	Custom Use
2Ah	Custom Use
2Ch	Custom Use
2Eh	Custom Use
30h	Custom Use
32h	Custom Use
34h	Custom Use
36h	Custom Use
38h	Custom Use
3Ah	Custom Use
3Ch	Custom Use
3Eh	Custom Use

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3. ASCOR Register Description

3.1 ASCOR Relay Registers for Switch Modules

ASCOR Relay Registers are located in the A24 Address Space. The number of registers varies from module to module. Typically relay registers start at offset 8000h within the A24 address space assigned by the resource manager. There are a few ASCOR VXI Modules with the relay registers starting at offset 40h. Consult the module's Technical Manual to find the offset addresses of the relay registers. All ASCOR relay registers can be accessed as a 16-bit word. For ASCOR VXI Modules with VXIMAX, relay registers can also be accessed as 32-bit words.

Read / Write
ASCOR 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. On some rare occasions, the design of the module may not permit matching results.

Read Mode
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.

Emulating resistors
Typically, when relay register bits are not associated with relay coils, resistors are used in place of coils. The resistors emulate the coils and eliminate the need for masking. These emulated bits can be written with either 0's or 1's with no side effect to the module.

Offset Address in 16-Bit Mode
ASCOR Relay Registers are accessed in 16-bit mode. Address offsets for the relay registers increment by two (2). Sample offsets for the ASCOR Relay Registers in 16-bit word mode:

Offset	Description
8000h	First Relay Register (read) / (write)
8002h	Second Relay Register (read) / (write)
8004h	Third Relay Register (read) / (write)
8006h	Fourth Relay Register (read) / (write)
. . . and so on . . .	

Offset address in 32-Bit Mode
If the ASCOR VXI Module is a VXIMAX module the Relay Registers can additionally be accessed in 32-bit mode. Address offsets for the relay registers increment by four (4). Sample offsets for the ASCOR Relay Registers in 32-bit word mode:

Offset	Description
8000h	First Relay Register (read) / (write)
8004h	Second Relay Register (read) / (write)
8008h	Third Relay Register (read) / (write)
800Ch	Fourth Relay Register (read) / (write)
. . . and so on . . .	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.2 ASCOR Custom Registers for ASCOR A16/A24 Modules

ASCOR Custom Registers for A16/A24 Modules are located in the A24 Address Space. The number of registers varies from module to module. Typically custom registers start at offset 8000h within the A24 address space assigned by the resource manager. All ASCOR custom registers can be accessed as a 16-bit word. For ASCOR VXI Modules with VXIMAX, custom registers can also be accessed as 32-bit words.

Offset Address in 16-Bit Mode
ASCOR Custom Registers for A16/A24 Modules can be accessed in 16-bit mode. Address offsets for the custom registers increment by two (2). Sample offsets for the ASCOR Custom Registers in 16-bit word mode:

Offset	Description
8000h	First Custom Register
8002h	Second Custom Register
8004h	Third Custom Register
8006h	Fourth Custom Register
. . . and so on . . .	

Offset address in 32-Bit Mode
If the ASCOR VXI Module is a VXIMAX module the Custom Registers can additionally be accessed in 32-bit mode. Address offsets for the custom registers increment by four (4). Sample offsets for the ASCOR Custom Registers in 32-bit word mode:

Offset	Description
8000h	First Custom Register
8004h	Second Custom Register
8008h	Third Custom Register
800Ch	Fourth Custom Register
. . . and so on . . .	

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

3.3 ASCOR Custom Registers for ASCOR A16/A32 Modules

ASCOR Custom Registers for A16/A32 Modules are located in the A32 Address Space. The number of registers varies from module to module. Typically custom registers start at offset 8000h within the A32 address space assigned by the resource manager. All ASCOR custom registers are accessed as 32-bit words.

Offset address in 32-Bit Mode

ASCOR Custom Registers for A16/A32 Modules can only be accessed in 32-bit mode. Address offsets for the custom registers increment by four (4). Sample offsets for the ASCOR Custom Registers in 32-bit word mode:

Offset	Description
8000h	First Custom Register
8004h	Second Custom Register
8008h	Third Custom Register
800Ch	Fourth Custom Register

. . . and so on . . .

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4. Programming with Registers

4.1 Programming VXI Device Registers

VXI Device Registers are read / write registers, but some device registers must be written only by a resource manager. See *Section 2: VXI Device Register Description* for registers reserved for the resource manager access. Since all 16 bits 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 ASCOR VXI 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.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.2 *Resetting ASCOR VXI Module*

ASCOR VXI Modules 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. ASCOR VXI Modules 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 ASCOR VXI 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.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/*  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.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

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.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.3 Changing the Relay Register Read Mode for Switch Modules

ASCOR VXI Modules read mode can be changed to read the data register instead of the 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 ASCOR VXI 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 Relay 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 Relay Control Register */
        value16 |= 0x0002;


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

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


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

```
/*  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 Relay 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 Relay Control Register */
        value32 = value16;
        value32 |= 0x0002;


        /* Write to the Relay 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 Relay 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 Relay Control Register */
        value16 |= 0x0002;


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

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


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.4 *Programming ASCOR Relay Registers for Switch Module*

ASCOR relays can be closed or opened by setting or clearing the register bits in the A24 address space. Since all 16 or 32 bits are programmed with a single write operation, care must be taken when values are written to the relay registers in order to prevent unintended relay closing or opening. To preserve the states of the relays that you do not want to program, perform the following sequence of operations:

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

The correlation between the relays and the register bits can be found in the Register Map section of the ASCOR VXI Modules' Technical Manual.

All ASCOR Relay Registers are located in the A24 address space. 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 Relay Registers start at an offset from the module's assigned A24 base address. The sum of the two values, A24 base address and the relay register offset, gives the unique relay register address. Some interface library calls require the A24 relay 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 ASCOR relay registers.

Example using National Instruments NI-VXI calls

```
/*  C code segment for writing the value 0x1000 to the first relay 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 relay register  */
        value32 = 0x1000;   /*  Value to write to the first relay register  */

        /* Write to the first relay 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 relay register  */

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

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

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

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


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.5 Programming ASCOR Custom Registers for A16/A24 Module

ASCOR Custom Registers can be accessed through the registers in the A24 address space. Since all 16 or 32 bits are programmed with a single write operation, care must be taken when values are written to these registers in order to prevent unintended side effects. To preserve the configuration that you do not want to program, perform the following sequence of operations:

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

The definition of the custom registers can be found in ASCOR VXI Modules' Technical Manual.

All ASCOR Custom Registers are located in the A24 address space. 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 Custom Registers start at an offset from the module's assigned A24 base address. The sum of the two values, A24 base address and the custom register offset, gives the unique custom register address. Some interface library calls require the A24 custom 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 ASCOR custom 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.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

4.6 *Programming ASCOR Custom Registers for A16/A32 Module*

ASCOR Custom Registers can be accessed through the registers in the A32 address space. Since all 32 bits are programmed with a single write operation, care must be taken when values are written to these registers in order to prevent unintended side effects. To preserve the configuration that you do not want to program, perform the following sequence of operations:

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

The definition of the custom registers can be found in ASCOR VXI Modules' Technical Manual.

All ASCOR Custom Registers are located in the A32 address space. A unique A32 base address is assigned by the resource manager to the A32 module in the system. The assignment of the base address is performed every time when the resource manager is executed. ASCOR Custom Registers start at an offset from the module's assigned A32 base address. The sum of the two values, A32 base address and the custom register offset, gives the unique custom register address. Some interface library calls require the A32 custom register address. VXI*plug&play* library calls require only the offset of the register from the base address. The A32 base address is added to the offset internally.

Here are some example codes for writing to the ASCOR custom registers.

Example using National Instruments NI-VXI calls

```
/*  C code segment for writing the value 0x10000000 to the first custom register,
    assume A32 Base Address of 20000000h */

        int16   ret;
        uint16  accessparms  = 3;    /* A32, Nonprivileged data access, Motorola Byte Order */
        uint32  address;
        uint16  width        = 4;    /* Longword */
        uint32  value32;

        address = 0x20008000;        /*  A32 Base Address + first custom register offset  */
        value32 = 0x10000000;        /*  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 0x10000000 to the first custom register  */

        ViStatus        as3xxx_status;
        ViSession       vi;
        ViUInt16        space    = VI_A32_SPACE;
        ViBusAddress    offset   = 0x8000;        /*  Offset of the first custom register  */
        ViUInt32        value32;
        value32 = 0x10000000;  /*  Value to write to the first custom register  */

        /* Write to the first custom register */
        as3xxx_status = viOut32 (vi, space, offset, value32);

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


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5. Miscellaneous Questions and Answers

5.1 How to Get the Module's A16 Base Address

A16 Base Address 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	Relay Control Register

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

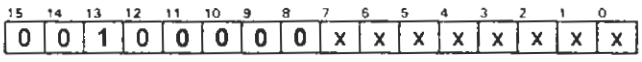
Address	Device Registers
C200h	ID Register / Logical Address Register
C202h	Device Type Register
C204h	Status / Control Register
C206h	Offset Register
C23Eh	Relay Control Register

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

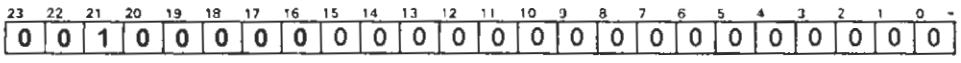
5.2 *How to Get the Module's A24 Base Address*

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 is set to 7.

Offset Register (06h)



A24 Base Address



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);

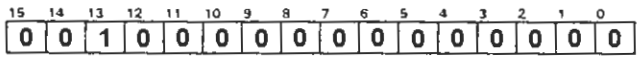
        /* Check for function error */
        if (ret < 0)
                /*  Error occurred during GetDevInfo.  */;
```


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

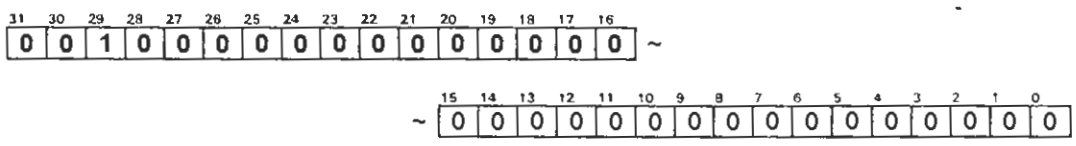
5.3 How to Get the Module's A32 Base Address

A32 Base Address of a VXI module can be derived from the value stored in the Offset Register (06h). To obtain ASCOR VXI Module's A32 base address, take the 16 most significant bits of the Offset register and map them to the 16 most significant bits of the A32 Base Address. All other bits in the A32 Base Address are set to zeroes. This conversion works for the modules whose Required Memory in the Device Type Register is set to Fh (15).

Offset Register (06h)



A32 Base Address



Following are some examples of the Offset Register Values and the corresponding A32 Base Addresses.

Offset Register Values	Derived A32 Base Addresses
2000h	20000000h
3000h	30000000h
7000h	70000000h

Alternatively, A32 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 A32 Base Address  */

        int16   ret;
        uint16  la          = 5;    /* Logical Address */
        uint16  field       = 12;   /* Base of A24/A32 address space */
        uint32  baseA32;

        /* Get the A32 Base Address */
        ret = GetDevInfo (la, field, &baseA32);

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


Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

5.4 How to Change the Module's Logical Address

The Logical Address of ASCOR VXI Modules can be changed manually using the two rotary switches located in the back of the module. These rotary switches represent the high and low hex digits of an eight bit Logical Address. Orient the module so that the component side of the PC board is on top. The two rotary switches are now aligned side by side. The rotary switch on the left represents the high hex digit and the rotary switch on the right represents the low hex digit. Each rotary switch can be turned clockwise or counter-clockwise. Align the rotary switch with the small white dot on the left side of the rotary switch casing. Turn each rotary switch to the desired hex digit value. Valid Logical Addresses for Static Configuration are between 01h (1) and FEh (254). The Logical Address of 00h (0) is reserved for Slot 0 computer. **Do not** set the Logical Address of ASCOR VXI Modules to 0.

ASCOR VXI Modules also supports Dynamic Configuration methods of assigning Logical Addresses. In order for Dynamic Configuration to work properly the resource manager software must also support Dynamic Configuration. Set the rotary switches on the modules to FFh (255) so that the resource manager software can dynamically assign Logical Addresses to ASCOR VXI Modules.

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6. ASCOR VXIPlug&play DRIVER QUESTIONS AND ANSWERS

6.1 How to add ASCOR VXI*plug&play* driver functions in LabVIEW (5.x)

1) **Install ASCOR VXI*plug&play* driver.**
2) **Run LabVIEW.**
3) **Open New VI.**
4) **On the Diagram window, select "Update VXIplug&play driver..." from the File menu.**
5) **Select the driver(s) that you want to incorporate from the "CVI Instrument Drivers (Win95) to be converted:" list (bottom window)**
   (our driver name starts with "as" and the model number is mangled in the following form: for example: 3000-2 is as3002; 3000-45 is as3045; and so on)
6) **Click on the OK button**
7) **The driver(s) will be converted.**

- **You can verify if the driver(s) was/were updated as follow:**

8) **Right click on the Diagram window to bring the Functions palette to view (or select "Show Functions Palette" from the Windows menu)**
9) **Click on the Instrument Drivers button**
10) **Check the driver(s) that are just added.**

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

6.2 How to add ASCOR VXIplug&play driver functions in HPVEE (4.0)

1) Install ASCOR VXIplug&play driver.

2) Run HPVEE

3) From I/O menu select "Instrument Manager ..."

4) Click on "Add..."

5) Enter a device name

6) Choose "VXI" for the interface

7) Set address to 0

8) Click on "Advanced I/O Config ..."

9) Select "Plug&play Driver" tab

10) Select our driver from the Plug&Play Driver Name
(our driver name starts with "as" and the model number is mangled in the following form:
for example 3000-2 is as3002; 3000-45 is as3045; 3000-47 is as3047)

11) Specify the address with the resource name you get from HPs VIC or some other program.

12) OK back to Instrument Manager panel

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

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

ARTISAN® TECHNOLOGY GROUP