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 *Ametek / VTI Instruments / VXI Technology VM6069* at our website: *Click HERE*

VM6069

CUSTOM PROTOCOL INTERFACE

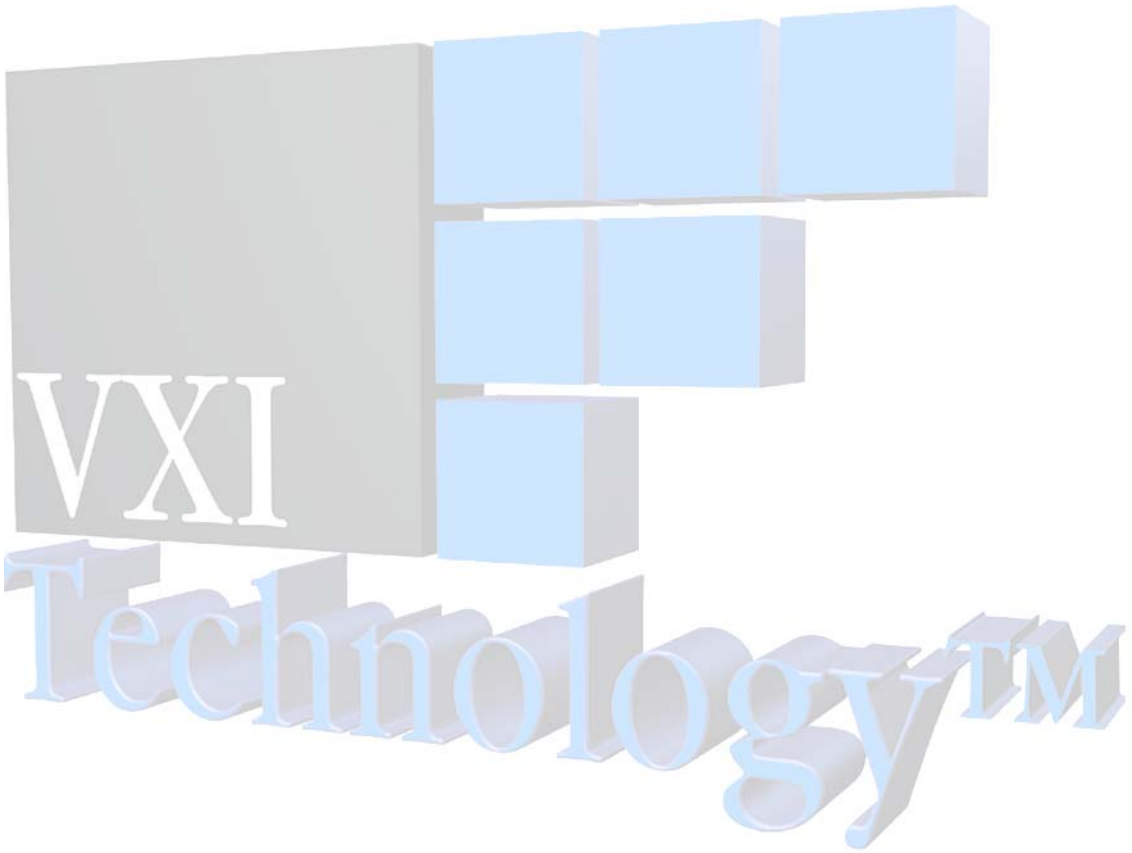
USER'S MANUAL

82-0041-000
Rev. April 7, 2003

VXI Technology, Inc.

2031 Main Street
Irvine, CA 92614-6509
(949) 955-1894

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



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

TABLE OF CONTENTS

INTRODUCTION

Certification 6
Warranty 6
Limitation of Warranty 6
Restricted Rights Legend 6
DECLARATION OF CONFORMITY 7
GENERAL SAFETY INSTRUCTIONS 9
Terms and Symbols 9
Warnings 9
SUPPORT RESOURCES 11

SECTION 1 13

INTRODUCTION 13
Introduction 13
Description 14
VM6069 Features 15

SECTION 2 17

PREPARATION FOR USE 17
Installation 17
Calculating System Power and Cooling Requirements 17
Setting the Chassis Backplane Jumpers 18
Setting the Logical Address 18
Front Panel Interface 19

SECTION 3 23

PROGRAMMING 23
Introduction 23
Interface FPGA 24
User FPGA 24
Register Access 26
Flash Memory 29
Flash - Read / Write 29
Downloading User FPGA Code / Data Pattern 29
USER FPGA DESIGN 33
User FPGA Design Guidelines 33
User FPGA Signal Description 35
VXI Register Address Decoding 39
VMIP Data Bus Interface 41
VXI Bus Acknowledge 41
VMIP Bus Timing 42
Using the Interrupt Signal 44
Accessing Peripherals 44
SRAM 44
Flash Memory 46
Open Collector Buffers 48
SIPEX Transmitters 50
SIPEX Receivers 50
SIPEX Transmitter and Receiver Termination 54
VXI Bus TTL Trigger Input 55
VXI Bus TTL Trigger Output 56

SECTION 4 57

COMMAND DICTIONARY 57
Introduction 57
Programming 57

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

Notation....58
Alphabetical Command Listing....58
Command Dictionary....62
IEEE 488.2 COMMON COMMANDS....63
*CLS....63
*ESE....64
*ESR?....65
*IDN?....66
*OPC....67
*RST....68
*SRE....69
*STB?....70
*TRG....71
*TST?....72
*WAI....73
INSTRUMENT SPECIFIC SCPI COMMANDS....74
CALibration:SECure:CODE....74
CALibration:SECure[:STATe]....75
OUTPut[:STATe]....76
OUTPut:TTLTrg....77
OUTPut[:TTLTrg]:POLarity....78
PROGram:MODule....79
TRIGger:SOURce:TTLTrg....80
TRIGger[:STATe]....81
REQUIRED SCPI COMMANDS....82
STATus:OPERation:CONDition?....82
STATus:OPERation:ENABle....83
STATus:OPERation:NTR....84
STATus:OPERation:PTR....85
STATus:OPERation[:EVENt]?....86
STATus:PRESet....87
STATus:QUEStionable:CONDition?....88
STATus:QUEStionable:ENABle....89
STATus:QUEStionable[:EVENt]?....90
APPENDIX A - APPLICATION EXAMPLE....91
Sample Code....91
Memory Access....91
SIPEX Latch....92
Open Collector Buffers....93
Misc. Registers....93
UF_STD.UCF....94
UF_STD.V....104
LOOPBACK CONNECTOR USED TO TEST VM6069....118
APPENDIX B - VM6069 SCHEMATIC....120
50-0110-000 - SCHEMATIC, VM6069, UNIVERSAL SERIAL INTERFACE....120
INDEX....121

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

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

CERTIFICATION

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

WARRANTY

The product referred to herein is warranted against defects in material and workmanship for a period of three years from the receipt date of the product at customer's facility. The sole and exclusive remedy for breach of any warranty concerning these goods shall be repair or replacement of defective parts, or a refund of the purchase price, to be determined at the option of VTI.

For warranty service or repair, this product must be returned to a VXI Technology authorized service center. The product shall be shipped prepaid to VTI and VTI shall prepay all returns of the product to the buyer. However, the buyer shall pay all shipping charges, duties, and taxes for products returned to VTI from another country.

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

LIMITATION OF WARRANTY

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

VXI Technology, Inc. shall not be liable for injury to property other than the goods themselves. Other than the limited warranty stated above, VXI Technology, Inc. makes no other warranties, express or implied, with respect to the quality of product beyond the description of the goods on the face of the contract. VTI specifically disclaims the implied warranties of merchantability and fitness for a particular purpose.

RESTRICTED RIGHTS LEGEND

Use, duplication, or disclosure by the Government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

VXI Technology, Inc.
2031 Main Street
Irvine, CA 92614-6509 U.S.A.

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

DECLARATION OF CONFORMITY

Declaration of Conformity According to ISO/IEC Guide 22 and EN 45014

MANUFACTURER'S NAME	VXI Technology, Inc.
MANUFACTURER'S ADDRESS	2031 Main Street Irvine, California 92614-6509
PRODUCT NAME	Custom Protocol Interface
MODEL NUMBER(S)	VM6069
PRODUCT OPTIONS	All
PRODUCT CONFIGURATIONS	All

VXI Technology, Inc. declares that the aforementioned product conforms to the requirements of the Low Voltage Directive 73/23/EEC *and the* EMC Directive 89/366/EEC *(inclusive* 93/68/EEC*) and carries the "CE" mark accordingly. The product has been designed and manufactured according to the following specifications:*

SAFETY	EN61010 (2001)
EMC	EN61326 (1997 w/A1:98) Class A CISPR 22 (1997) Class A VCCI (April 2000) Class A ICES-003 Class A (ANSI C63.4 1992) AS/NZS 3548 (w/A1 & A2:97) Class A FCC Part 15 Subpart B Class A EN 61010-1:2001

The product was installed into a C-size VXI mainframe chassis and tested in a typical configuration.

I hereby declare that the aforementioned product has been designed to be in compliance with the relevant sections of the specifications listed above as well as complying with all essential requirements of the Low Voltage Directive.

April 2003

CE

Jerry Patton, QA Manager

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

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

GENERAL SAFETY INSTRUCTIONS

Review the following safety precautions to avoid bodily injury and/or damage to the product. These precautions must be observed during all phases of operation or service of this product. Failure to comply with these precautions, or with specific warnings elsewhere in this manual, violates safety standards of design, manufacture, and intended use of the product.

Service should only be performed by qualified personnel.

TERMS AND SYMBOLS

These terms may appear in this manual:

WARNING Indicates that a procedure or condition may cause bodily injury or death.

CAUTION Indicates that a procedure or condition could possibly cause damage to equipment or loss of data.

These symbols may appear on the product:



ATTENTION - Important safety instructions



Frame or chassis ground

WARNINGS

Follow these precautions to avoid injury or damage to the product:

Use Proper Power Cord To avoid hazard, only use the power cord specified for this product.

Use Proper Power Source To avoid electrical overload, electric shock or fire hazard, do not use a power source that applies other than the specified voltage.

Use Proper Fuse To avoid fire hazard, only use the type and rating fuse specified for this product.

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

WARNINGS (CONT.)

Avoid Electric Shock

To avoid electric shock or fire hazard, do not operate this product with the covers removed. Do not connect or disconnect any cable, probes, test leads, etc. while they are connected to a voltage source. Remove all power and unplug unit before performing any service. ***Service should only be performed by qualified personnel.***

Ground the Product

This product is grounded through the grounding conductor of the power cord. To avoid electric shock, the grounding conductor must be connected to earth ground.

Operating Conditions

To avoid injury, electric shock or fire hazard:

- Do not operate in wet or damp conditions.
- Do not operate in an explosive atmosphere.
- Operate or store only in specified temperature range.
- Provide proper clearance for product ventilation to prevent overheating.
- DO NOT operate if any damage to this product is suspected. ***Product should be inspected or serviced only by qualified personnel.***



Improper Use

The operator of this instrument is advised that if equipment is used in a manner not specified in this manual, the protection provided by this equipment be may be impaired.

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

SUPPORT RESOURCES

Support resources for this product are available on the Internet and at VXI Technology customer support centers.



Internet Support

E-mail: support@vxitech.com
Web Address: http://www.vxitech.com

Telephone Support (U.S.)

Tel: (949) 955-1894 **West Coast**
(216) 447-8950 **East Coast**

Fax: (949) 955-3041 **West Coast**
(216) 447-8951 **East Coast**

VXI Technology Headquarters

Technical Support
VXI Technology, Inc.
2031 Main Street
Irvine, CA 92614-6509

Tel: (949) 955-1894
Fax: (949) 955-3041

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

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

SECTION 1

INTRODUCTION

INTRODUCTION

The VM6069 is a Custom Protocol Interface that provides various I/Os that are programmed and controlled by the User FPGA. It provides easy interface design to the VXIbus so that the User FPGA code can be designed, accessed and controlled by device dependent registers.

The most powerful feature of the VM6069 is that it is a member of the VXI Technology VMIP™ (*VXI Modular Instrumentation Platform*) family of VXIbus. This gives the user the added flexibility of combining it with other instruments, such as digital multimeters or digitizers, to create a multi-function C-size card. The VM6069 may be combined with any of the other members of the VMIP family to form a customized and highly integrated instrument (see Figure 1-1). This allows the user to reduce system size and cost by combining the VM6069 with two other instrument functions in a singlewide C-size VXIbus module. Up to three VM6069s can also be combined together on a single VXIbus card, making it an ideal choice for many applications.

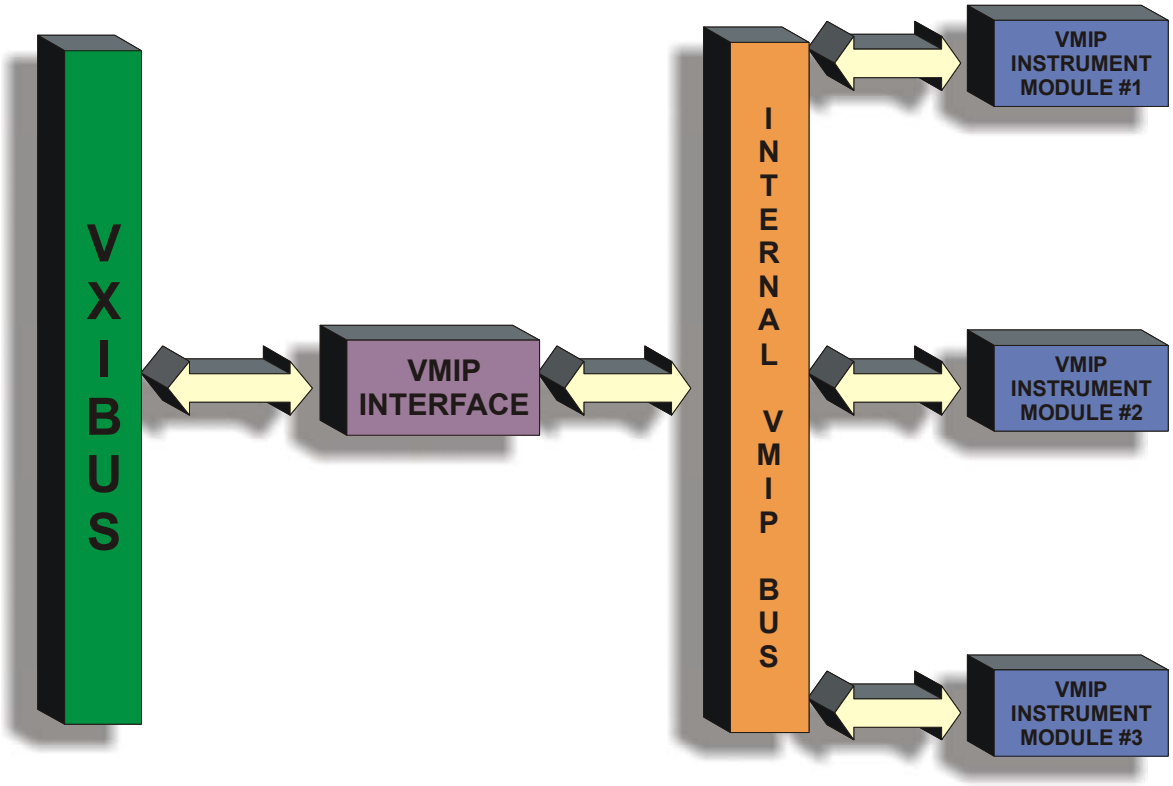


FIGURE 1-1 VMIP PLATFORM

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

DESCRIPTION

The VM6069 Custom Protocol Interface can be used in standard and non-standard protocol interfacing, where the hardware remains the same, and the FPGA program changes for each interface.

The protocol to be executed by this card can be downloaded to flash memory from the VXI controller through registers. At power-up, the User FPGA loads itself from flash memory. Module control and module status are done through the registers. All registers are mapped to device dependent registers.

The heart of the VM6069 is the User FPGA, which can be modified over the VXIbus backplane. Logic patterns are created to make the VM6069 conform to a desired protocol and downloaded to flash memory on the VM6069. There is 512k of on-board Flash memory; 64k is used for the User FPGA patterns and the remainder is available to the VM6069 for other user-defined pattern. The User FPGA is also supported by two Mbytes of RAM for data reception and transmission. There are three types of electrical interfaces that can be manipulated by the User FPGA:

- Two programmable multi-mode serial transceivers: RS232, RS422, RS449, RS485, V35, and EIA530. These two programmable interfaces can be used as two standard serial interface channels; or, a total of twelve differential drivers and eight differential receivers can be manipulated by the User FPGA.

- Two 8-bit open collector transceivers controlled as either input or output on an 8-bit basis.

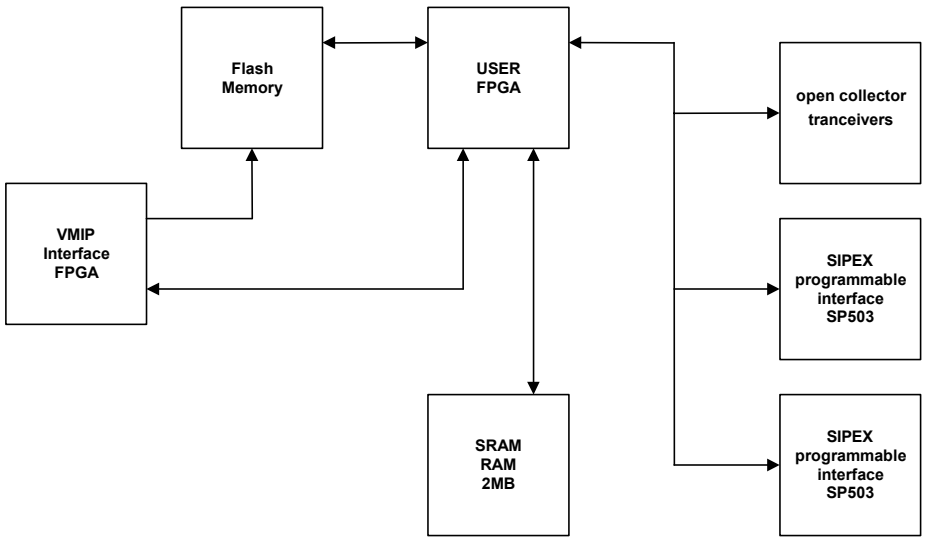


FIGURE 1-2 VM6069 BLOCK DIAGRAM

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

VM6069 FEATURES

FEATURES	
INTERFACE FPGA	
	Provides startup and control interface to the User FPGA. Provides controls to download or read flash memory.
FLASH MEMORY	
	Flash memory has 4 Mbits of memory. It is split into a code and data area. In the code area, the User FPGA code is loaded. In the data area, user defined data patterns can be loaded. Flash memory is written from the VXIbus via the Interface FPGA.
USER FPGA	
	The User FPGA is the heart of the system and provides the hardware interface to all the peripherals and memory. This also has buffered VMIP signals so that the device dependent registers can access it. On power up, it reads the configuration from flash code memory automatically. The User FPGA can read the data area of flash memory if the user designs his code to do so.
SRAM	
	Static RAM has 2 MB of memory. Both word and byte access is possible. Use of this memory depends on the User FPGA design code.
Peripherals	
	The VM6069 provides various flexible I/Os. The use of these I/Os depends on the User FPGA code design.
SIPEX Drivers & Receivers	There are twelve drivers and eight receivers. They can be configured in RS232/RS422 and RS485.
Open Collector I/O	There are sixteen open collector I/Os that are byte configurable in input or output mode.

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

TABLE 1-1 VM6069 GENERAL SPECIFICATIONS

User FPGA	✓ 13k gates. Xilinx XC4013XL
RAM	✓ 2 Mbytes
Serial Drivers/Receivers	✓ 12 drivers, 8 receivers ✓ 2 channels. Sipex 503CF ✓ 5 Mbs data rate ✓ Programmable RS232, RS422, RS449, RS485, V35, EIA530
Open Collector Transceivers	✓ 16 channels, programmable on an 8-bit basis as TTL inputs or open collector outputs ✓ 10 MHz data rate

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

SECTION 2

PREPARATION FOR USE

INSTALLATION

When the VM6069 is unpacked from its shipping carton, the contents should include the following items:

VM6069 Custom Protocol Interface Module
(1)VM6069 Custom Protocol Interface Module User's Manual (this manual)
(2) 3 ½" diskettes: VM6069 Flash Memory Load and sample code (VTI P/N 72-0025-000)

All components should be immediately inspected for damage upon receipt of the unit.

Once the VM6069 is assessed to be in good condition, it may be installed into an appropriate C-size or D-size VXIbus chassis in any slot other than slot zero. The chassis should be checked to ensure that it is capable of providing adequate power and cooling for the VM6069. Once the chassis is found adequate, the VM6069's logical address and the chassis' backplane jumpers should be configured prior to the VM6069's installation.

CALCULATING SYSTEM POWER AND COOLING REQUIREMENTS

The power and cooling requirements of the VM2601 are given in the specification table in Section 1 of this manual. It is imperative that the chassis provide adequate power and cooling for this module. Referring to the chassis user manual, confirm that the power budget for the system (the chassis and all modules installed therein) is not exceeded and that the cooling system can provide adequate airflow at the specified backpressure.



It should be noted that if the chassis cannot provide adequate power to the module, the instrument may not perform to specification or possibly not operate at all. In addition, if adequate cooling is not provided, the reliability of the instrument will be jeopardized and permanent damage may occur. Damage found to have occurred due to inadequate cooling will void the warranty on the instrument in question.

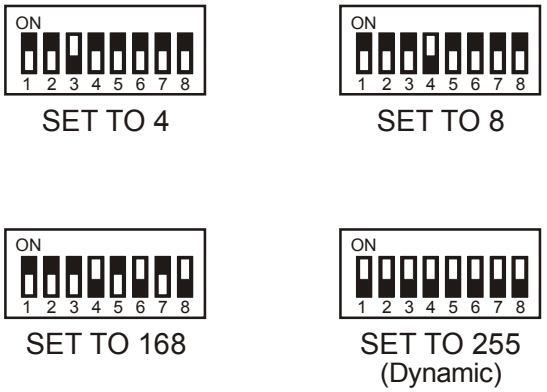
Artisan Technology Group - Quality Instrumentation ... Guaranteed | (888) 88-SOURCE | www.artisantg.com

SETTING THE CHASSIS BACKPLANE JUMPERS

Please refer to the chassis operation manual for further details on setting the backplane jumpers.

SETTING THE LOGICAL ADDRESS

The logical address of the VM6069 is set by a single 8-position DIP switch located near the VMIP module's backplane connectors (this is the only switch on the module). The switch is labeled with positions 1 through 8 and with an ON position. A switch pushed toward the ON legend will signify a logic 1; switches pushed away from the ON legend will signify a logic 0. The switch located at position 1 is the least significant bit while the switch located at position 8 is the most significant bit. See Figure 2-1 for examples of setting the logical address switch.



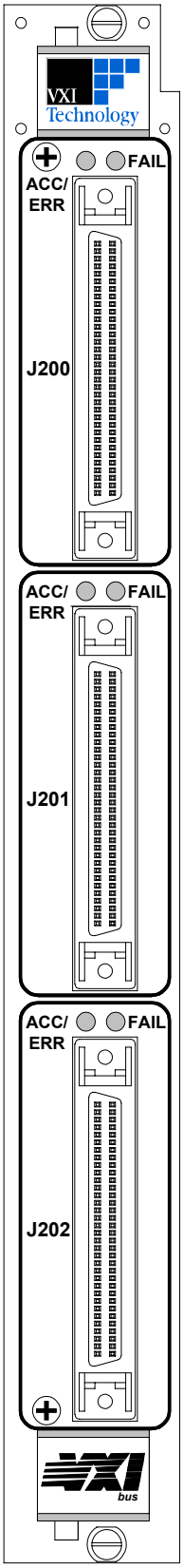
Switch Position	Switch Value
1	1
2	2
3	4
4	8
5	16
6	32
7	64
8	128

FIGURE 2-1 LOGICAL ADDRESS SWITCH SETTING EXAMPLES

The VMIP may contain three separate instruments and will allocate logical addresses as required by the VXIbus specification (revisions 1.3 and 1.4). The logical address of the instrument is set on the VMIP carrier. The VMIP logical addresses must be set to an even multiple of 4 *unless dynamic addressing is used*. Switch positions 1 and 2 must always be set to the OFF position. Therefore, only addresses of 4, 8, 12, 16, ...252 are allowed. The address switch should be set for one of these legal addresses and the address for the second instrument (the instrument in the center position) will automatically be set to the switch set address plus one; while the third instrument (the instrument in the lowest position) will automatically be set to the switch set address plus two. If dynamic address configuration is desired, the address switch should be set for a value of 255 (All switches set to ON). Upon power-up, the slot 0 resource manager will assign the first available logical addresses to each instrument in the VMIP module.

If dynamic address configuration is desired, the address switch should be set for a value of 255. (All switches set to ON). Upon power-up, the slot-0 resource manager will assign the first available logical addresses to each instrument in the VMIP module.

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



FRONT PANEL INTERFACE

The peripheral I/Os are terminated at the high-density 68-pin SCSI connector.

Regardless of whether the VM6069 is configured with other VM6069 modules or with other VMIP modules, each interface is treated as an independent instrument in the VXIbus chassis. Each has its own Unique Logical Address and, as such, its own front panel ACCESS and FAIL indicators.

FIGURE 2-2 FRONT PANEL LAYOUT

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

TABLE 2-1 VM6069 CONNECTOR PINOUTS

SCSI Conn. Pin	Signal Name	SCSI Conn. Pin	Signal Name
1	TXD1+ (OUT)	35	TXC2- (OUT)
2	TXD1- (OUT)	36	TXC2+ (OUT)
3	RXD1+ (IN)	37	RXC2- (IN)
4	RXD1- (IN)	38	RXC2+ (IN)
5	RTS1+ (OUT)	39	ST2- (OUT)
6	RTS1- (OUT)	40	ST2+ (OUT)
7	CTS1+ (IN)	41	RL2- (OUT)
8	CTS1- (IN)	42	RL2+ (OUT)
9	DTR1+ (OUT)	43	R1IN+ (QQ0)
10	DTR1- (OUT)	44	R1IN- (QQ1)
11	DSR1+ (IN)	45	R2IN+ (QQ2)
12	DSR1- (IN)	46	R2IN- (QQ3)
13	TXC1+ (OUT)	47	R3IN+ (QQ4)
14	TXC1- (OUT)	48	R3IN- (QQ5)
15	RXC1+ (IN)	49	R4IN+ (QQ6)
16	RXC1- (IN)	50	R4IN- (QQ7)
17	ST1+ (OUT)	51	(NC)
18	ST1- (OUT)	52	(NC)
19	RL1+ (OUT)	53	(NC)
20	RL1- (OUT)	54	(NC)
21	GND	55	(NC)
22	GND	56	(NC)
23	TXD2+ (OUT)	57	(NC)
24	TXD2- (OUT)	58	(NC)
25	RXD2+ (IN)	59	GND
26	RXD2- (IN)	60	GND
27	RTS2+ (OUT)	61	DOUT1 (QQ8)
28	RTS2- (OUT)	62	DOUT2 (QQ9)
29	CTS2+ (IN)	63	DOUT3 (QQ10)
30	CTS2- (IN)	64	DOUT4 (QQ11)
31	DTR2+ (OUT)	65	DOUT5 (QQ12)
32	DTR2- (OUT)	66	DOUT6 (QQ13)
33	DSR2+ (IN)	67	DOUT7 (QQ14)
34	DSR2- (IN)	68	DOUT8 (QQ15)

The pin locations for J200, J201 and J202 are shown in Figure 2-3. Contact the factory for information on mating connectors.

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

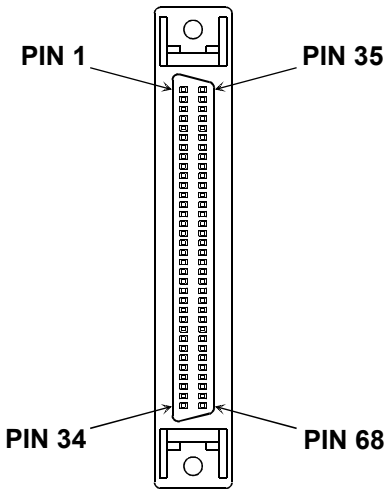


FIGURE 2-3 J200, J201 AND J202 PIN LOCATIONS

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

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

SECTION 3

PROGRAMMING

INTRODUCTION

The VM6069 is a register-based module that allows for fast data throughput along the VXIbus backplane. The examples in this section show how to integrate customer-defined logic patterns with what is required to operate the VM6069. Services are also available through VXI Technology's custom engineering services to develop the VHDL/VERILOG code from protocol information.

The protocol to be executed by this card can be downloaded to flash memory from the VXI controller through the registers. The module control, module status and FIFO access are all done through the registers. These registers are mapped to device dependent registers.

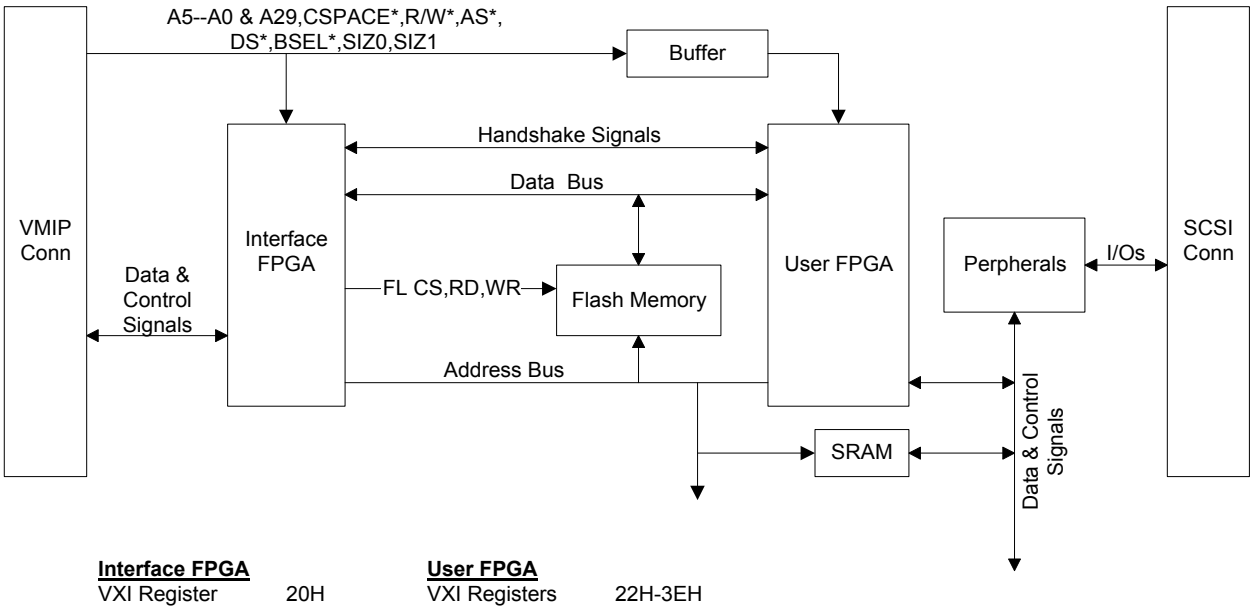


FIGURE 3-1 VM6069 UNIVERSAL SERIAL INTERFACE

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

INTERFACE FPGA

The Interface FPGA (QL2003-1TQFP144C) handles the interface between the VMIP and the VM6069. The Interface FPGA generates the necessary hand-shaking signals to the VMIP. It provides the VXIbus access to flash memory (AM29F040) for loading or changing the Protocol FPGA dynamically.

The figure on the next page describes the Interface FPGA design. The user can not change this design, it is given only for reference.

The Interface FPGA uses one register of the device dependent registers (20H). The rest of the device dependent registers (22H-3EH) may be decoded and used in the User FPGA design.

The Interface FPGA handles the VXI/VMIP and flash memory interface. It allows read/write access to flash memory from the VXI backplane. It also provides buffered VXI/VMIP signals to the User FPGA.

USER FPGA

The User FPGA (XILINX XC4013XL) is used for protocol execution. The configuration data for this device is loaded from flash memory by the master parallel configuration mode. This allows changing the VHDL/VERILOG design code dynamically for different protocol FPGA designs by loading flash memory through the VXIbus. The reset bit should be asserted and de-asserted after loading new VHDL/VERILOG design code to flash memory, which will force the device into configuration mode.

If the User FPGA has to transmit data, data would be written to SRAM through the VXIbus. The User FPGA then reads the data from SRAM and transmits it through the external interface.

If the User FPGA has to receive data, the User FPGA will receive the data from the external interface and then writes the data into SRAM. Data can then be read from SRAM through the VXIbus.

The 2 Mbytes of SRAM can be used as FIFO. The FIFO logic to interface the SRAM is emulated in the User FPGA.

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

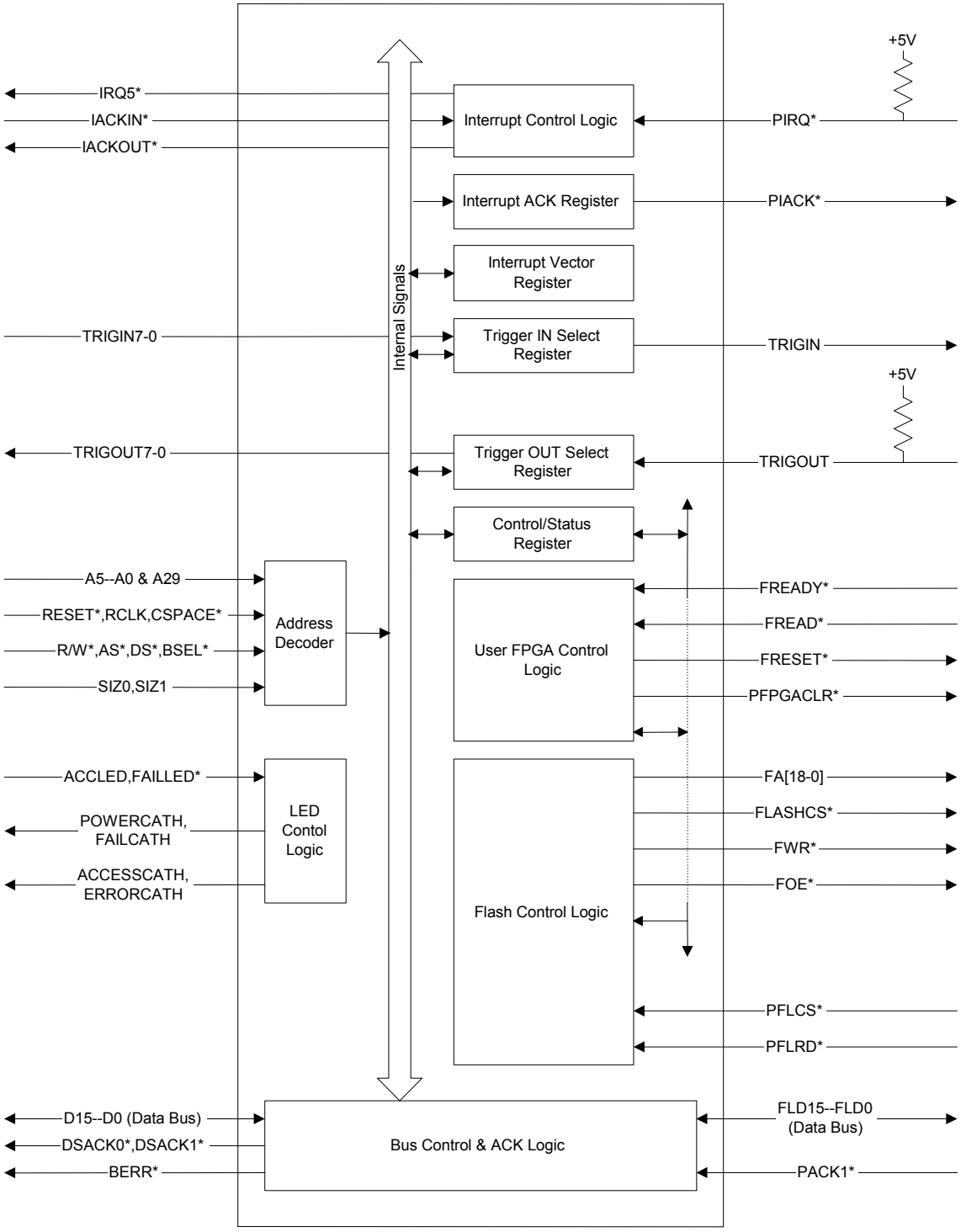


FIGURE 3-2 VM6069 INTERFACE FPGA

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

REGISTER ACCESS

The module is accessed as a register-based VXI card. The device dependent registers (20H to 3EH) are available for use.

TABLE 3-1 A16 ADDRESS SPACE - REGISTER MAP

Address in Hex	Function
22 - 3E	User FPGA Access Registers (Depends on the User FPGA Code)
20	Interface FPGA Control / Status Register
1E	
1C	
1A	
18	
16	[A32 Pointer Low]
14	[A32 Pointer High]
12	[A24 Pointer Low]
10	[A24 Pointer High]
E	Data Low
C	Data High
A	Response [/Data Extended]
8	Protocol [/Signal] Register
6	[Offset Register]
4	Status / Control Register
2	Device Type
0	ID Register

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

TABLE 3-2 CONTROL / STATUS REGISTER (20H)

Bit	Function		Access Type
0 - 7	Flash Data (Bit-0 is D0 / Bit-7 is D7)		Read / Write
8	User FPGA Ready	1 Ready, 0 Not Ready	Read Only
9	Reset User FPGA	1 Reset, 0 Not Reset	Read / Write
10	Clear User FPGA	1 Clear, 0 Not Clear	Read / Write
11	Code / Data Select	1 Flash Code, 0 Flash Data	Read / Write
12-15	See Table 3-3		

User FPGA Ready: This bit is high once the User FPGA reads flash memory and configures itself. A low on this bit indicates that the User FPGA configuration has not been completed. Configuration failure may be caused by the following:

1. Flash memory does not have the valid code for the User FPGA.
2. An electrical fault has occurred in the User FPGA related circuits.

Reset User FPGA: A high on this bit forces the User FPGA into a "sleep" mode and the next falling edge causes the User FPGA to read Flash memory and configure itself.

Clear User FPGA: This is similar to the Reset User FPGA bit, but its use depends on the User FPGA design. Writing to this bit causes a low going pulse on the PFPGACLR* signal to the User FPGA. This can be used as a system reset in the User FPGA design; however, it does not force the User FPGA to read flash memory.

Code / Data Select: A high on this bit selects the lower half of flash memory, and a low on this bit selects the upper half of flash memory. The FPGA design code needs to be loaded into the lower half of flash memory.

Flash Data: When a write is made, these bits (one byte) are written to flash memory. When a read is made, these bits read flash memory data. See the next table for addressing flash memory.

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

TABLE 3-3 CONTROL / STATUS REGISTER (20H) - BITS 12-15

Bits 12 to 15 for Flash Access Control from VXIbus						
Hex Value	**Bit 15**	**Bit 14**	**Bit 13**	**Bit 12**	**Function**	**Access Type**
0	0	0	0	0	No Access for VXI Bus	Read / Write
1	0	0	0	1	Reset Address Counter - *see note*	Read / Write
2	0	0	1	0	Increment Address Counter - *see note*	Read / Write
3	0	0	1	1	Spare 1	Read / Write
4	0	1	0	0	Access Sequential Address	Read / Write
5	0	1	0	1	Access Address 5555H	Read / Write
6	0	1	1	0	Access Address 2AAAH	Read / Write
7	0	1	1	1	Spare 2	Read / Write
8	1	0	0	0	Sector 0 - Addr 00000-0FFFF	Read / Write
9	1	0	0	1	Sector 1 - Addr 10000-1FFFF	Read / Write
A	1	0	1	0	Sector 2 - Addr 20000-2FFFF	Read / Write
B	1	0	1	1	Sector 3 - Addr 30000-3FFFF	Read / Write
C	1	1	0	0	Sector 4 - Addr 40000-4FFFF	Read / Write
D	1	1	0	1	Sector 5 - Addr 50000-5FFFF	Read / Write
E	1	1	1	0	Sector 6 - Addr 60000-6FFFF	Read / Write
F	1	1	1	1	Sector 7 - Addr 70000-7FFFF	Read / Write

NOTE: Address counter reset and increment is done only when the bits are written to. If a read is done when these bits are set, it uses the current address to read the flash memory.

These bits are used to access the flash memory. For normal operation (for the User FPGA to function), the **No Access for VXI Bus** function needs to be selected. A flash load utility is provided to download the User FPGA design (.EXO format) to flash memory. The utility loads the design into the lower half of flash memory. The user defined pattern (data that is not part of the design) needs to be loaded into the upper half of flash memory. Refer to this table to access the upper half of flash memory.

No Access for VXI Bus: This disables access to flash memory from the VXI backplane. For normal module operation, these bits need to be zero.

Addressing Flash Memory: A flash memory address is generated by the Interface FPGA when accessed from the VXI backplane. It uses the internal address counter (A0-A18 to access the flash memory. This address counter can be reset (resetting the address counter points it to the first location in flash memory), and incremented by writing to these bits. Only sequential flash memory access is available from the VXI backplane; however, two pre-defined address locations (5555H and 2AAAH) can be accessed independent of the address counter (see above table). In addition, individual sectors can be selected independent of the address counter. This sector addressing is only useful for erasing the specified sector; it cannot be used to read or write to flash memory.

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

FLASH MEMORY

The flash memory (AM29F40) has 4M bit (524,228 x 8 bits) divided into eight sectors; each sector is 64KB. The User FPGA (XC4013XL) needs 393,623 bits of configuration data. This is less than one sector of the flash memory. Four sectors (0-3) are allocated for the User FPGA and four sectors (4-7) are allocated for data pattern storage. This means there are more than three sectors available for future expansion of the FPGA configuration.

FLASH - READ / WRITE

The flash memory address is generated by the Interface FPGA and it depends on the control/status register. The required control bits are provided to select the code/data memory area: to access the memory sequentially, and to access the memory in predefined addresses, so as to specify the various command sequences.

Erasing, downloading and reading flash can be done per the following sequence. Though multiple sectors can be erased simultaneously, each sector should be erased one at a time. The time between two sector addresses (to erase two sectors at once) is 87µs max., which is too time constraining for most programming environments.

DOWNLOADING USER FPGA CODE / DATA PATTERN

The following steps explain the flash memory downloading sequence; however, it is recommended to use the software utility provided to download the User FPGA design (.EXO format) to flash memory.

1. Put the User FPGA in reset mode
   (a) Write 0200H to register 20H

2. Do Chip Reset
   (a) Write 42F0H to register 20H

3. Check Manufacturer Code and Device Code

(a)	Write 52AAH to register 20H	*Writes AAH to address 5555H*
(b)	Write 6255H to register 20H	*Writes 55H to address 2AAAH*
(c)	Write 5290H to register 20H	*Writes 90H to address 5555H*
(d)	Write 1200H to register 20H	*Reset the address counter (to select even address)*
(e)	Read from register 20H	*The data should be 01H (manufacturer code)*
(f)	Write 2200H to register 20H	*Increment address counter (to become odd address)*
(g)	Read from register 20H	*The data should be A4H (device code)*

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

4. Do Chip Erase / Sector Erase

(a)	Write 1200H to register 20H	*Reset the address counter*
(b)	Write 52AAH to register 20H	*Writes AAH to address 5555H*
(c)	Write 6255H to register 20H	*Writes 55H to address 2AAAH*
(d)	Write 5280H to register 20H	*Writes 80H to address 5555H*
(e)	Write 52AAH to register 20H	*Writes AAH to address 5555H*
(f)	Write 6255H to register 20H	*Writes 55H to address 2AAAH*

After step 4f, select one of the following steps - I, II or III - and proceed further:

I. Erase all sectors - Chip Erase

(g)	Write 5210H to register 20H	*Writes 10H to address 5555H*
(h)	Read from register 20H	*Check that data-bit 7 is '1'. Read data until data-bit 7 is '1'. After this step, go to step 5a.*

II. Erase sectors 0-3 - For down loading **code**

Each sector has to be erased one by one. Send one erase command, poll the data to check that erasing is complete, and then erase the next sector. Pick one step from *g - j*, then do step *k*.

(g)	Write 8230H to register 20H	*Select sector 0, the data is 30H*
(h)	Write 9230H to register 20H	*Select sector 1, the data is 30H*
(i)	Write A230H to register 20H	*Select sector 2, the data is 30H*
(j)	Write B230H to register 20H	*Select sector 3, the data is 30H*
(k)	Read from register 20H	*Check that data-bit 7 is '1'. Read data until data- bit 7 is '1'. Do not change register 20H in the mean time because it has to poll the data in the same sector. If all the sectors (0-3) are erased proceed to step 5a else go back to step 4a*

II. Erase sectors 4-7 - For down loading **data**

Each sector has to be erased one by one. Send one erase command, poll the data to check that erasing is complete, and then erase the next sector. Pick one step from *g - j*, then do step *k*.

(g)	Write C230H to register 20H	*Select sector 4, the data is 30H*
(h)	Write D230H to register 20H	*Select sector 5, the data is 30H*
(i)	Write E230H to register 20H	*Select sector 6, the data is 30H*
(j)	Write F230H to register 20H	*Select sector 7, the data is 30H*
(k)	Read from register 20H	*Check that data-bit 7 is '1'. Read data until data- bit 7 is '1'. Do not change register 20H in the mean time because it has to poll the data in the same sector. If all the sectors (4-7) are erased proceed to step 5a else go back to step 4a*

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

5. Download Code / Data

   (a) Reset Flash Address Counter

Write 1200H to register 20H	*Reset the address counter*

   (b) Select Code Area or Data Area

Write 0A00H to register 20H	*To select code area*
Write 0200H to register 20H	*To select data area*

   Choose either the code area or the data area.
   Both areas are not simultaneously available.

   (c) Send program command

Write 52AAH to register 20H	*Writes AAH to address 5555H*
Write 6255H to register 20H	*Writes 55H to address 2AAAH*
Write 52A0H to register 20H	*Writes A0H to address 5555H*

   (d) Write single byte to current address

Write 42XXH to register 20H	*Where XX is the data*

   (e) Poll the data from the current address

Read from register 20H	*Bits D0-D7 represent the data. Check that data-bit 7 is the same as the written bit. If they are not the same, then read and check again. Continue to poll the data until data-bit 7 is the same as that of the written bit.*

   (f) Increment Address Counter

Write 2200H to register 20H	*Increments the address counter*

   (g) Continue step c. *Alternatively, if the download is completed, do a chip reset again (step 2).*

6. Read back the data to check whether it is written correctly.

   (a) Reset Flash Address Counter

Write 1200H to register 20H	*Resets the address counter*

   (b) Select Code Area or Data Area

Write 0A00H to register 20H	*To select code area*
Write 0200H to register 20H	*To select data area*

   Choose either the code area or the data area.
   Both areas are not simultaneously available.

   (c) Send Read / Reset Command

Write 52AAH to register 20H	*Writes AAH to address 5555H*
Write 6255H to register 20H	*Writes 55H to address 2AAAH*
Write 52F0H to register 20H	*Writes F0H to address 5555H*

   (d) Read single byte from current address

Read from register 20H	*Bits D0-D7 represent the data*

   (e) Compare the data with the written data

   (f) Increment the Address Counter

Write 2200H to register 20H	*Increments the address counter*

   (g) Continue step d. *Alternatively, if the comparison (read) is completed, then do a chip reset again (step 2).*

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

7. Put the User FPGA in normal mode and set flash access bits to ‘No Access for VXIbus’ status.
   (a) Write 0000H to register 20H

Notes:

1. *When reading from flash memory, steps d &f can be contiguous if the comparison is not required (i.e. to reach a particular address).*

2. *If only reading from flash is required, follow steps 1, 6a, 6b & 6c, then follow steps 6d to 6g to read the required code/data. Finish with steps 2 and 7.*

3. *Any digit represented by an XX indicates data to be written.*

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

USER FPGA DESIGN

The User FPGA is the heart of the system and provides the hardware interface to all the peripherals and memory. This also has buffered VMIP signals so that the device dependent registers can access it. On power-up, it reads the configuration from the flash code memory area and configures itself. The User FPGA can read the data area of flash memory if the user designs the code as such. The functioning of the board entirely depends on the User FPGA code that is designed by the user.

The User FPGA design is developed in VHDL/Verilog and converted to a **.EXO** hex file. This file is then downloaded to flash memory through the VXI backplane. A software utility is provided to download the **.EXO** hex file to flash memory. Since only a small area of flash memory is used for the User FPGA design, the rest of flash memory may be used to store user defined data.

USER FPGA DESIGN GUIDELINES

Use the following guidelines to get started with your User FPGA design:

1. Familiarize yourself with the User FPGA signals - see Table 3-4.

2. Familiarize yourself with the Interface FPGA, VMIP Interface, memory, Sipex latches and open collector buffers - see figures in this section.

3. Convert your application requirement to Verilog/VHDL/schematics in the Xilinx foundation series or Alliance series.

4. The User FPGA design may make use of the memory, Sipex drivers, receivers or the open collector buffers:

   Note: It is necessary to generate a data transfer acknowledge whenever a VXI backplane access is made. If the module is accessed to any device dependent registers (22H - 3EH), and the User FPGA does not acknowledge this, the front panel will indicate a system fail has occurred (red LED).

5. Down load the design to flash memory using the software utility provided.

6. Now the module is ready to function per the User Design loaded in flash memory.

7. Use registers 22H - 3EH (per the design) to transfer information (memory, I/O, etc.) between the VXI backplane and the module.

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

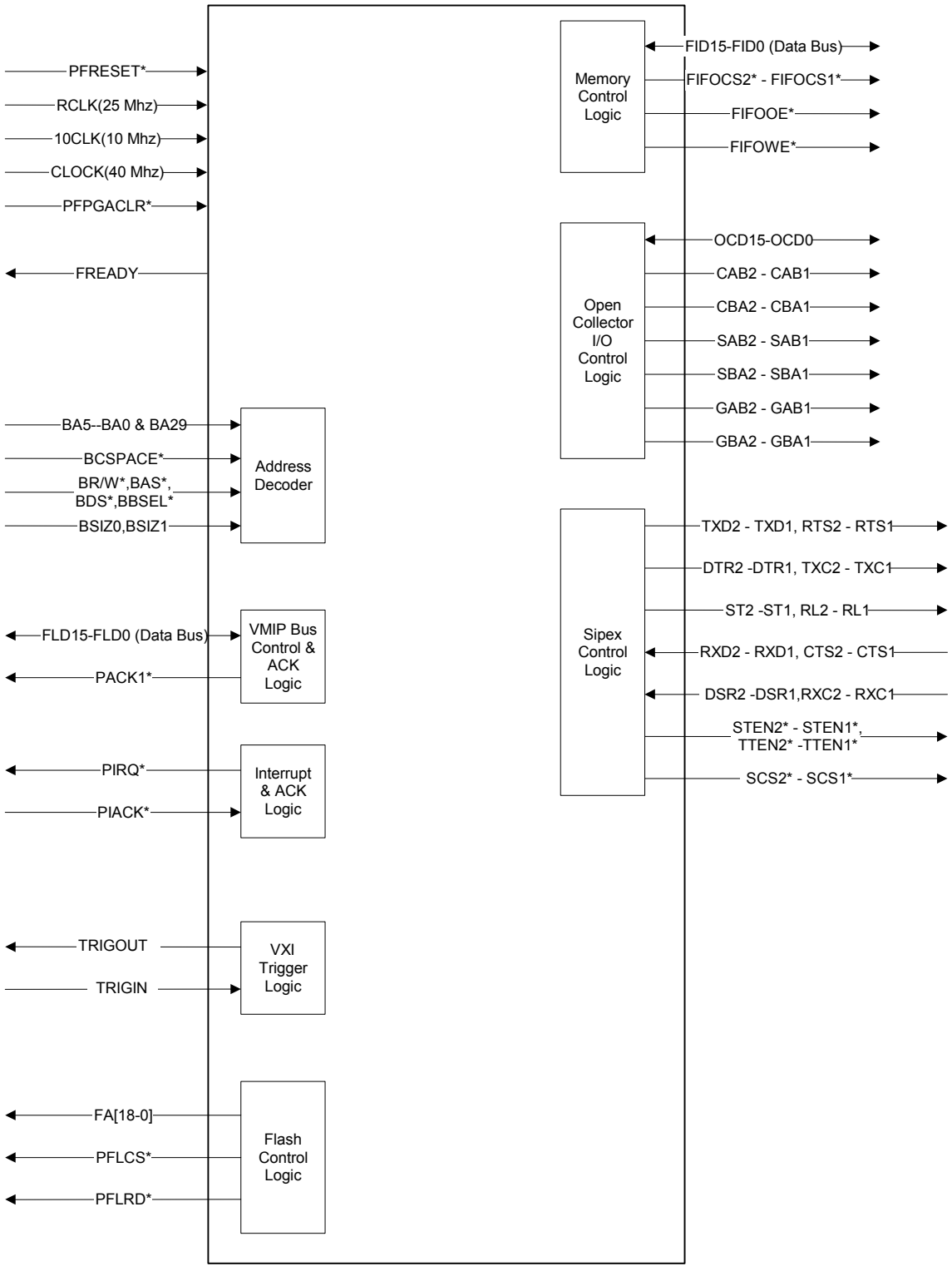


FIGURE 3-3 USER FPGA

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

USER FPGA SIGNAL DESCRIPTION

The following table describes the User FPGA signals. Most of these signals are available for design use. The active low signal is marked with a 'N' or a '*' (asterisk) symbol. The Signal Direction column indicates whether the signal is an input or an output:

I = Input to User FPGA
O = Output from User FPGA

Pin numbers are listed respective to their signal names.

TABLE 3-4 USER FPGA SIGNALS

Signal Name	Pin No.	Signal Direction	Description	Comments
PFRESET*	122	I	Reset signal for the User FPGA	On the rising edge of this signal, the User FPGA reads flash memory and configures itself. **This pin is not available to the user.** The user design that needs a system reset can use the signal PFGACLR* (see below).
RCLK	63	I	25 MHz VMIP clock	Any VMIP interface logic would use this clock for timing.
10CLK	107	I	10 MHz VXI clock	This is the VXI backplane clock for general use.
CLOCK	57	I	40 MHz clock for general use	For high-speed applications, this clock can be used for control to access peripherals. However, the 25 MHz clock should still be used for the VMIP interface.
PFPGACLR*	25	I	User FPGA Clear Signal from the Interface FPGA	This signal can be used to clear all the internal registers and user defined variables in the User FPGA. This signal is active low when the system reset is low, or when the clear control bit is set in the control register.
FREAD*	120	O	User FPGA read signal to the Interface FPGA	The User FPGA generates this signal during configuration. **This pin is not available to the user.**
FREADY	68	O	User FPGA Ready signal to the Interface FPGA	The User FPGA generates this signal once it reads the code and configures itself. **This pin is not available to the user.**
BA5-BA0 & BA29	218, 104, 198, 111, 115, 170, 11	I	VMIP Address lines	Buffered VMIP address lines for register address decoding.
BCSPACE*	4	I	VMIP signal for address decoding	Buffered. Must be high for address decoding.

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

User FPGA Signals – (Cont.)

Signal Name	Pin No.	Signal Direction	Description	Comments
BR/W*	208	I	VMIP read/write signal	Buffered. High for read, low for write.
BAS*	9	I	VMIP address strobe	Buffered. Should be low for address decoding.
BDS*	67	I	VMIP data strobe	Buffered.
BBSEL*	65	I	VMIP board select	Buffered VMIP board select signal. Should be low for address decoding.
BSIZ0, BSIZ1	176, 224	I	VMIP bus width	Buffered VMIP bus width signal for byte or word access.
FLD15-FLD0	36, 138, 132, 127, 50, 47, 46, 49, 123, 129, 141, 148, 152, 159, 173, 177	I/O	VMIP data bus	Buffered.
PACK1*	230	O	Bus acknowledge	Bus acknowledge signal to VMIP from the User FPGA. The User FPGA should generate this signal upon completion of the data transfer between the VMIP and the User FPGA. This signal should go low within 20 RCLK clocks otherwise the Interface FPGA will generate a time-out and the current bus access is terminated. The time-out signal lights the System Fail LED and will remain lit until the next system reset. However, further bus access is allowed.
PIRQ*	13	O	Interrupt	This active low signal generates the interrupt IRQ5 to the VMIP motherboard. When the motherboard completes the interrupt acknowledge cycle with the VXI Controller end-software, the Interface FPGA generates the PIACK* pulse signal to the User FPGA.
PIACK*	56	I	Interrupt acknowledge	This signal is used as described above in the PIRQ* signal description.
TRIGIN	207	I	VXI Trigger Input line	Before using this signal, one of the backplane trigger lines (7-0) needs to be selected by word serial command. See section 4.
TRIGOUT	17	O	VXI Trigger Output line	This signal is used to generate a signal on one of the VXI bus TTLTRG lines. See Section 4, the output commands, which deal with this signal. Polarity of this signal is determined by a word serial command.

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

User FPGA Signals – (Cont.)

Signal Name	Pin No.	Signal Direction	Description	Comments
FA18-FA0	216, 3, 2, 239, 238, 233, 232, 221, 220, 214, 213, 210, 209, 203, 202, 188, 187, 184, 183	O	Address lines	Address lines for flash memory and SRAM. Used by both the User FPGA and the Interface FPGA. When this signal is used by the Interface FPGA, it puts the User FPGA in reset mode.
PFLCS*	163	O	Flash Chip select	Flash Chip select generated by the User FPGA.
PFLRD*	160	O	Flash read select	Flash read select generated by the User FPGA.
FID15-FID0	15, 16, 167, 18, 164, 21, 156, 165, 154, 27, 52, 34, 147, 144, 142, 39	I/O	Data bus	Data bus to access the SRAM and other peripherals solely used by the User FPGA.
FIFOCS1*, FIFOCS2*	87, 189	O	SRAM chip select signals	FIFOCS1* is for the first half of the 2MB of SRAM; FIFOCS2* is for the second half. These signals are generated by the User FPGA.
FIFOOE*	186	O	SRAM read signal	This signal is generated by the User FPGA.
FIFOWE*	236	O	SRAM write signal	This signal is generated by the User FPGA.
OCD15-OCD0	95, 84, 200, 237, 12, 102, 169, 23, 55, 24, 155, 162, 31, 149, 146, 33	I/O	Open collector buffer data lines	These signals are generated or received by the User FPGA, depending on the open collector control line.
CAB1, CAB2, CBA1, CBA2, SAB1, SAB2, SBA1, SBA2, GAB1, GAB2, GBA1*, GBA2*	53, 114, 168, 117, 171, 174, 54, 172, 73, 112, 206, 113	O	Open collector buffer control lines	CAB1, CBA1, SAB1, SBA1, GAB1 and GBA1* are used for the lower data byte (D7-D0). CAB2, CBA2, SAB2, SBA2, GAB2 and GBA2* are used for the upper data byte (D15-D8). These signals are generated by the User FPGA.
TXD1, TXD2, RTS1, RTS2, DTR1, DTR2, TXC1, TXC2, ST1, ST2, RL1, RL2	81, 43, 76, 131, 77, 136, 125, 38, 191, 231, 93, 26	O	SIPEX driver signals	12 drivers. These signals are generated by the User FPGA.
TTEN1*, TTEN2*, STEN1*, STEN2*	194, 225, 228, 205	O	Enable signals	Enable signals for the TXC1, TXC2, ST1 and ST2 drivers respective. If an enable bit is high, the respective driver goes into tri-state. If the enable bit is low, the respective driver is in output mode.

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

User FPGA Signals – (Cont.)

Signal Name	Pin No.	Signal Direction	Description	Comments
RXD1, RXD2, CTS1, CTS2, DSR1, DSR2, RXC1, RXC2	72, 109, 126, 137, 69, 41, 110, 134	I	SIPEX receiver signals	8 receivers. These signals are to be received by the User FPGA.
ISENSE8-ISENSE1	199, 71, 7, 197, 190, 193, 92, 99	Unused	Spare	Unused in current design Reserved for future expansion
OE8-OE1	139, 133, 103, 108, 97, 44, 86, 88	Unused	Spare	Unused in current design Reserved for future expansion
PD8-PD1	48, 28, 157, 32, 35, 51, 128, 42	Unused	Spare	Unused in current design Reserved for future expansion
PDRPD*	235	Unused	Spare	Unused in current design Reserved for future expansion
DA3-DA1	223, 215, 100	Unused	Spare	Unused in current design Reserved for future expansion
LD2-LD1	175, 105	Unused	Spare	Unused in current design Reserved for future expansion
DRD	185	Unused	Spare	Unused in current design Reserved for future expansion
DCS3-DCS1	217, 229, 116	Unused	Spare	Unused in current design Reserved for future expansion
DPSP3-DPSP1	192, 153, 20	Unused	Spare FPGA interconnect	These signals are spares connection between the User FPGA and the Interface FPGA. Presently unused.
PSP1, PSP2, PSP3, PSP4	66, 10, 5, 226	Unused	Spare User FPGA connection	Spare I/O pins from the User FPGA brought out to jumper block JP3.
PFINIT*	89	I	FPGA specific	Non-user definable pin. Pulled high on PCB. Do not use.
PTCLK	118	I	Global clock input	This signal has been brought out to jumper block JP3. It may be used to route a user specified clock source into the User FPGA.
PR8-PR1	85, 130, 70, 95, 79, 82, 78, 74	Unused	Unused	Presently unused.
LL1, LL2	94, 145	Unused	Unused	Presently unused.
SCS1*, SCS2*	234, 8	O	Sipex Driver Latch Signals	Rising edge latches.

Note: Any signals not used inside the User FPGA by the user's end-design may be left unconnected in the FPGA design. The signals on the PCB are pulled to proper voltage levels to allow normal operation of the VM6069.

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

VXI REGISTER ADDRESS DECODING

There are 15 VXI device dependent registers (from 22H to 3EH) available for the User FPGA. Whenever a VXI bus cycle takes place, the VMIP converts it into a VMIP cycle for the VM6069. These registers can be decoded and used in the User FPGA as desired. For example:

REG22CS* = !(BA29 &!BA5 &!BA4 &!BA3 &!BA2 &BA1 &!BAS &!BDS &!BBSEL &!BSIZ0 &BSIZ1 &BCSPACE)	*This would decode register 22H for word access*

BA29 is always high for a backplane access. BAS is always low for a backplane access. BA5 is always low for a backplane access. BBSEL is always low for a backplane access. BCSPACE is always high for a backplane access. Register decoding should always include these signals. BA4 to BA1 are used for the individual register decoding. The BA0, BSIZ0, BSIZ1 and BDS* are used as needed. The word and byte access can be decoded as follows:

BSIZ1	BSIZ0	Type
Low	Low	No Access
Low	High	Byte Access
High	Low	Word Access
High	High	Not Used

For byte access, the address line BA0 may be used to decode the low address byte (BA0=0; FID15-FID8), and high address byte (BA0=1; FID7-FID0).

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

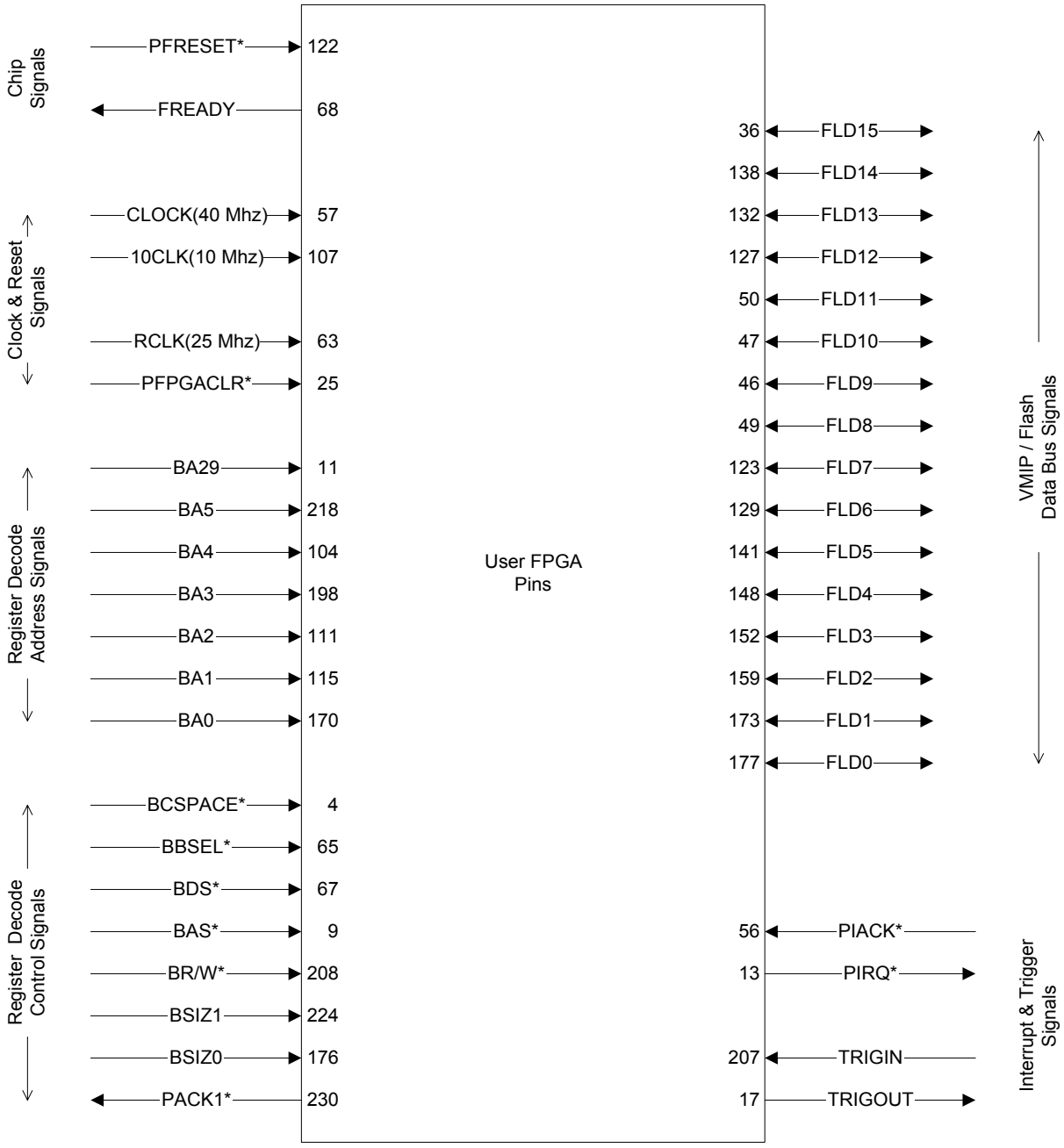


FIGURE 3-4 USER FPGA AND VMIP INTERFACE SIGNALS

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

VMIP DATA BUS INTERFACE

The data bus (FLD15-FLD0) is used to transfer the data between the VMIP and the User FPGA, and between the User FPGA and flash memory. The same data bus is used to access the flash memory from the VXI bus via the Interface FPGA. When the Interface FPGA is accessing the flash memory, it puts the User FPGA in reset mode.

VXI BUS ACKNOWLEDGE

The signal PACK1* is used as a VXI DTACK signal to acknowledge the data transfer between the User FPGA and the VXI bus. This signal is to be generated by the User FPGA. Byte and word access acknowledge is done by making PACK1* low after completion of the data access. The User FPGA should generate this signal upon completion of the data transfer between the VMIP and the User FPGA. This signal should go low within 60 RCLK clocks. If no PACK1* signal is generated in time, a bus error will occur and the system will lock up.

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

VMIP BUS TIMING

The following figures show the bus access timing between the VMIP and the User FPGA.

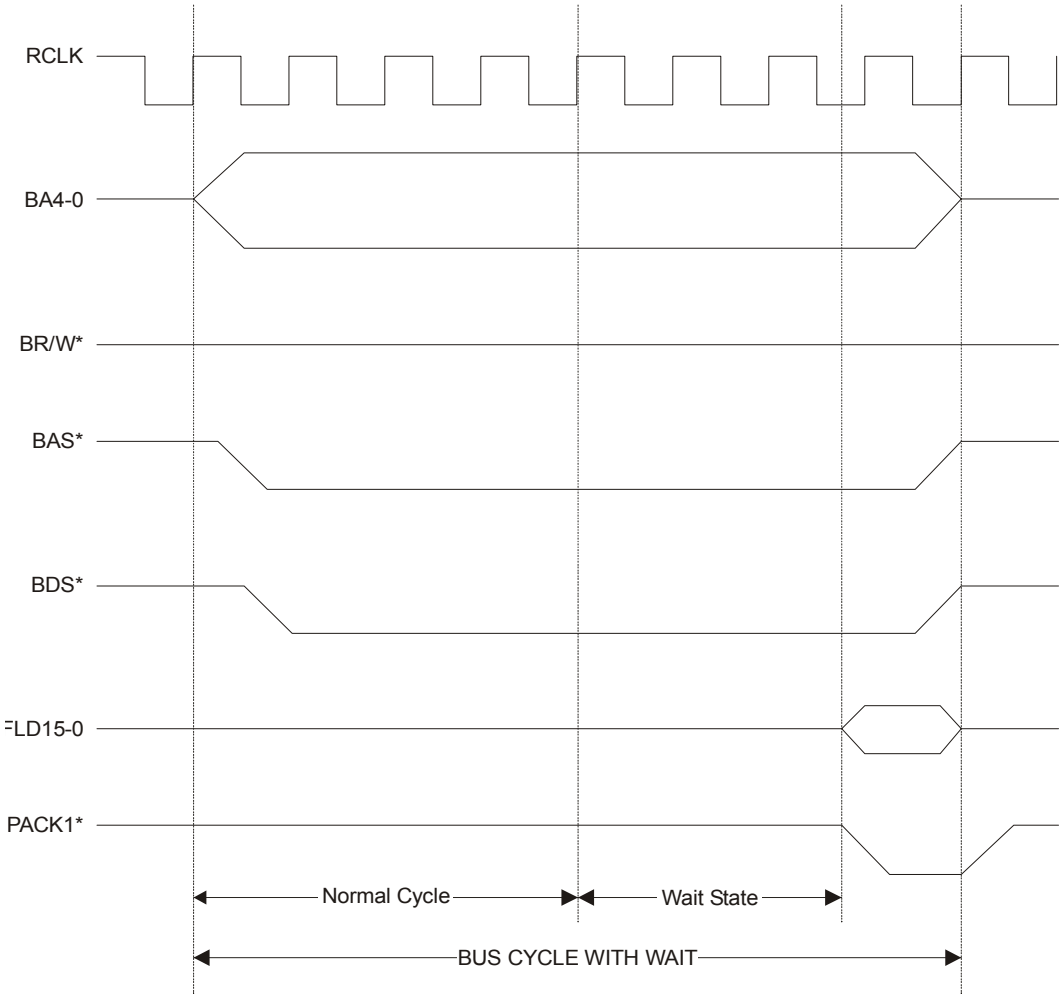


FIGURE 3-5 READ CYCLE TIMING DIAGRAM

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

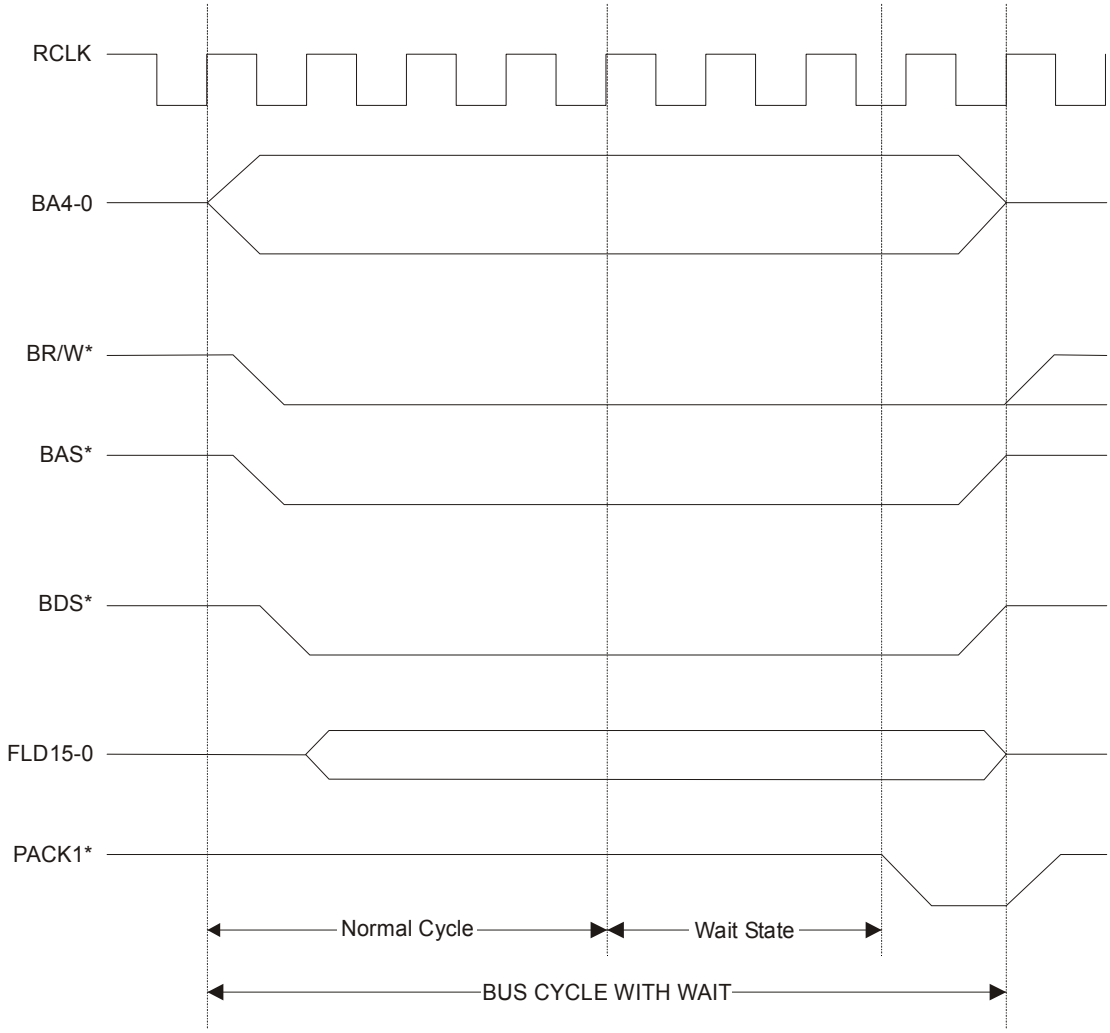


FIGURE 3-6 WRITE CYCLE TIMING DIAGRAM

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

USING THE INTERRUPT SIGNAL

The User FPGA can use the VXI interrupt IRQ5 if required. When the PIRQ* signal goes low, it generates the interrupt IRQ5 to the VMIP motherboard. Then the Interface FPGA waits for the interrupt acknowledge cycle. When it completes the interrupt acknowledge cycle (from the Controller end-software), the Interface FPGA generates the active low PIACK* pulse signal for three clock periods to indicate that the interrupt cycle is over. The PIRQ* signal is edge sensitive. A falling edge causes the generation of the interrupt. Other falling edges are ignored until PIACK* is generated. To generate the next interrupt, the PIRQ* signal should generate a low going edge again.

On getting an interrupt, the base unit (VM9000) generates a VXI backplane interrupt.

ACCESSING PERIPHERALS

SRAM

Two MB of static RAM is provided on the board for use by the User FPGA. They are arranged as two banks of memory, each bank having one MB RAM. The chip select signals (FIFOCS1* and FIFOCS2*), read signal (FIFOOE*), write signal (FIFOWE*), address lines (FA0-FA18) and data lines (FID0-FID15) are available in the User FPGA. The following table describes these signals.

TABLE 3-5 SRAM SIGNALS

FIFO1*	FIFO2*	FIFOOE*	FIFOWE*	Description
Low	High	Low	High	Read from bank1. FAx points to the address of the memory and FID has the data from the first bank of memory
Low	High	High	Low	Write to bank1. FAx points to the address of the memory and FID has the data to the first bank of memory data
High	Low	Low	High	Read from bank2. FAx points to the address of the memory and FID has the data from the second bank of memory
High	Low	High	Low	Write to bank2. FAx points to the address of the memory and FID has the data to the second bank of memory data

How this memory is used depends on the user design of the User FPGA. For more details on the SRAM regarding timing, etc., refer to data on Samsung's part number KM684000ALG-70.

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

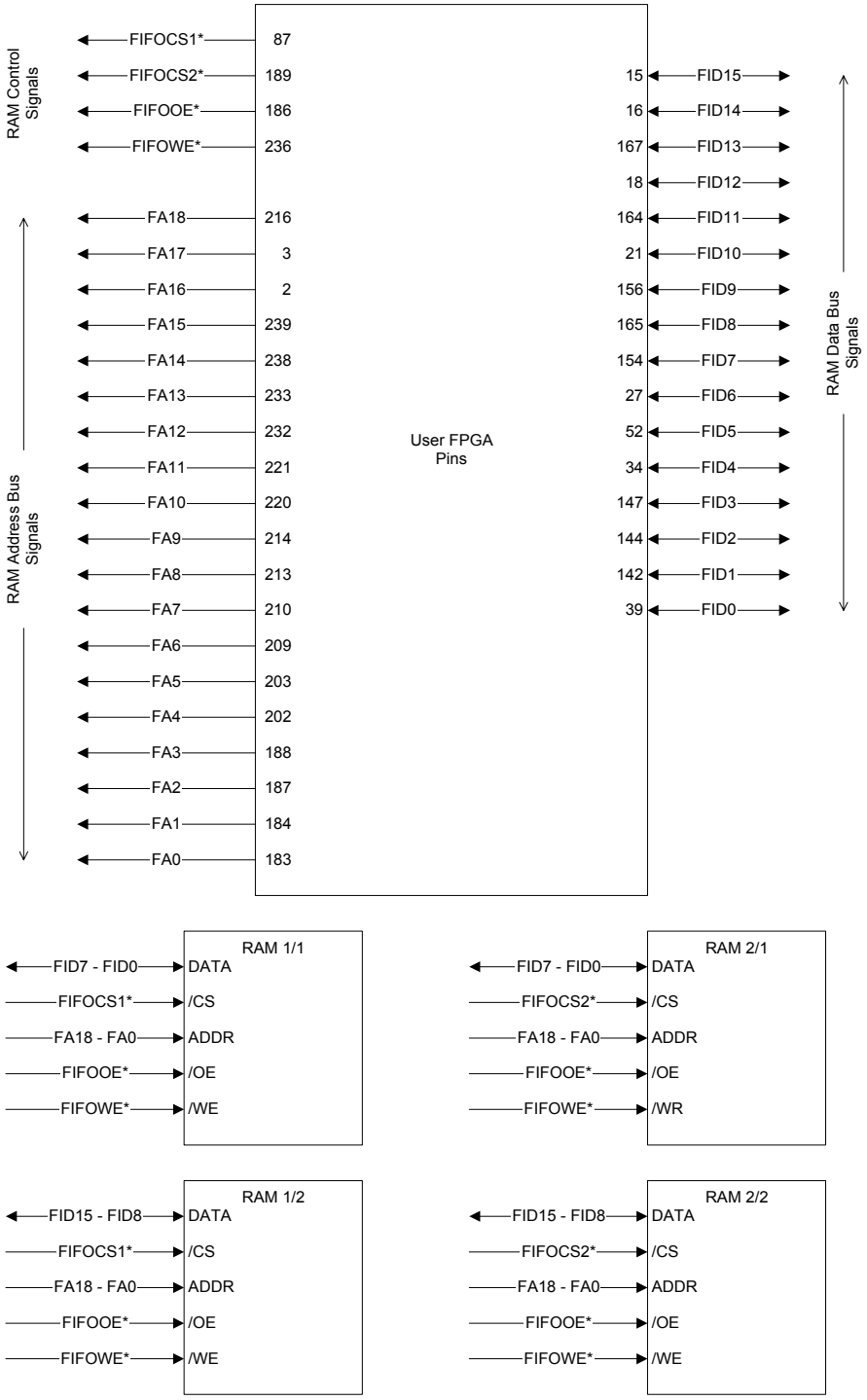


FIGURE 3-7 USER FPGA AND RAM INTERFACE SIGNALS

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

FLASH MEMORY

The flash memory is 512K x 8. The first half (256K) is allocated for the User FPGA code (Code Area), and the second half is for data (Data Area). User defined data may be stored in Data Area. Upon receiving a reset, the User FPGA reads the Code Area and configures itself. After configuration, the address lines (FA0-FA18), data lines (FLD0-FLD16, flash chip select (PFLCS*) and the read signal (PFLRD*) are available in the User FPGA. The following table describes these signals.

TABLE 3-6 FLASH MEMORY SIGNALS

PFLCS*	PFLRD*	Description
High	High	No flash access
Low	Low	FA points to the address of the flash memory and FID has the data from flash memory

For more details on flash memory regarding timing, etc., refer to data for AMD's part number AM29F040-90JC.

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

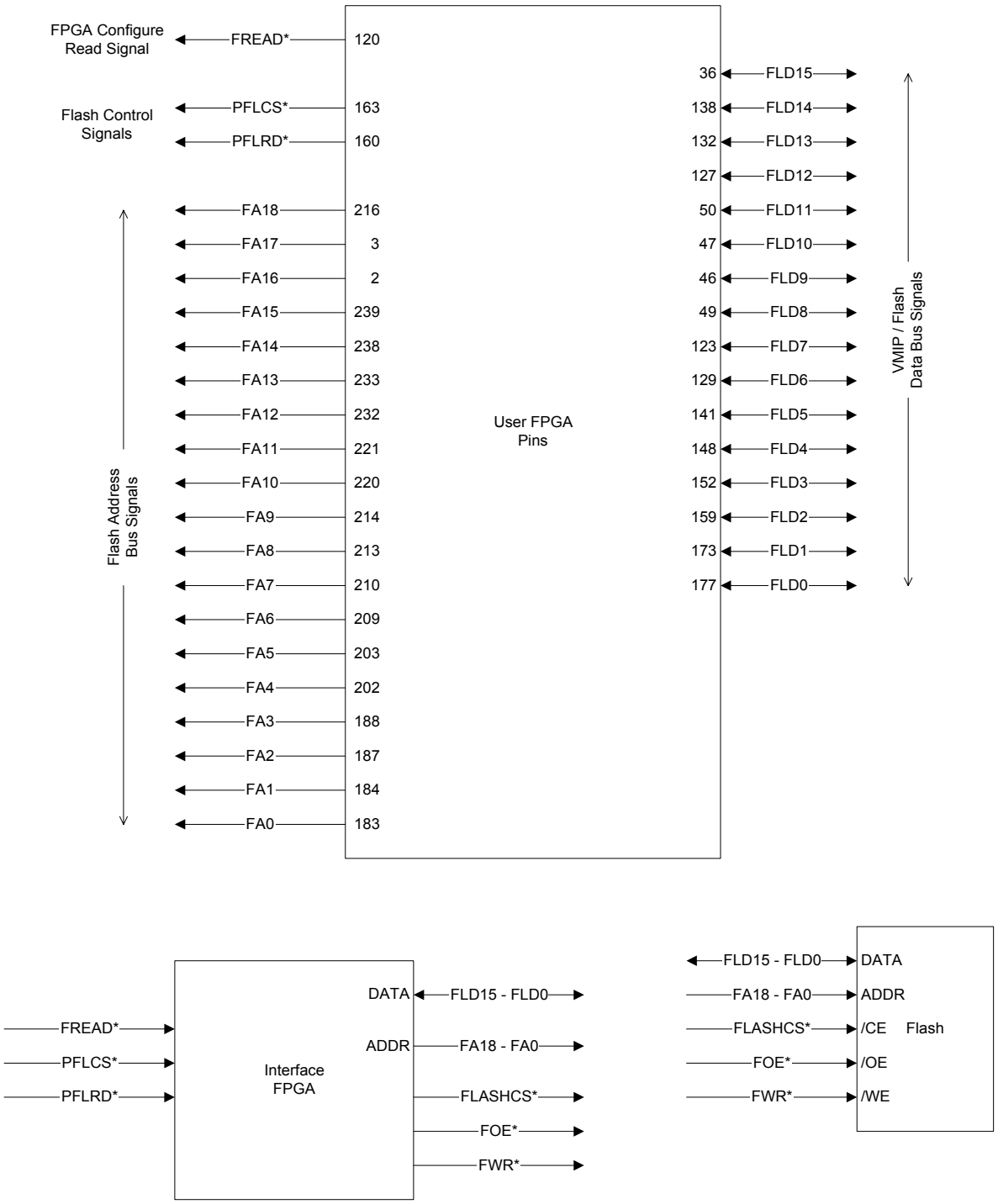


FIGURE 3-8 USER FPGA AND FLASH MEMORY INTERFACE SIGNALS

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

OPEN COLLECTOR BUFFERS

There are two Open Collector Buffers, each having eight I/O lines. Each buffer may be configured as an input or output separately. The control lines CAB1, CBA1, SAB1, SBA1, GAB1 and GBA1* are for the lower data byte buffer. CAB2, CBA2, SAB2, SBA2, GAB2 and GBA2* are for the upper data byte buffer. The A-bus side is connected to the outside world (the user connector). The B-bus side is connected to the User FPGA. The following table describes these signals.

TABLE 3-7 OPEN COLLECTOR BUFFER SIGNALS

Controls						Data I/O		Description
GAB	GBA*	CAB	CBA	SAB	SBA	A1-A8 (world)	B1-B8 (FPGA)	
L	L	L	L	H	L	Output	Input	FPGA drive the world. Real time B data to A-bus
H	H	L	L	L	H	Input	Output	FPGA reads the world. Real time A data to B-bus

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

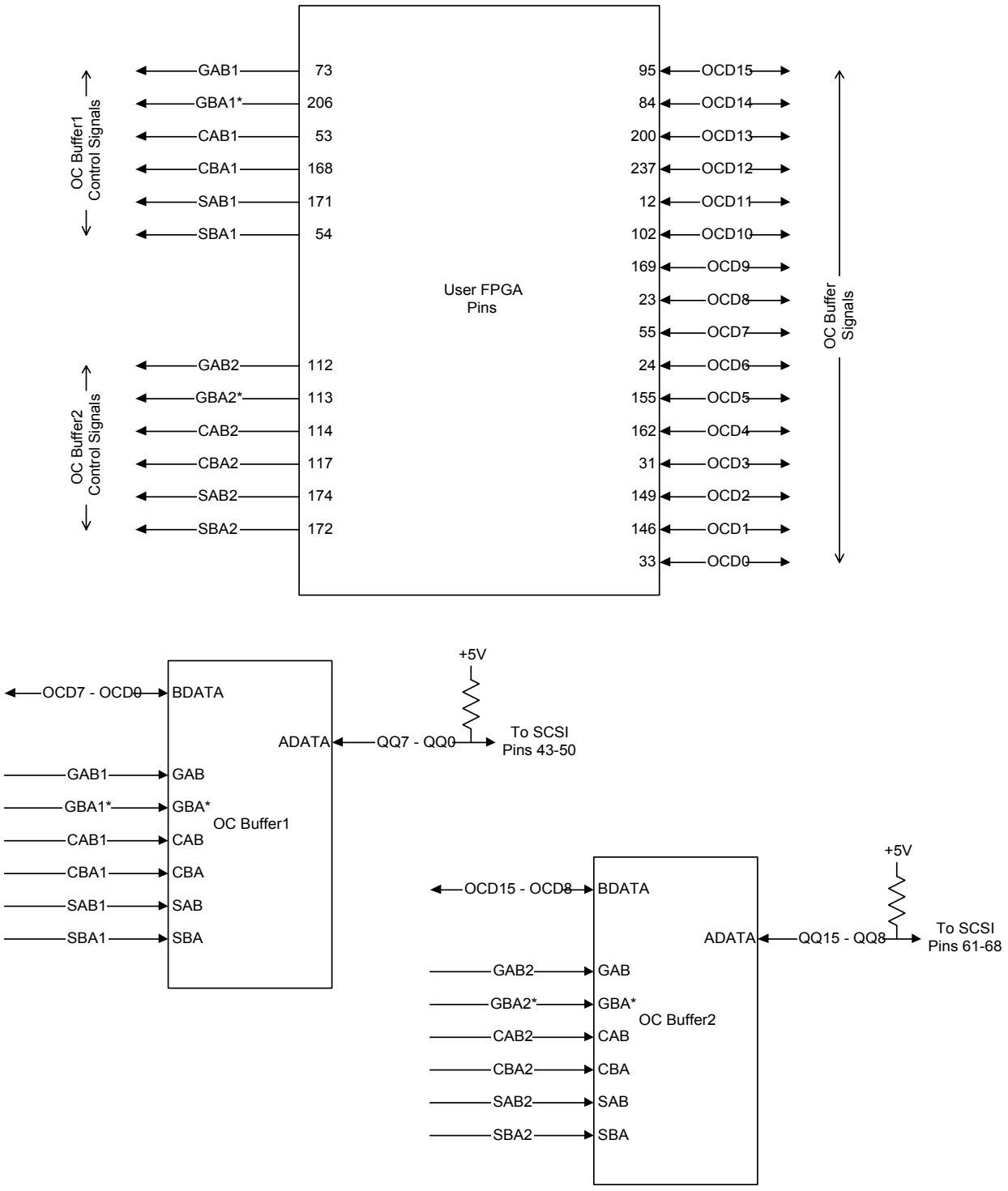


FIGURE 3-9 USER FPGA AND OPEN COLLECTOR BUFFER INTERFACE

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

The first command sets the buffer in output mode and the second command sets the buffer in input mode. The buffer has a pull-up on the output side (A-bus side). There are other modes available in the open collector buffer such as Data Store, etc. For further details, refer to data for a 74ALS653.

SIPEX TRANSMITTERS

There are two SIPEX ICs on the board with six drivers from each IC brought out to the connector. Though the signals are named TXD1, TXD2, RTS1, RTS2, DTR1, DTR2, TXC1, TXC2, ST1, ST2, RL1 and RL2, there is no particular function assigned to them, they can just be called driver1, driver 2, etc. These drivers can be configured in different modes such as RS232, RS422 and RS485. Two 8-bit latches on the board latch the configuration mode data to the SIPEX ICs. The chip select signals, SCS1* and SCS2*, are used to latch the configuration data to latch1 and latch2 respectively. These chip select signals are rising edge sensitive. The inputs to these latches come from FID0 to FID15, the data bus shared with SRAM. The mode selection is shown in the following table.

TABLE 3-8 SIPEX TRANSMITTER SIGNALS

Drivers 1- 6 @ SCS1* rising edge		Drivers 7 – 12 @ SCS2* rising edge	
Bit 3-0	**Mode**	**Bit 3-0**	**Mode**
0000	Tri-state	0000	Tri-state
0010	RS232	0010	RS232
0100	RS422	0100	RS422
0101	RS485	0101	RS485
1100	RS449	1100	RS449

To set a particular mode, write the configuration bits to the respective latch. See figures 3-6 and 3-7 for simplified schematics of the SIPEX ICs.

SIPEX RECEIVERS

There are two SIPEX ICs on the board with four receivers from each IC brought out to the connector. Though the signals are named RXD1, RXD2, CTS1, CTS2, DSR1, DSR2, RXC1 and RXC2, there is no particular function assigned to them, they can just be called receiver1, receiver2, etc. These receivers can be configured in different modes such as RS232, RS422 and RS485. Two 8-bit latches on the board latch the configuration mode data to the SIPEX ICs. The chip select signals, SCS1* and SCS2*, are used to latch the configuration data to latch1 and latch2 respectively. These chip select signals are rising edge sensitive. The inputs to these latches come from FID0 to FID15, the data bus shared with SRAM. The mode selection is shown in the following table.

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

TABLE 3-9 SIPEX RECEIVER SIGNALS

Receivers 1- 4 @ SCS1* rising edge		Receivers 5 – 8 @ SCS2* rising edge	
Bit 7-4	**Mode**	**Bit 7-4**	**Mode**
0000	Tri-state	0000	Tri-state
0010	RS232	0010	RS232
0100	RS422	0100	RS422
0101	RS485	0101	RS485
1100	RS449	1100	RS449

To set a particular mode, write the configuration bits to the respective latch. See figures 3-6 and 3-7 for simplified schematics of the SIPEX ICs.

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

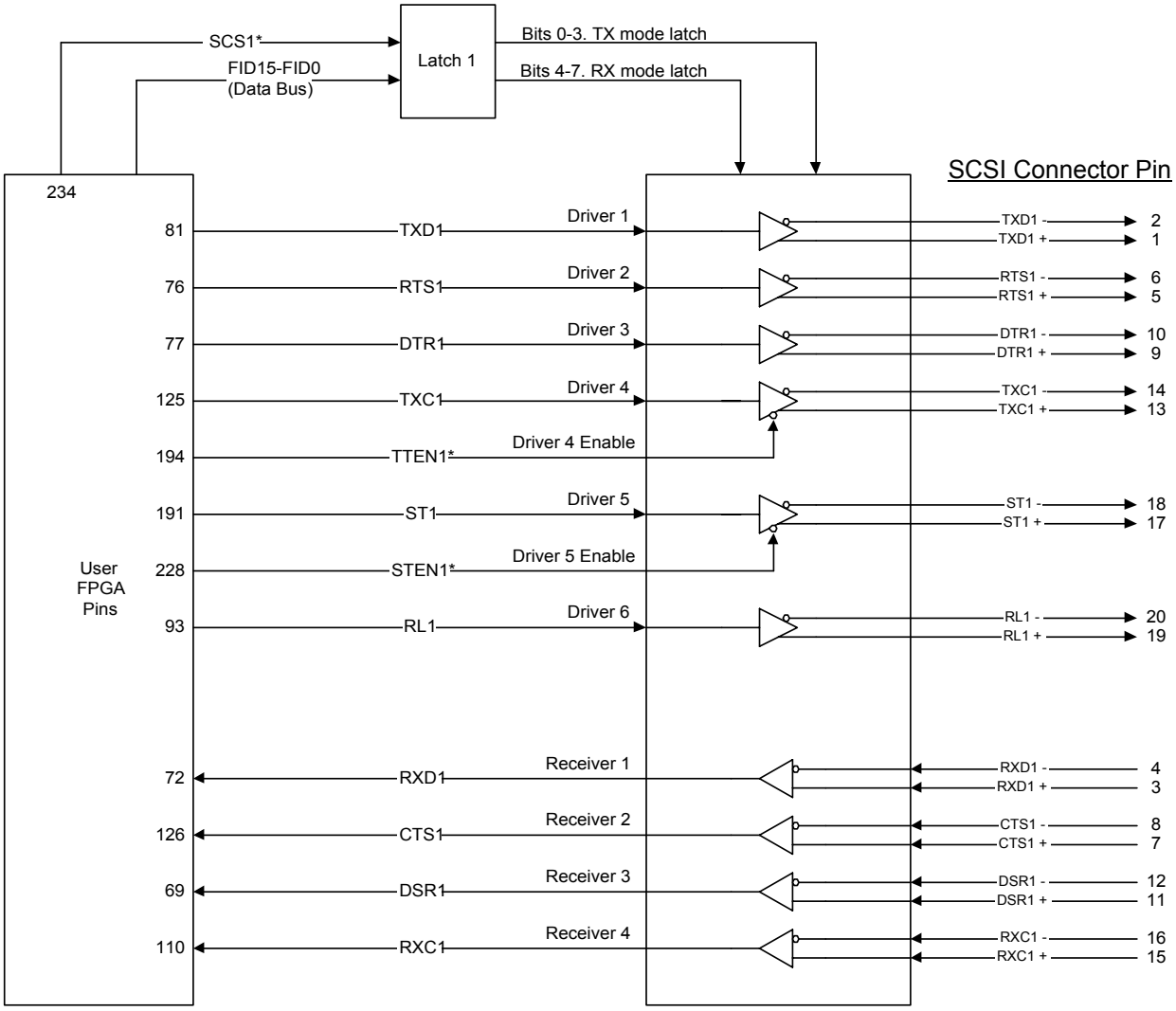


FIGURE 3-10 SIPEX1 TRANSMITTER AND RECEIVER SIGNALS

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

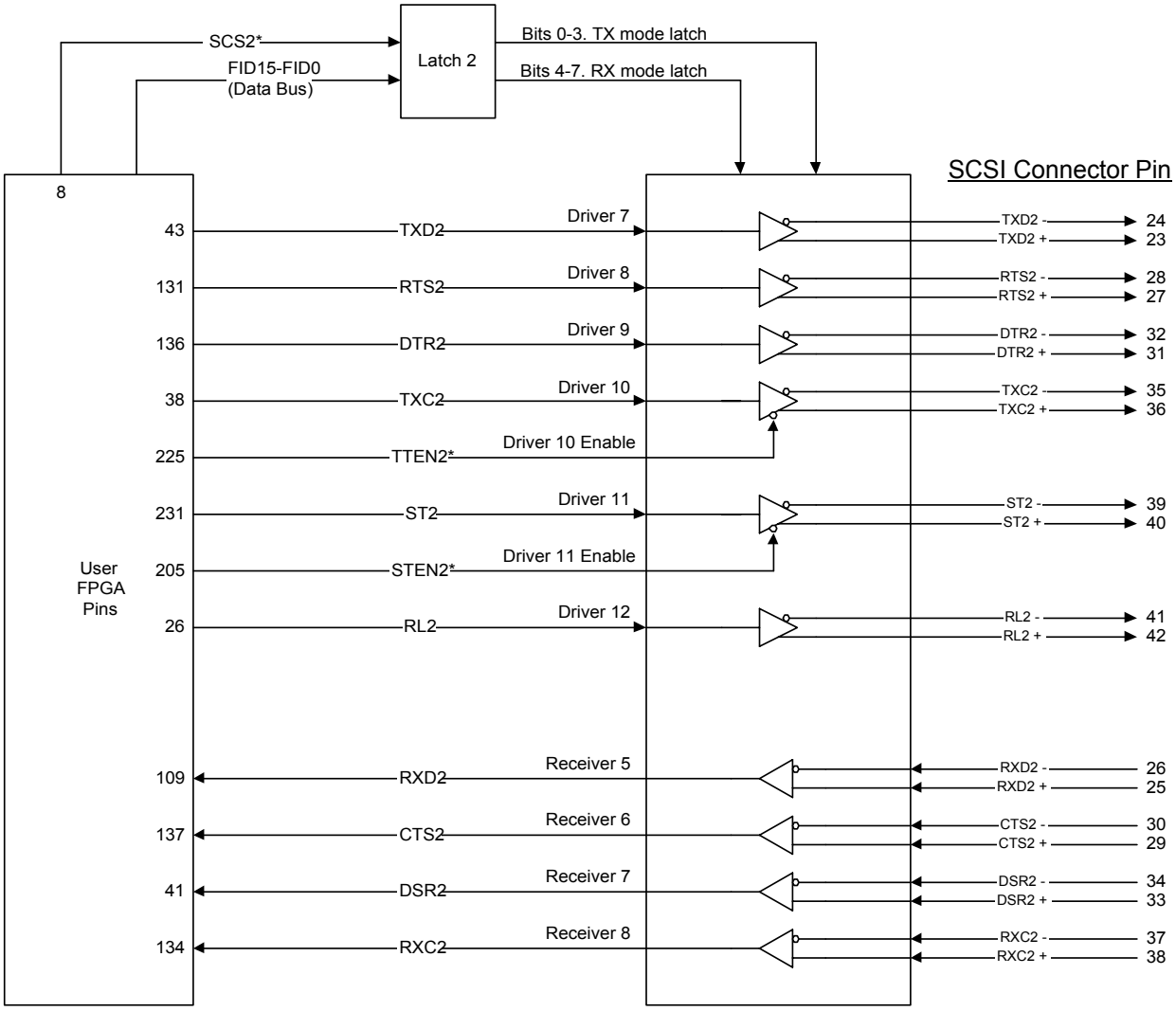


FIGURE 3-11 SIPEX2 TRANSMITTER AND RECEIVER SIGNALS

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

SIPEX TRANSMITTER AND RECEIVER TERMINATION

The transmitters and receivers are terminated as shown below. Also, note how the signals are qualified for single ended (RS232) and differential ended (RS422/RS485).

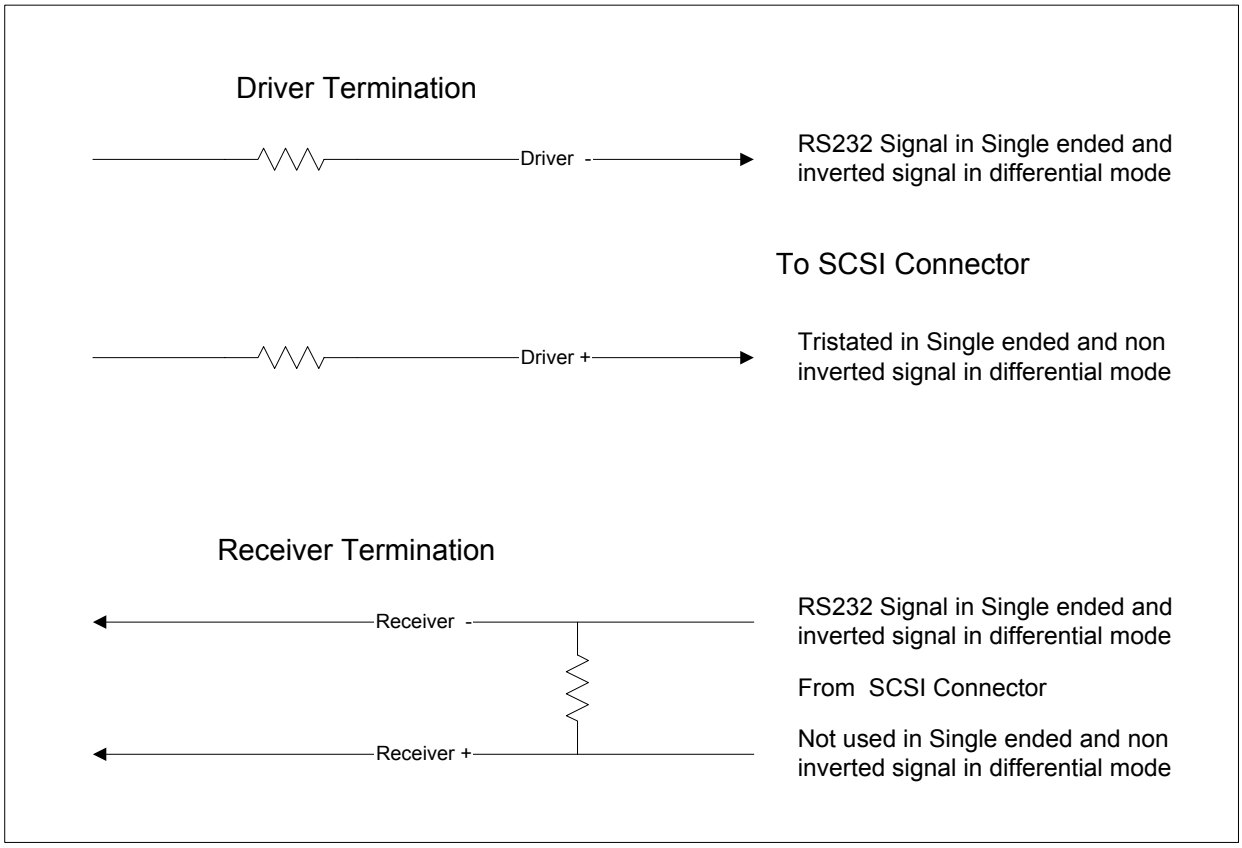


FIGURE 3-12 SIPEX TRANSMITTER AND RECEIVER TERMINATION

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

VXI BUS TTL TRIGGER INPUT

A simplified schematic of the TTL trigger input is shown in Figure 3-13. Note that there is no inversion between the VXIBus signals and the signal presented to the user FPGA. The signal to the user FPGA is called TRIGIN.

Any one of the 8 VXIBus TTL trigger lines can be used as an input to the user FPGA. Which line is used is controlled by the word serial command:

TRIGger:SOURce <trigline>

See Section 4 for a more complete description of the command.

The selected TTL trigger line is enabled/disabled with the word serial command:

TRIGger:STATe <boolean>

See Section 4 for a more complete description of the command.

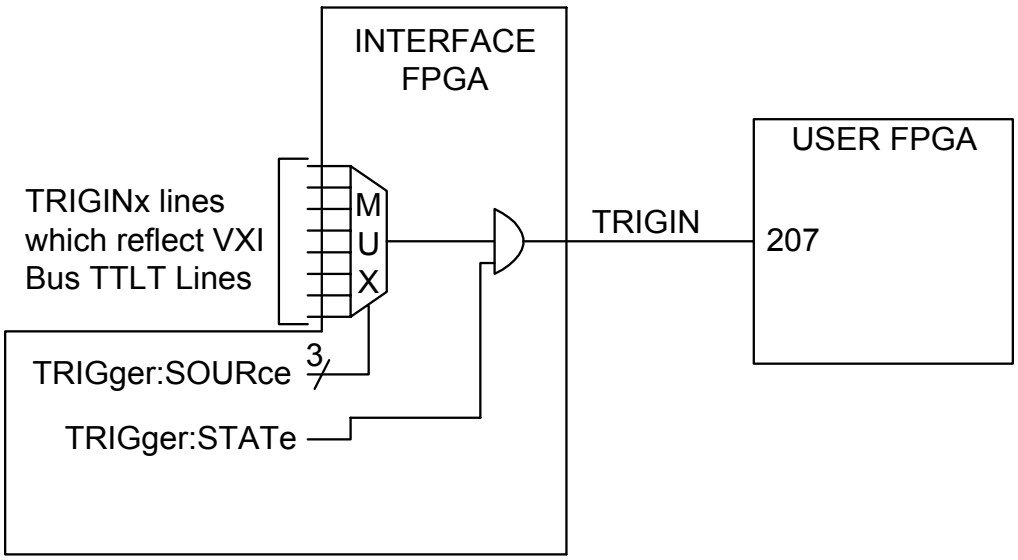


FIGURE 3-13 VXIBUS TLL TRIGGER INPUT

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

VXI BUS TTL TRIGGER OUTPUT

A simplified schematic of the TTL trigger input is shown in Figure 3-14. The default output polarity is NEG. When NEG polarity is selected, one input to the exclusive or gate is "1" and in effect the exclusive or gate inverts the TRIGOUT signal. If output state is a "1", the inverted TRIGOUT signal is passed through to the enable of the 1 of 8 decoder. The decoder will drive one of its outputs high which will turn one tristate driver on.

When one of the tristate drivers turns on it will drive its line low. The VXIBus TTL lines are meant to be driven by open collector drivers. The tristate drivers driving low or tristating perform a function equivalent to an open collector.

Any one of the 8 TTL trigger lines can be driven low by the user FPGA. Which line is driven low is controlled by the word serial command:

OUTPut:TTLTrg <trigline>

See Section 4 for a more complete description of this command.

The selected TTL trigger line is enabled/disabled with the word serial command.

OUTPut:STATe <boolean>

See Section 4 for a more complete description of this command.

The active polarity of the TRIGOUT signal is controlled by the serial command.

OUTPut:POLarity <polarity>

See Section 4 for a more complete description of this command.

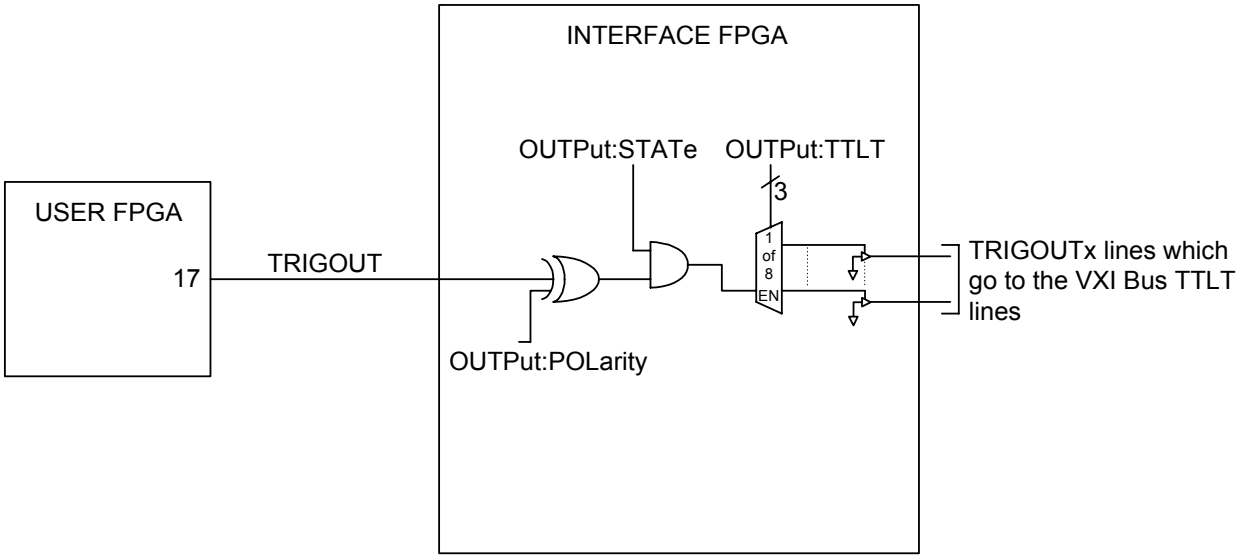


FIGURE 3-14 VXIBUS TLL TRIGGER OUTPUT

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

SECTION 4

COMMAND DICTIONARY

INTRODUCTION

The VM6069 is mainly a register driven interface; however, there are a few Word Serial commands to do some configuration. This section presents the instrument command set. It includes an alphabetical list of all the commands supported by the VM6069 divided into three sections: IEEE 488.2 commands, the instrument specific SCPI commands and the required SCPI commands. With each command is a brief description of its function, whether the command's value is affected by the *RST command, and its default value.

Each command is described, one per page, in detail. The description is presented in a regular and orthogonal way assisting the user in the use of each command. Every command entry describes the exact command and query syntax, the use and range of parameters and a complete description of the command's purpose.

PROGRAMMING

The VM6069 is a VXIbus message-based device whose command set is compliant with the Standard Command for Programmable Instruments (SCPI) programming language.

All module commands are sent over the VXIbus backplane to the module. Commands may be in upper, lower or mixed case. All numbers are sent in ASCII decimal unless otherwise noted.

The module recognizes SCPI commands. SCPI is a tree-structured language based on IEEE-STD-488.2 Specifications. It utilizes the IEEE-STD-488.2 Standard command, and the device dependent commands are structured to allow multiple branches off the same trunk to be used without repeating the trunk. To use this facility, terminate each branch with a semicolon.

See the Standard Command for Programmable Instruments (SCPI) Manual, Volume 1: Syntax & Style, Section 6, for more information.

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

The SCPI commands in this section are listed in upper and lower case. Character case is used to indicate different forms of the same command. Keywords can have both a short form and a long form (some commands only have one form). The short form uses just the keyword characters in uppercase. The long form uses the keyword characters in uppercase plus the keyword characters in lowercase. Either form is acceptable. Note that there are no intermediate forms. All characters of the short form or all characters of the long form must be used. Short forms and long forms may be freely intermixed. The actual commands sent can be in upper case, lower case or mixed case (case is only used to distinguish short and long form for the user). As an example, these commands are all correct and all have the same effect:

```
PROGram:MODule <ID>, <code>
PROGRAM:MODULE <ID>, <code>
program:module <ID>, <code>
PROG:MODule <ID>, <code>
PROG:MOD <ID>, <code>
prog:mod <ID>, <code>
```

The following command is **not** correct because it doesn't use the complete short form of **PROGram**:

```
pro:mod <ID>, <code>
```
(incorrect syntax - missing "g" - only prog or program is correct)

All of the SCPI commands also have a query form unless otherwise noted. Query forms contain a question mark (?). The query form allows the system to ask what the current setting of a parameter is. The query form of the command generally replaces the parameter with a question mark (?). Query responses do not include the command header. This means only the parameter is returned: no part of the command or "question" is returned.

NOTATION

Keywords or parameters enclosed in square brackets ([]) are optional. If the optional part is a keyword, the keyword can be included or left out. Omitting an optional parameter will cause its default to be used.

Parameters are enclosed by angle brackets (<>). Braces ({}), or curly brackets, are used to enclose one or more parameters that may be included zero or more times. A vertical bar (|), read as "or", is used to separate parameter alternatives.

ALPHABETICAL COMMAND LISTING

The following tables provide an alphabetical listing of each command supported by the VM6069 along with a brief description. If an X is found in the column titled *RST, then the value or setting controlled by this command is possibly changed by the execution of the *RST command. If no X is found, then *RST has no effect. The default column gives the value of each command's setting when the unit is powered up or when a *RST command is executed.

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

TABLE 4-1 IEEE 488.2 COMMON COMMANDS

Command	Description	*RST	Reset Value
*CLS	Clear the Status Register.	X	
*ESE	Set the Event Status Enable Register.		N/A
*ESR?	Query the Standard Event Status Register		N/A
*IDN?	Query the module identification string.		N/A
*OPC	Set the OPC bit in the Event Status Register	X	0
*RST	Reset the module to a known state		N/A
*SRE	Set the service request enable register		N/A
*STB?	Query the Status Byte Register.		N/A
*TRG	Causes a trigger event to occur.		N/A
*TST?	Starts and reports a self-test procedure.		N/A
*WAI	Halts execution and queries		N/A

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

TABLE 4-2 INSTRUMENT SPECIFIC SCPI COMMANDS

Command	Description	*RST	Reset Value
CALibration:SECure:CODE	Sets the code required to disable calibration security		N/A
CALibration:SECure[:STATe]	Enable or disable calibration security	*	1
OUTPut[:STATe]	Enables or disables the generation of a TTLTrg signal from TRIGOUT	*	0
OUTPut:TTLTrg	Sets the output trigger line for the TRIGOUT signal	*	0
OUTPut[:TTLTrg]:POLarity	Sets the polarity for the TRIGOUT signal	*	NEG
PROGram:MODule	Sets user defined manufacturer ID and model code		N/A
TRIGger:SOURce:TTLTrg	Selects the input trigger line for TRIGIN	*	0
TRIGger[:STATe]	Enables or disables the selected input trigger line	*	0

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

TABLE 4-3 REQUIRED SCPI COMMANDS

Command	Description	*RST	Reset Value
STATus:OPERation:CONDition?	Query the Operation Status Condition Register.	X	
STATus:OPERation:ENABle	Sets the Operation Status Enable Register.	X	
STATus:OPERation[:EVENt]?	Query the Operation Status Event Register.	X	
STATus:PRESet	Presets the Status Register.	X	
STATus:QUEStionable: CONDition?	Query the Questionable Status Condition Register	X	
STATus:QUEStionable:ENABle	Sets the Questionable Status Enable Register.	X	
STATus:QUEStionable[:EVENt]?	Query the Questionable Status Event Register	X	
SYSTem:ERRor?	Query the Error Queue	X	Clears queue
SYSTem:VERsion?	Query which version of the SCPI standard the module complies with.		N/A

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

COMMAND DICTIONARY

The remainder of this section is devoted to the actual command dictionary. Each command is fully described on its own page. In defining how each command is used, the following items are described:

Purpose	Describes the purpose of the command.
Type	Describes the type of command such as an event or setting.
Command Syntax	Details the exact command format.
Command Parameters	Describes the parameters sent with the command and their legal range.
Reset Value	Describes the values assumed when the *RST command is sent.
Query Syntax	Details the exact query form of the command.
Query Parameters	Describes the parameters sent with the command and their legal range. The default parameter values are assumed the same as in the command form unless described otherwise.
Query Response	Describes the format of the query response and the valid range of output.
Description	Describes in detail what the command does and refers to additional sources.
Examples	Present the proper use of each command and its query (when available).
Related Commands	Lists commands that affect the use of this command or commands that are affected by this command.

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

IEEE 488.2 COMMON COMMANDS

*CLS

Purpose	Clears all status and event registers	
Type	IEEE 488.2 Common Command	
Command Syntax	*CLS	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	N/A	
Query Parameters	N/A	
Query Response	N/A	
Description	This command clears the Status Event Register, Operation Status Register and the Questionable Data/Signal Register. It also clears the OPC flag and clears all queues (except the output queue).	
Examples	**Command / Query**	**Response *(Description)***
	`*CLS`	*(Clears all status and event registers)*
Related Commands	N/A	

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

*ESE

Purpose	Sets the bits of the Event Status Enable Register
Type	IEEE 488.2 Common Command
Command Syntax	*ESE <mask>
Command Parameters	<mask> = numeric ASCII value
***RST Value**	N/A – required parameter
Query Syntax	*ESE?
Query Parameters	N/A
Query Response	Numeric ASCII value from 0 to 255
Description	The Event Status Enable (ESE) command is used to set the bits of the Event Status Enable Register. See ANSI/IEEE 488.2-1987 section 11.5.1 for a complete description of the ESE register. A value of 1 in a bit position of the ESE register enables generation of the Event Status Bit (ESB) in the Status Byte by the corresponding bit in the Event Status Register (ESR). If the ESB is set in the Service Request Enable (SRE) register, then an interrupt will be generated. See the *ESR? command for details regarding the individual bits. The ESE register layout is: Bit 0 - Operation Complete Bit 1 - Request Control Bit 2 - Query Error Bit 3 - Device Dependent Error Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request Bit 7 - Power On The Event Status Enable query reports the current contents of the Event Status Enable Register.

Examples	**Command / Query**	**Response *(Description)***
	`*ESE 36` `*ESE?`	36 *(Returns the value of the event status enable register)*
Related Commands	*ESR?	

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

*ESR?

Purpose	Queries and clears the Standard Event Status Register	
Type	IEEE 488.2 Common Command	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*ESR?	
Query Parameters	N/A	
Query Response	Numeric ASCII value from 0 to 255	
Description	The Event Status Register (ESR) query - queries and clears the contents of the Standard Event Status Register. This register is used in conjunction with the ESE register to generate the Event Status Bit (ESB) in the Status Byte. The layout of the ESR is: Bit 0 - Operation Complete Bit 1 - Request Control Bit 2 - Query Error Bit 3 - Device Dependent Error Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request Bit 7 - Power On The Operation Complete bit is set when it receives an *OPC command. The Query Error bit is set when data is over-written in the output queue. This could occur if one query is followed by another without reading the data from the first query. The Execution Error bit is set when an execution error is detected. Errors that range from -200 to -299 are execution errors. The Command Error bit is set when a command error is detected. Errors that range from -100 to -199 are command errors. The Power On bit is set when the module is first powered on or after it receives a reset via the VXI Control Register. Once the bit is cleared (by executing the *ESR? command) it will remain cleared.	
Examples	**Command / Query**	**Response *(Description)***
	*ESR?	4
Related Commands	*ESE	

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

*IDN?

Purpose	Queries the module for its identification string	
Type	IEEE 488.2 Common Command	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*IDN?	
Query Parameters	N/A	
Query Response	ASCII character string	
Description	The Identification (IDN) query returns the identification string of the module. The response is divided into four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is an optional serial number and the fourth field is the firmware revision number. If a serial number is not supplied, the third field is set to 0 (zero).	
Examples	**Command / Query**	**Response *(Description)***
	`*IDN`	VXI Technology, Inc.,VM2164,0,1.0 *(The revision listed here is for reference only; the response will always be the current revision of the instrument.)*
Related Commands	N/A	

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

*OPC

Purpose	Sets the OPC bit in the Event Status Register	
Type	IEEE 488.2 Common Command	
Command Syntax	*OPC	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*OPC?	
Query Parameters	N/A	
Query Response	1	
Description	The Operation Complete (OPC) command sets the OPC bit in the Event Status Register when all pending operations have completed. The OPC query will return a 1 to the output queue when all pending operations have completed.	
Examples	**Command / Query**	**Response *(Description)***
	`*OPC` `*OPC?`	*(Sets the OPC bit in the Event Status Register)* 1 *(Returns the value of the Event Status Register)*
Related Commands	*WAI	

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

*RST

Purpose	Resets the module's hardware and software to a known state	
Type	IEEE 488.2 Common Command	
Command Syntax	*RST	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	N/A	
Query Parameters	N/A	
Query Response	N/A	
Description	The Reset (RST) command resets the module's hardware and software to a known state. See the command index at the beginning of this chapter for the default parameter values used with this command.	
Examples	**Command / Query**	**Response *(Description)***
	*RST	*(Resets the module)*
Related Commands	N/A	

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

*SRE

Purpose	Sets the service request enable register
Type	IEEE 488.2 Common Command
Command Syntax	*SRE <mask>
Command Parameters	<mask> = Numeric ASCII value from 0 to 255
***RST Value**	None – required parameter
Query Syntax	*SRE?
Query Parameters	N/A
Query Response	Numeric ASCII value from 0 to 255
Description	The Service Request Enable (SRE) mask is used to control which bits in the status byte generate back plane interrupts. If a bit is set in the mask that newly enables a bit set in the status byte and interrupts are enabled, the module will generate a REQUEST TRUE event via an interrupt. See the *STB? Command for the layout of bits. **Note:** Bit 6 is always internally cleared to zero as required by IEEE 488.2 section 11.3.2.3. The layout of the Service Request Enable Register is: Bit 0 – Unused Bit 1 – Unused Bit 2 – Error Queue Has Data Bit 3 – Questionable Status Summary (Not Used) Bit 4 – Message Available Bit 5 – Event Status Summary Bit 6 – 0 (per IEEE 488.2 section 11.3.2.3) Bit 7 – Operation Status Summary

Examples	**Command / Query**	**Response *(Description)***
	*SRE 4	*(Sets the service request enable register)*
	*SRE?	4 *(Returns the value of the SRE register)*
Related Commands	N/A	

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

*STB?

Purpose	Queries the Status Byte Register	
Type	IEEE 488.2 Common Command	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*STB?	
Query Parameters	N/A	
Query Response	Numeric ASCII value from 0 to 255	
Description	The Read Status Byte (STB) query fetches the current contents of the Status Byte Register. See the IEEE 488.2 specification for additional information regarding the Status byte Register and its use. The layout of the Status Register is: Bit 0 – Unused Bit 1 – Unused Bit 2 – Error Queue Has Data Bit 4 – Questionable Status Summary (not used) Bit 5 – Message Available Bit 6 – Master Summary Status Bit 7 – Operation Status Summary	
Examples	**Command / Query**	**Response *(Description)***
	*STB?	16 (*Queries the Status Byte Register)*
Related Commands	N/A	

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

*TRG

Purpose	Causes a trigger event to occur	
Type	IEEE 488.2 Common Command	
Command Syntax	*TRG	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	N/A	
Query Parameters	N/A	
Query Response	N/A	
Description	The Trigger command causes a trigger event to occur.	
Examples	**Command / Query**	**Response *(Description)***
	*TRG	*(Triggers an event)*
Related Commands	N/A	

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

*TST?

Purpose	Causes a self-test procedure to occur and queries the results	
Type	IEEE 488.2 Common Command	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*TST?	
Query Parameters	N/A	
Query Response	Numeric ASCII value from 0 to 143	
Description	Initiates the counter self-test operation. If the test fails, an error message is placed in the error queue and then the error LED blinks. The self test tests the following: • Two 4 kb counter measurement buffers • Logic registers • Analog front end per-amp offset, pre-amp inverter and pre-amp gain digital to analog converters (DACs) • A 2.5 MHz signal is routed through a test source and checked for accuracy	
Examples	**Command / Query**	**Response *(Description)***
	*TST	0 (*Begins the self-test procedure returns the result)*
Related Commands	N/A	

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

*WAI

Purpose	Halts execution of additional commands and queries until the No Operation Pending message is true	
Type	IEEE 488.2 Common Command	
Command Syntax	*WAI	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	N/A	
Query Parameters	N/A	
Query Response	N/A	
Description	The Wait to Continue command halts the execution of commands and queries until the No Operation Pending message is true. This command makes sure that all previous commands have been executed before proceeding. It provides a way of synchronizing the module with its commander.	
Examples	**Command / Query**	**Response *(Description)***
	`*WAI`	*(Pauses the execution of additional commands until the No Operation Pending message is true.)*
Related Commands	*OPC	

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

INSTRUMENT SPECIFIC SCPI COMMANDS

CALibration:SECure:CODE

Purpose	Sets the code required to disable calibration security	
Type	Setting	
Command Syntax	CALibration:CODE #	
Command Parameters	# = the code string can be from 1 to 12 ASCII characters in length entered in IEEE-488.2 definite or indefinite length arbitrary block format	
***RST Value**	N/A	
Query Syntax	CALibration:CODE?	
Query Parameters	N/A	
Query Response	#	
Description	Calibration security must first be disabled before the code can be changed. Before shipping the instrument, the factory sets the code to VM6069. **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**	
Examples	**Command / Query**	**Response *(Description)***
	`CAL:SEC:CODE #16VM6069`	*(Sets the security code to the factory setting of VM6069)*
Related Commands	CALibration:SECure[:STATe]	

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

CALibration:SECure[:STATe]

Purpose	Enable or disable calibration security
Type	Event
Command Syntax	CALibration:SECure[:STATe] <mode>, #
Command Parameters	<mode> = boolean - 0 | 1 | OFF | ON # = The code must be present to disable the security or it will generate an error.
***RST Value**	1
Query Syntax	CALibration:SECure[:STATe]?
Query Parameters	N/A
Query Response	0 | 1
Description	The module is powers up with the secure state enabled. While security is on, no stores to non-volatile memory are allowed. This command turns the state on or off. In order to disable the security state, the current security code must be supplied. To turn on security, code does not need to be supplied. If it is supplied the code is checked. The security code must be supplied in IEEE-488.2 definite or indefinite length arbitrary block format. **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:SEC OFF`	*(Disables security mode)*
	`CAL:SEC:STAT 1`	*(Turns calibration security back on again)*
	`CAL:SEC:SEC?`	1 *(Indicates the calibration security is enables so that no new information can be stored in non-volatile memory)*

Related Commands	CALibration:SECure:CODE PROGram:MODule

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

OUTPut[:STATe]

Purpose	Enables or disables the generation of a TTLTrg signal from TRIGOUT	
Type	Setting	
Command Syntax	OUTPut[:STATe] <boolean>	
Command Parameters	<boolean> = 0 | 1 | OFF | ON	
Reset Value	0	
Query Syntax	OUTPut[:STATe]?	
Query Parameters	N/A	
Query Response	0 | 1	
Description	The Output State command enables or disables the output signal from TRIGOUT. The default state is disabled.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP 1`	*(Enables the output signal)*
	`OUTP?`	*1 (Queries and returns that the output signal is enabled)*
Related Commands	OUTPut:TTLTrg OUTPut[:TTLTrg]:POLarity	

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

OUTPut:TTLTrg

Purpose	Sets the output trigger line for the TRIGOUT signal	
Type	Setting	
Command Syntax	OUTPut:TTLTrg <trigline>	
Command Parameters	<trigline> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7	
Reset Value	0	
Query Syntax	OUTPut:TTLTrg?	
Query Parameters	N/A	
Query Response	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7	
Description	The Output TTL Trigger command sets the output trigger line used for the TRIGOUT output signal. The default setting is trigger line 0. Note that the Output State must be enabled for TRIGOUT to have any effect.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:TTLT 3` `OUTP:TTLT?`	*(Sets the trigger line to #3)* 3 *(Verifies that the output trigger line is #3)*
Related Commands	OUTPut[:STATe] OUTPut[:TTLTrg]:POLarity	

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

OUTPut[:TTLTrg]:POLarity

Purpose	Sets the polarity for the TRIGOUT signal	
Type	Setting	
Command Syntax	OUTPut[:TTLTrg]:POLarity <polarity>	
Command Parameters	<polarity> = NEG | POS	
Reset Value	NEG	
Query Syntax	OUTPut[:TTLTrg]:POLarity?	
Query Parameters	N/A	
Query Response	NEG | POS	
Description	The Output TTL Trig Polarity command sets the TRIGOUT signal to show as a falling or rising edge of a pulse. The default setting is on the falling edge (NEG).	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:POL POS`	*(Sets the operation complete output signal to happen on a POSitive edge)*
	`OUTP:POL?`	*(Verifies that the signal will be on the POSitive edge)*
Related Commands	OUTPut[:STATe] OUTPut:TTLTrg	

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

PROGram:MODule

Purpose	Sets user defined manufacturer ID and model code	
Type	Setting	
Command Syntax	PROGram:MODule <ID>, <code>	
Command Parameters	<ID> = user defined manufacturer's ID <code> = user defined model code	
Reset Value	N/A	
Query Syntax	PROGram:MODule?	
Query Parameters	N/A	
Query Response	<ID>, <code>	
Description	The Program Module command set a user-defined manufacturer's I.D. and model code. The Calibration Secure State must be disabled before information can be changed with the Program Module command.	
Examples	**Command / Query**	**Response *(Description)***
	`PROG:MOD 3915,278` `PROG:MOD?`	*(Sets the manufacturer's ID and model code)* 3915,278 *(3915 is VXI Technology's manufacturing code. 278 is the model code VXI Technology has assigned to the VM6069)*
Related Commands	CALibration:SECure[:STATe]	

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

TRIGger:SOURce:TTLTrg

Purpose	Selects the input trigger line for TRIGIN	
Type	Setting	
Command Syntax	TRIGger:SOURce:TTLTrg <trigline>	
Command Parameters	<trigline> = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7	
Reset Value	0	
Query Syntax	TRIGger:SOURce:TTLTrg?	
Query Parameters	N/A	
Query Response	0 | 1 | 2 | 3 | 4 | 5 | 6 | 7	
Description	The Trigger Source TTLTrg command selects the trigger line used for the trigger-input signal (TRIGIN). The input trigger function must also be enabled. See TRIGger[:STATe].	
Examples	**Command / Query**	**Response *(Description)***
	`TRIG:SOUR:TTLT 4` `TRIG:SOUR?`	*(Sets the trigger input source to trigger line #4)* *(Verifies that the trigger signal source is set to trigger line #4)*
Related Commands	TRIGger[:STATe]	

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

TRIGger[:STATe]

Purpose	Enables or disables the selected input trigger line	
Type	Setting	
Command Syntax	TRIGger[:STATe] <boolean>	
Command Parameters	<boolean> = 0 | 1 | OFF | ON	
Reset Value	0	
Query Syntax	TRIGger[:STATe]?	
Query Parameters	N/A	
Query Response	0 | 1	
Description	The Trigger State command enables or disables the selected input trigger line for the TRIGIN signal.	
Examples	**Command / Query**	**Response *(Description)***
	`TRIG 1`	*(Enables the selected input trigger line)*
	`TRIG?`	*(Verifies that the selected input trigger line is enabled)*
Related Commands	TRIGger:SOURce:TTLTrg	

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

REQUIRED SCPI COMMANDS

STATus:OPERation:CONDition?

Purpose	The STATus:OPERation:CONDition query returns the current operational status of the counter
Type	Required SCPI query
Command Syntax	N/A
Command Parameters	N/A
***RST Value**	0
Query Syntax	STATus:OPERation:CONDition?
Query Parameters	N/A
Query Response	This query returns the operational condition register value.
Description	The STATus:OPERation:CONDition query returns the current operational status of the counter. The bit definitions of the value are (bit ( ) = the least significant bit):

Bit	Definition	Function
0	Calibrating	Set when any CALibration operation is running. Cleared when the CALibration operation is complete.
1	Setting	Set when the instrument changes its function or range. Cleared when the all circuitry has settled.
2	Ranging	Set when the instrument is auto-ranging. Cleared when the input range has been found.
3	Sweeping	Not used.
4	Measuring	Set when an INITiate command is executed. Cleared when the command is complete or aborted
5	Triggering	Not used.
6	Arming	Set when the instrument is waiting for an arm signal. Cleared when the arm is received.
7	Correcting	Set when the instrument is performing an auto-zero operation. Cleared when the auto-zero operation is complete.
8	Testing (User 1)	Set when the instrument is performing a self-test. Cleared when the self-test is complete.
9	Testing (User 2)	Set when the instrument is in the process of aborting an operation. Cleared when the abort is complete.
10	User 3	Not used
11	User 4	Not used
12	User 5	Reserved
13	Instrument Summary	Not used
14	Program Running	Not used
15	Reserved	Always 0

Example	**Command / Query**	**Response *(Description)***
	`STAT:OPER:COND?`	16 *(Makes a measurement (0010 hex)*

Related Commands	MEASure? READ? INITiate ABORt

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

STATus:OPERation:ENABle

Purpose	Sets the Operation Status Register's enable register	
Type	Required SCPI command	
Command Syntax	STATus:OPERation:ENABle <NRf>	
Command Parameters	<NRf> = numeric ASCII value from 0 to 32767	
***RST Value**	<NRf> must be specified	
Query Syntax	STATus:OPERation:ENABle?	
Query Parameters	N/A	
Query Response	<NRf> = Numeric ASCII value from 0 to 32767	
Description	This command enables bits in the Operation Status Register's enable register to report to the summary bit; sets Status Bytes register bit 7 to true. The query reports the bits enabled in the Operation Status Register's enable register, then clears the register contents and enters the value into the computer.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER ENAB 33` `STAT:OPER:ENAB?`	*(Enables bit 0 and bit 5)* 33 *(Indicates that bit 0 and 5 are enabled)*
Related Commands	STATus:OPERation:CONDition? STATus:OPERation[:EVENt]	

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

STATus:OPERation:NTR

Purpose	Sets the negative transition filter	
Type	Required SCPI command	
Command Syntax	STATus:OPERation:NTR	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation:NTR?	
Query Parameters	N/A	
Query Response	N/A	
Description	Sets the negative transition filter. Setting a bit in the negative transition filter shall cause a 1 to 0 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register. Note that 32767 is the maximum value returned as the most-significant bit of the register cannot be set true.	
Examples	**Command / Query**	**Response *(Description)***
	STAT:PRES	
Related Commands	N/A	

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

STATus:OPERation:PTR

Purpose	Sets the positive transition filter	
Type	Required SCPI command	
Command Syntax	STATus:OPERation:PTR	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation:PTR?	
Query Parameters	N/A	
Query Response	N/A	
Description	Sets the positive transition filter. Setting a bit in the positive transition filter shall cause a 0 to1 transition in the corresponding bit of the associated condition register to cause a 1 to be written in the associated bit of the corresponding event register. Note that 32767 is the maximum value returned as the most-significant bit of the register cannot be set true.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER:PTR`	
Related Commands	N/A	

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

STATus:OPERation[:EVENt]?

Purpose	Queries the Operation Status Register's event register	
Type	Required SCPI query	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation[:EVENt]?	
Query Parameters	N/A	
Query Response	0	
Description	Queries the bits set in the event register of the Operation Status Register. This command clears all bits in the event register.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER?`	0
Related Commands	STATus:OPERation:CONDition? STATus:OPERation:ENABle?	

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

STATus:PRESet

Purpose	Presets the Status Registers	
Type	Required SCPI command	
Command Syntax	STATus:PRESet	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	N/A	
Query Parameters	N/A	
Query Response	N/A	
Description	The Status Preset command presets the Status Registers. The Operational Status Enable Register is set to 0 and the Questionable Status Enable Register is set to 0. This command is provided for SCPI compliance only.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:PRES`	
Related Commands	N/A	

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

STATus:QUEStionable:CONDition?

Purpose	Queries the Questionable Status Condition Register	
Type	Required SCPI query	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:QUEStionable:CONDition?	
Query Parameters	N/A	
Query Response	0	
Description	The Questionable Status Condition Register query is provided for SCPI compliance only. The VM2164 does not alter any bits in this register and a query always reports a 0.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES:COND?`	0
Related Commands	N/A	

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

STATus:QUEStionable:ENABle

Purpose	Sets the Questionable Status Enable Register	
Type	Required SCPI command	
Command Syntax	STATus:QUEStionable:ENABle <NRf>	
Command Parameters	<NRf> = numeric ASCII value from 0 to 32767	
***RST Value**	<NRf> must be supplied	
Query Syntax	STATus:QUEStionable:ENABle?	
Query Parameters	N/A	
Query Response	<NRf> = Numeric ASCII value from 0 to 32767	
Description	The command sets the bits in the Questionable Data/Signal Register's enable register to be reported to the summary bit (sets Status Byte Register bit 3 to true). The Status Questionable Enable query reports the contents of the Questionable Data/Signal Register's enable register, then clears the register contents and enters the value into the computer.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES:ENAB 64` `STAT:QUES:ENAB?`	64
Related Commands	N/A	

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

STATus:QUEStionable[:EVENt]?

Purpose	Queries the Questionable Status Event Register	
Type	Required SCPI query	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:QUEStionable[:EVENt]?	
Query Parameters	N/A	
Query Response	Decimal number	
Description	The query reports the bits set in the event register of the Questionable Data/Signal register. This command reads the event register, then clears all bits in the event register and enters the value into the computer.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES?`	0
Related Commands	N/A	

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

APPENDIX A - APPLICATION EXAMPLE

SAMPLE CODE

The sample code supplied with the board gives an example of how the various peripherals are accessed. These peripherals may be used in different ways by changing the design code. The sample uses the following register map. This sample code is available in files **uf_std.ucf** and **uf_std.v**.

VXI Register	Function
22	RAM1 Higher address (D2-D0 =A18-A16)
24	RAM1 Lower address (D15-D0 =A15-A0)
26	RAM1 Data
28	RAM2 Higher address (D2-D0 =A18-A16)
2A	RAM2 Lower address (D15-D0 =A15-A0)
2C	RAM2 Data
2E	SIPEX Latch (D15-D0) SIPEX 1 and 2
30	SIPEX Transmitters (D11-D0= Driver12-Driver1) (D15-D12 = Driver11 enable, Driver10 enable, Driver5 enable and Driver4 enable, 0 = enable, 1 = disable)
32	SIPEX Receivers (D7-D0=Receiver8-Receiver1)
34	Open Collector Buffer Configure (D0- Buffer1, D1-Buffer2) 0=Output, 1=Input
36	Open Collector Buffer Data (D15-D0= OCD15-OCD0)
38	Bit 0 = PIRQ Bit 1 = TRIGOUT Bit 2 = TRIGIN
3A	10 MHz Clock Counter
3C	40 MHz Clock Counter
3E	Not used in this example

MEMORY ACCESS

Two sets of memory each (1MB) is read from and written to the memory through VXI registers assigned above.

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

SIPEX LATCH

SIPEX Latch is configured as given below.

Bit	Function	Mode
0 to 3	**SIPEX-1, Driver:** "0000" Tri-state. "0010" RS-232. "0100" RS-422. "0101" RS-485. "1100" RS-449. "1101" EIA-530. "1110" V.35.	Read / Write
4 to 7	**SIPEX-1, Receiver:** "0000" Undefined. "0010" RS-232. "0100" RS-422. "0101" RS-485. "1100" RS-449. "1101" EIA-530. "1110" V.35.	Read / Write
8 to 11	**SIPEX-2, Driver:** "0000" Tri-state. "0010" RS-232. "0100" RS-422. "0101" RS-485. "1100" RS-449. "1101" EIA-530. "1110" V.35.	Read / Write
12 to 15	**SIPEX-2, Receiver:** "0000" Undefined. "0010" RS-232. "0100" RS-422. "0101" RS-485. "1100" RS-449. "1101" EIA-530. "1110" V.35.	Read / Write

The register illustrates:

- writing to the Sipex transmitters
- enabling or disabling the four tri-state drivers
- reading the Sipex receivers

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

OPEN COLLECTOR BUFFERS

Each buffer is configured either in input mode or output mode. The data is written or read from the VXI registers.

MISC. REGISTERS

PIRQ	This bit is assigned to the PIRQ pin. An interrupt can be generated by toggling this bit.
TRIGOUT	This bit is assigned to the TRIGOUT signal. The TRIGOUT line (7-0) is selected by word serial command.
TRIGIN	This is a read-only bit that indicates the status of the TRIGIN signal. The TRIGIN line (7-0) is selected by word serial command.
10 and 40 MHz Counters	The 10 MHz counter simply counts the 10 MHz clock. Though the counter is 32 bit, only the upper 16 bits are assigned to this register. The 40 MHz counter operates the same as the 10 MHz counter.

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

UF_STD.UCF

```
##############################################
#      BASIC UCF SYNTAX EXAMPLES V2.1.6      #
##############################################
#
# The "#" symbol is a comment character.   To use this sample file, find the
# specification necessary, remove the comment character (#) from the
beginning
# of the line, and modify the line (if necessary) to fit your design.
#
#              TIMING SPECIFICATIONS
#
# Timing specifications can be applied to the entire device (global) or to
# specific groups in your design (called "time groups'). The time groups are
# declared in two basic ways.
#
# Method 1: Based on a net name, where 'my_net' is a net that touches all the
#           logic to be grouped in to 'logic_grp'. Example:
#NET my_net TNM_NET = logic_grp ;
#
# Method 2: Group using the key word 'TIMEGRP' and declare using the names of
#           logic in your design. Example:
#TIMEGRP group_name = FFS ("U1/*");
#           creates a group called 'group_name' for all flip-flops within
#           the hierarchical block called U1. Wildcards are valid.
#
# Grouping is very important because it lets you tell the software which
parts
# of a design run at which speeds.  For the majority of the designs with only
# one clock, use simple global constraints.
#
# The type of grouping constraint you use can vary depending on the synthesis
# tools you are using.  Foundation Express does better with Method 2.
#
#
############################################################
# Internal to the device clock speed specifications - Tsys #
############################################################
#
# data      __________       /^^^^^\       __________   out
# ----------| D   Q |-----{ LOGIC } -----| D   Q |------
#           |       |      \vvvvv/       |       |
#        ---|> CLK  |                  ---|> CLK  |
# clock  |  ---------                  |  ---------
# ------------------------------------
#
# ---------------
# Single Clock
# ---------------
#
```


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

```
# ----------------
# PERIOD TIME-SPEC
# ----------------
# The PERIOD spec. covers all timing paths that start or end at a
# register, latch, or synchronous RAM which are clocked by the reference
# net (excluding pad destinations).  Also covered is the setup
# requirement of the synchronous element relative to other elements
# (ex. flip flops, pads, etc...).
# NOTE:  The default unit for time is nanoseconds.
#
#NET clock PERIOD = 50ns ;
#
#       -OR-
#
# ------------------
# FROM:TO TIME-SPECs
# ------------------
# FROM:TO style timespecs can be used to constrain paths between time
# groups.  NOTE:  Keywords:  RAMS, FFS, PADS, and LATCHES are predefined
# time groups used to specify all elements of each type in a design.
#TIMEGRP RFFS = RISING FFS ("*");  // creates a rising group called RFFS
#TIMEGRP FFFS = FALLING FFS ("*");  // creates a falling group called FFFS
#TIMESPEC TSF2F  = FROM : FFS   : TO : FFS   : 50 ns; // Flip-flips with the
same edge
#TIMESPEC TSR2F  = FROM : RFFS  : TO : FFFS  : 25 ns; // rising edge to
falling edge
#TIMESPEC TSF2R  = FROM : FFFS  : TO : RFFS  : 25 ns; // falling edge to
rising edge
#
# ---------------
# Multiple Clocks
# ---------------
# Requires a combination of the 'Period' and 'FROM:TO' type time
specifications
#NET clock1 TNM_NET = clk1_grp ;
#NET clock2 TNM_NET = clk2_grp ;
#
#TIMESPEC TS_clk1 = PERIOD : clk1_grp : 50 ;
#TIMESPEC TS_clk2 = PERIOD : clk2_grp : 30 ;
#TIMESPEC TS_ck1_2_ck2 = FROM : clk1_grp : TO : clk2_grp : 50 ;
#TIMESPEC TS_ck2_2_ck1 = FROM : clk2_grp : TO : clk1_grp : 30 ;
#
#
##############################################################
# CLOCK TO OUT specifications - Tco                          #
##############################################################
#
# from      __________       /^^^^^\       --------\
# ----------| D   Q |-----{ LOGIC } -----| Pad    >
# PLD       |       |      \vvvvv/       --------/
#        ---|> CLK  |
# clock  |  ---------
# --------
#
```


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

```
# ----------------
# OFFSET TIME-SPEC
# ----------------
# To automatically include clock buffer/routing delay in your
# clock-to-out timing specifications, use OFFSET constraints .
# For an output where the maximum clock-to-out (Tco) is 25 ns:
#
#NET out_net_name OFFSET = OUT 25 AFTER clock_net_name ;
#
#      -OR-
#
# ------------------
# FROM:TO TIME-SPECs
# ------------------
#TIMESPEC TSF2P  = FROM : FFS   : TO : PADS  : 25 ns;
# Note that FROM: FFS : TO: PADS constraints start the delay analysis
# at the flip flop itself, and not the clock input pin.  The recommended
# method to create a clock-to-out constraint is to use an OFFSET constraint.
#
#
############################################################
# Pad to Flip-Flop speed specifications - Tsu              #
############################################################
#
# ------\         /^^^^^\         __________   into PLD
# |pad   >-------{ LOGIC } -----| D   Q |------
# ------/         \vvvvv/       |       |
#                            ---|> CLK  |
# clock                      |  ---------
# ---------------------------
#
# ----------------
# OFFSET TIME-SPEC
# ----------------
# To automatically account for clock delay in your input setup timing
# specifications, use OFFSET constraints.
# For an input where the maximum setup time is 25 ns:
#NET in_net_name OFFSET = IN 25 BEFORE clock_net_name ;
#
#      -OR-
#
# ------------------
# FROM:TO TIME-SPECs
# ------------------
#TIMESPEC TSP2F  = FROM : PADS  : TO : FFS   : 25 ns;
# Note that FROM: PADS : TO: FFS constraints do not take into account any
# delay for the clock path.  The recommended method to create an input
# setup time constraint is to use an OFFSET constraint.
#
#
```


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

```
############################################################
# Pad to Pad speed specifications - Tpd                    #
############################################################
#
# ------\         /^^^^^\       -------\
# |pad   >-------{ LOGIC } -----| pad   >
# ------/         \vvvvv/       -------/
#
# ------------------
# FROM:TO TIME-SPECs
# ------------------
#TIMESPEC TSP2P  = FROM : PADS  : TO : PADS  : 125 ns;
#
#
############################################################
# Other timing specifications                              #
############################################################
#
# -------------
# TIMING IGNORE
# -------------
# If you can ignore timing of paths, use Timing Ignore (TIG). NOTE: The
# "*" character is a wild card, which can be used for bus names.  A "?"
# character can be used to wild-card one character.
# Ignore timing of net reset_n:
#NET : reset_n : TIG ;
#
# Ignore data_reg(7:0) net in instance mux_mem:
#NET : mux_mem/data_reg* : TIG ;
#
# Ignore data_reg(7:0) net in instance mux_mem as related to a TIMESPEC
# named TS01 only:
#NET : mux_mem/data_reg* : TIG = TS01 ;
#
# Ignore data1_sig and data2_sig nets:
#NET : data?_sig : TIG ;
#
# ---------------
# PATH EXCEPTIONS
# ---------------
# If your design has outputs that can be slower than others, you can
# create specific timespecs similar to this example for output nets
# named out_data(7:0) and irq_n:
#TIMEGRP slow_outs = PADS(out_data* : irq_n) ;
#TIMEGRP fast_outs = PADS : EXCEPT : slow_outs ;
#TIMESPEC TS08 = FROM : FFS : TO : fast_outs : 22 ;
#TIMESPEC TS09 = FROM : FFS : TO : slow_outs : 75 ;
#
# If you have multi-cycle FF to FF paths, you can create a time group
# using either the TIMEGRP or TNM statements.
#
```


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

```
# WARNING:  Many VHDL/Verilog synthesizers do not predictably name flip
# flop Q output nets.  Most synthesizers do assign predictable instance
# names to flip flops, however.
#
# TIMEGRP example:
#TIMEGRP slowffs = FFS(inst_path/ff_q_output_net1* :
#inst_path/ff_q_output_net2*);
#
# TNM attached to instance example:
#INST inst_path/ff_instance_name1_reg* TNM = slowffs ;
#INST inst_path/ff_instance_name2_reg* TNM = slowffs ;
#
# If a FF clock-enable is used on all flip flops of a multi-cycle path,
# you can attach TNM to the clock enable net.  NOTE:  TNM attached to a
# net "forward traces" to any FF, LATCH, RAM, or PAD attached to the
# net.
#NET ff_clock_enable_net TNM = slowffs ;
#
# Example of using "slowffs" timegroup, in a FROM:TO timespec, with
# either of the three timegroup methods shown above:
#TIMESPEC TS10 = FROM : slowffs : TO : FFS : 100 ;
#
# Constrain the skew or delay associate with a net.
#NET any_net_name MAXSKEW = 7 ;
#NET any_net_name MAXDELAY = 20 ns;
#
#
# Constraint priority in your .ucf file is as follows:
#
#    highest 1.  Timing Ignore (TIG)
#            2.  FROM : THRU : TO specs
#            3.  FROM : TO specs
#    lowest  4.  PERIOD specs
#
# See the on-line "Library Reference Guide" document for
# additional timespec features and more information.
#
#
############################################################
#
#
#         LOCATION and ATTRIBUTE SPECIFICATIONS            #
#
#
############################################################
# Pin and CLB location locking constraints                 #
############################################################
#
```


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

```
# ----------------------
# Assign an IO pin number
# ----------------------
#INST io_buf_instance_name  LOC = P110 ;
#NET io_net_name  LOC = P111 ;
#
# ----------------------
# Assign a signal to a range of I/O pins
# ----------------------
#NET "signal_name" LOC=P32, P33, P34;
#
# ----------------------
# Place a logic element(called a BEL) in a specific CLB location.
# BEL = FF, LUT, RAM, etc...
# ----------------------
#INST instance_path/BEL_inst_name  LOC = CLB_R17C36 ;
#
# ----------------------
# Place CLB in rectangular area from CLB R1C1 to CLB R5C7
# ----------------------
#INST /U1/U2/reg<0> LOC=clb_r1c1:clb_r5c7;
#
# ----------------------
# Place hierarchical logic block in rectangular area from CLB R1C1 to CLB
R5C7
# ----------------------
#INST /U1* LOC=clb_r1c1:clb_r5c7;
#
# ----------------------
# Prohibit IO pin P26 or CLBR5C3 from being used:
# ----------------------
#CONFIG PROHIBIT = P26 ;
#CONFIG PROHIBIT = CLB_R5C3 ;
# Config Prohibit is very important for forcing the software to not use
critical
# configuration pins like INIT or DOUT on the FPGA.  The Mode pins and JTAG
# Pins require a special pad so they will not be available to this constraint
#
# ----------------------
# Assign an OBUF to be FAST or SLOW:
# ----------------------
#INST obuf_instance_name FAST ;
#INST obuf_instance_name SLOW ;
#
# ----------------------
# FPGAs only:  IOB input Flip-flop delay specification
# ----------------------
# Declare an IOB input FF delay (default = MAXDELAY).
# NOTE:  MEDDELAY/NODELAY can be attached to a CLB FF that is pushed
# into an IOB by the "map -pr i" option.
#INST input_ff_instance_name MEDDELAY ;
#INST input_ff_instance_name NODELAY ;
#
```


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

```
# ----------------------
# Assign Global Clock Buffers Lower Left Right Side
# ----------------------
# INST gbuf1 LOC=SSW
#
# #

NET ERCLK               LOC = P63;
NET EPFPGACLRN          LOC = P25;
NET ERD                 LOC = P208;
NET EA29                LOC = P11;

NET EA<0>               LOC = P170;
NET EA<1>               LOC = P115;
NET EA<2>               LOC = P111;
NET EA<3>               LOC = P198;
NET EA<4>               LOC = P104;
NET EA<5>               LOC = P218;

NET EBSELN              LOC = P65;
NET EASN                LOC = P9;
NET EDSN                LOC = P67;

#NET ESIZ0              LOC = P176;
#NET ESIZ1              LOC = P224;
NET ECSPACEN            LOC = P4;

NET EDSACK1N            LOC = P230;
NET ETRIGIN             LOC = P207;
NET ETRIGOUT            LOC = P17;

# Flash/interface data bus
NET ED<0>               LOC = P177;
NET ED<1>               LOC = P173;
NET ED<2>               LOC = P159;
NET ED<3>               LOC = P152;
NET ED<4>               LOC = P148;
NET ED<5>               LOC = P141;
NET ED<6>               LOC = P129;
NET ED<7>               LOC = P123;
NET ED<8>               LOC = P49;
NET ED<9>               LOC = P46;
NET ED<10>              LOC = P47;
NET ED<11>              LOC = P50;
NET ED<12>              LOC = P127;
NET ED<13>              LOC = P132;
NET ED<14>              LOC = P138;
NET ED<15>              LOC = P36;
```


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

```
NET EFID<0>             LOC = P39;
NET EFID<1>             LOC = P142;
NET EFID<2>             LOC = P144;
NET EFID<3>             LOC = P147;
NET EFID<4>             LOC = P34;
NET EFID<5>             LOC = P52;
NET EFID<6>             LOC = P27;
NET EFID<7>             LOC = P154;
NET EFID<8>             LOC = P165;
NET EFID<9>             LOC = P156;
NET EFID<10>            LOC = P21;
NET EFID<11>            LOC = P164;
NET EFID<12>            LOC = P18;
NET EFID<13>            LOC = P167;
NET EFID<14>            LOC = P16;
NET EFID<15>            LOC = P15;

# Flash address bus
NET EFA<0>              LOC = P183;
NET EFA<1>              LOC = P184;
NET EFA<2>              LOC = P187;
NET EFA<3>              LOC = P188;
NET EFA<4>              LOC = P202;
NET EFA<5>              LOC = P203;
NET EFA<6>              LOC = P209;
NET EFA<7>              LOC = P210;
NET EFA<8>              LOC = P213;
NET EFA<9>              LOC = P214;
NET EFA<10>             LOC = P220;
NET EFA<11>             LOC = P221;
NET EFA<12>             LOC = P232;
NET EFA<13>             LOC = P233;
NET EFA<14>             LOC = P238;
NET EFA<15>             LOC = P239;
NET EFA<16>             LOC = P2;
NET EFA<17>             LOC = P3;
NET EFA<18>             LOC = P216;

NET EMEMCS1N            LOC = P87;
NET EMEMCS2N            LOC = P189;
NET EMEMRDN             LOC = P186;
NET EMEMWRN             LOC = P236;

NET ESCS1               LOC = P234;
NET ESCS2               LOC = P8;

#open collector buffer
NET EGAB1               LOC = P73;
NET EGBA1               LOC = P206;
NET ECAB1               LOC = P53;
NET ECBA1               LOC = P168;
NET ESAB1               LOC = P171;
NET ESBA1               LOC = P54;
```


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

```
NET EGAB2               LOC = P112;
NET EGBA2               LOC = P113;
NET ECAB2               LOC = P114;
NET ECBA2               LOC = P117;
NET ESAB2               LOC = P174;
NET ESBA2               LOC = P172;

NET EOCD1<0>            LOC = P33;
NET EOCD1<1>            LOC = P146;
NET EOCD1<2>            LOC = P149;
NET EOCD1<3>            LOC = P31;
NET EOCD1<4>            LOC = P162;
NET EOCD1<5>            LOC = P155;
NET EOCD1<6>            LOC = P24;
NET EOCD1<7>            LOC = P55;

NET EOCD2<0>            LOC = P23;
NET EOCD2<1>            LOC = P169;
NET EOCD2<2>            LOC = P102;
NET EOCD2<3>            LOC = P12;
NET EOCD2<4>            LOC = P237;
NET EOCD2<5>            LOC = P200;
NET EOCD2<6>            LOC = P84;
NET EOCD2<7>            LOC = P95;

#NET EPTCLK             LOC = P118;

NET EIRQ                LOC = P13;

#Sipex driver
NET ESPDR<0>            LOC = P81;
NET ESPDR<1>            LOC = P76;
NET ESPDR<2>            LOC = P77;
NET ESPDR<3>            LOC = P125;
NET ESPDR<4>            LOC = P191;
NET ESPDR<5>            LOC = P93;
NET ESPDR<6>            LOC = P43;
NET ESPDR<7>            LOC = P131;
NET ESPDR<8>            LOC = P136;
NET ESPDR<9>            LOC = P38;
NET ESPDR<10>           LOC = P231;
NET ESPDR<11>           LOC = P26;

#tristate driver enable
NET ESPDREN<0>          LOC = P194;
NET ESPDREN<1>          LOC = P228;
NET ESPDREN<2>          LOC = P225;
NET ESPDREN<3>          LOC = P205;
```


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

```
#Sipex receiver

NET ESPR<0>             LOC = P72;
NET ESPR<1>             LOC = P126;
NET ESPR<2>             LOC = P69;
NET ESPR<3>             LOC = P110;
NET ESPR<4>             LOC = P109;
NET ESPR<5>             LOC = P137;
NET ESPR<6>             LOC = P41;
NET ESPR<7>             LOC = P134;

#10 Mhz clock
NET ECLK_10Mhz          LOC = P107;

#40 Mhz clock
NET ECLK_40Mhz          LOC = P57;

#End of File
```


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

UF_STD.V

```
/*********************************************************************
***
 **
 ** Copyright (c) 1997 VXI Technology, Inc. All Rights Reserved
 **
 ** Project Name: VM6069 USER FPGA VXI REGISTER DECODING
 **
 ** Author: MANI
 **
 ** Revision History: 04/26/99
 **
 ** Date          Initials         Modification
 **
 **
 ** Description
 **
 **

*********************************************************************
**/


/*********************************************************************
***
 Module Name: main

 Description Top module for uf_std
*********************************************************************
**/
module      main
(ERCLK,EPFPGACLRN,ERD,EA29,EA,EBSELN,EASN,EDSN,ESIZ0,ESIZ1,ECSPACEN,EDSACK1N,
      ETRIGIN,ETRIGOUT,ED,EFID,EFA,EMEMCS1N,EMEMCS2N,EMEMRDN,EMEMWRN,ESCS1,ES
CS2,
      EOCD1,EOCD2,EGAB1,EGBA1,ECAB1,ECBA1,ESAB1,ESBA1,EGAB2,EGBA2,ECAB2,
      ECBA2,ESAB2,ESBA2,EIRQ,ESPDR,ESPDREN,ESPR,ECLK_40Mhz,ECLK_10Mhz);

input       ERCLK,EPFPGACLRN,ERD,EA29,EBSELN,EASN,EDSN,ESIZ0,ESIZ1,ECSPACEN;
input       ECLK_40Mhz,ECLK_10Mhz;
input       [5:0]EA;
input       ETRIGIN;

output      EDSACK1N,ETRIGOUT;
output      EIRQ;

inout       [15:0]ED;          //VMIP and Flash data bus
output      [18:0]EFA;         //Flash and Memory address bus
inout       [15:0]EFID;        //memory data bus

output      EMEMCS1N,EMEMCS2N,EMEMRDN,EMEMWRN;
reg         EMEMCS1N,EMEMCS2N,EMEMRDN,EMEMWRN;
```


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

```
//sipex latch signals
output      ESCS1,ESCS2;
reg         ESCS1,ESCS2;
//sipex driver signals
output      [11:0]ESPDR;
//sipex tristate drivers' enable signals
output      [3:0]ESPDREN;
//sipex receivers
input       [7:0]ESPR;

//Bidirectional Data bus control condition

reg         [15:0] vmip_readdata;
wire        [15:0] ED = ERD ? vmip_readdata :16'bz;

//Memory data bus bidirectional control conition
reg         memory_read;
reg         [15:0] memory_data;
wire        [15:0] EFID = memory_read ? 16'bz : memory_data;

reg         [3:0] state;
reg         access_done;
reg         access_memory;
reg         [18:0] memory_address;
reg         [3:0]mem_type;

//vmip dsack control
wire        dsack;
assign      EDSACK1N= dsack ? EASN: 'bZ;

//Open collector buffer
output      EGAB1,EGBA1,ECAB1,ECBA1,ESAB1,ESBA1;
output      EGAB2,EGBA2,ECAB2,ECBA2,ESAB2,ESBA2;

inout       [7:0] EOCD1;
inout       [7:0] EOCD2;
wire        [7:0] EOCD1;
wire        [7:0] EOCD2;
reg         [15:0] oc_data;

//Internal registers

reg         [15:0] reg22data;
reg         [15:0] reg24data;
reg         [15:0] reg26data;
reg         [15:0] reg28data;
reg         [15:0] reg2adata;
reg         [15:0] reg2cdata;
reg         [15:0] reg2edata;
reg         [15:0] reg30data;
reg         [15:0] reg32data;
```


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

```
reg         [15:0] reg34data;
reg         [15:0] reg36data;
reg         [15:0] reg38data;
reg         [15:0] reg3adata;
reg         [15:0] reg3cdata;

reg         [31:0]CLK10Counter;
reg         [31:0]CLK40Counter;

//vmip bus access/register select signals
wire        REG22CS,REG24CS,REG26CS,REG28CS,REG2aCS,REG2cCS,REG2eCS,
                REG30CS,REG32CS,REG34CS,REG36CS,REG38CS,REG3aCS,REG3cCS,
                REG3eCS,CSH,CSL,REG_ACCESS,RD_ACTIVE,WR_ACTIVE;
reg         Generate_Ack;

wire        reset_term;


//instantiate modules

//address decoder
addrdec Decoder
(.RCLK(ERCLK),.reset_term(reset_term),.RD(ERD),.A29(EA29),.A(EA),
                .BSELN(EBSELN), .ASN(EASN), .DSN(EDSN),
                .SIZ0(ESIZ0), .SIZ1(ESIZ1), .CSPACEN(ECSPACEN),
                .REG22CS(REG22CS), .REG24CS(REG24CS), .REG26CS(REG26CS),
                .REG28CS(REG28CS), .REG2aCS(REG2aCS), .REG2cCS(REG2cCS),
                .REG2eCS(REG2eCS), .REG30CS(REG30CS), .REG32CS(REG32CS),
                .REG34CS(REG34CS), .REG36CS(REG36CS), .REG38CS(REG38CS),
                .REG3aCS(REG3aCS), .REG3cCS(REG3cCS), .REG3eCS(REG3eCS),
                .CSH(CSH), .CSL(CSL), .REG_ACCESS(REG_ACCESS),
                .RD_ACTIVE(RD_ACTIVE), .WR_ACTIVE(WR_ACTIVE) );

//VXI bus acknowlegde
vmip_bus_ack Bus_Ack(.RCLK(ERCLK),.RESETN(EPFPGACLRN),.ASN(EASN),
                        .dsack(dsack),.Generate_Ack(Generate_Ack));

//10 Mhz clock counter
CLK_Counter CLK_10Mhz_Counter
(.CLK(ECLK_10Mhz),.CLRN(EPFPGACLRN),.Q(CLK10Counter));

//40Mhz clock counter
CLK_Counter CLK_40Mhz_Counter
(.CLK(ECLK_40Mhz),.CLRN(EPFPGACLRN),.Q(CLK40Counter));

//Triggerout line
assign ETRIGOUT=reg38data[1];

//Interrupt line
assign EIRQ=reg38data[0];

//memory address
assign EFA=memory_address;
```


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

```
//OC configure
assign ECAB1 =0;
assign ECBA1 =0;
assign ESAB1 =0;
assign ESBA1 =0;

assign ECAB2 =0;
assign ECBA2 =0;
assign ESAB2 =0;
assign ESBA2 =0;

//bit 0 for port0 and bit 1 for port1
//0 for output and 1 for input
assign EGAB1 =reg34data[0];
assign EGAB2 =reg34data[1];

assign EGBA1 =reg34data[0];
assign EGBA2 =reg34data[1];

//assign data
assign EOCD1 = reg34data[0] ? 8'bz : oc_data[7:0];
assign EOCD2 = reg34data[1] ? 8'bz : oc_data[15:8];

//assign sipex driver
assign ESPDR[0] = reg30data[0];
assign ESPDR[1] = reg30data[1];
assign ESPDR[2] = reg30data[2];
assign ESPDR[3] = reg30data[3];
assign ESPDR[4] = reg30data[4];
assign ESPDR[5] = reg30data[5];
assign ESPDR[6] = reg30data[6];
assign ESPDR[7] = reg30data[7];
assign ESPDR[8] = reg30data[8];
assign ESPDR[9] = reg30data[9];
assign ESPDR[10] = reg30data[10];
assign ESPDR[11] = reg30data[11];

//assign sipex tristate enable signals
assign ESPDREN[0]=reg30data[12];
assign ESPDREN[1]=reg30data[13];
assign ESPDREN[2]=reg30data[14];
assign ESPDREN[3]=reg30data[15];

//to reset the register select signals
assign reset_term=!EPFPGACLRN | EASN | (dsack & !ERD);


// start the flow
      always @ (posedge ERCLK)
      begin
            if( EPFPGACLRN==0 )
```


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

```
            begin
                  //clear all the variables
                  reg22data<=0;
                  reg24data<=0;
                  reg26data<=0;
                  reg28data<=0;
                  reg2adata<=0;
                  reg2cdata<=0;
                  reg2edata<=0;
                  reg30data<=0;
                  reg34data<=0;
                  reg36data<=0;
                  reg38data<=0;
                  reg3adata<=0;
                  reg3cdata<=0;
                  oc_data <= 0;
                  vmip_readdata <= 0;
                  Generate_Ack<=0;
                  access_memory<=0;
                  memory_read<=1;
                  mem_type<=0;
            end

//generate ack for memory read/write
      else if( (access_memory==1) && (access_done==1) )
            begin
                  if(memory_read==1)
                        begin
                              vmip_readdata<=memory_data;
                        end
                  Generate_Ack<=1;
            end
      else if ( EASN==1 )
            begin
                  Generate_Ack<=0;
                  access_memory<=0;
            end
      else
            begin
 //vmip write cycle
                  if( (REG_ACCESS==1) && (WR_ACTIVE==1) )
                        begin
                              if(REG22CS)
                                    begin
                                          reg22data<=ED;
//memory1 higher address
                                          Generate_Ack<=1;
                                    end
                              if(REG24CS)
                                    begin
                                          reg24data<=ED;
//memory1 lower address
                                          Generate_Ack<=1;
                                    end
```


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

```
                              if(REG26CS)
      //memory1 data
                                    begin
                                          reg26data<=ED;
                                          memory_read<=0;
                                          access_memory<=1;
                                          mem_type<=0;
                                    end
                              if(REG28CS)
                                    begin
                                          reg28data<=ED;
      //memory2 higher address
                                          Generate_Ack<=1;
                                    end
                              if(REG2aCS)
                                    begin
                                          reg2adata<=ED;
      //memory2 lower address
                                          Generate_Ack<=1;
                                    end
                              if(REG2cCS)
      //memory2 data
                                    begin
                                          reg2cdata<=ED;
                                          memory_read<=0;
                                          access_memory<=1;
                                          mem_type<=1;
                                    end
                              if(REG2eCS)
      //sipex latch
                                    begin
                                          reg2edata<=ED;
                                          memory_read<=0;
                                          access_memory<=1;
                                          mem_type<=2;
                                    end
                              if(REG30CS)
      //sipex drivers
                                    begin
                                          reg30data<=ED;
                                          Generate_Ack<=1;
                                    end
                              if(REG32CS)
      //sipex receivers-no write
                                    begin
                                          Generate_Ack<=1;
                                    end
                              if(REG34CS)
      //OC configure
                                    begin
                                          reg34data<=ED;
                                          Generate_Ack<=1;
                                    end
```


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

```
                              if(REG36CS)
//OC data
                                    begin
                                          oc_data<=ED;
                                          Generate_Ack<=1;
                                    end
                              if(REG38CS)
//Misc
                                    begin
                                          reg38data<=ED;
                                          Generate_Ack<=1;
                                    end
                        end
//vmip read cycle
                  else if( (REG_ACCESS==1) && (RD_ACTIVE==1) )
                        begin
                              if(REG22CS)
                                    begin
//memory1 higher address
                                          vmip_readdata<=reg22data;
                                          Generate_Ack<=1;
                                    end
                              if(REG24CS)
                                    begin
//memory1 lower address
                                          vmip_readdata<=reg24data;
                                          Generate_Ack<=1;
                                    end
                              if(REG26CS)
                                    begin
//memory1 data
                                          memory_read<=1;
                                          access_memory<=1;
                                          mem_type<=0;
                                    end
                                    if(REG28CS)
                                    begin
//memory2 higher address
                                          vmip_readdata<=reg28data;
                                          Generate_Ack<=1;
                                    end
                              if(REG2aCS)
                                    begin

//memory2 lower address
                                          vmip_readdata<=reg2adata;
                                          Generate_Ack<=1;
                                    end
                              if(REG2cCS)
                                    begin
//memory2 data
                                          memory_read<=1;
                                          access_memory<=1;
                                          mem_type<=1;
                                    end
```


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

```
                                    if(REG2eCS)
                                          begin
      //sipex latch
                                                vmip_readdata<=reg2edata;
                                                Generate_Ack<=1;
                                          end
                                    if(REG30CS)
                                          begin
      //sipex drivers
                                                vmip_readdata<=reg30data;
                                                Generate_Ack<=1;
                                          end
                                    if(REG32CS)
                                          begin
      //sipex receivers
                                                vmip_readdata<=0;
                                                reg32data[7:0]<=ESPR[7:0];
                                                vmip_readdata<=reg32data;
                                                Generate_Ack<=1;
                                          end
                                    if(REG34CS)
                                          begin
      //OC configure
                                                vmip_readdata<=reg34data;
                                                Generate_Ack<=1;
                                          end
                                    if(REG36CS)
                                          begin
      //OC data
                                                vmip_readdata<=0;
                                    if(reg34data[0]==1)

      vmip_readdata[7:0]<=EOCD1[7:0];
                                    if(reg34data[1]==1)

      vmip_readdata[15:8]<=EOCD2[7:0];
                                                Generate_Ack<=1;
                                          end
                                    if(REG38CS)
                                          begin
      //Misc
      //save trignin status
                                                reg38data[2]<=ETRIGIN;
                                                vmip_readdata<=reg38data;
                                                Generate_Ack<=1;
                                          end
                                    if(REG3aCS)
                                          begin
      //Clk 10 Mhz


      reg3adata[15:0]<=CLK10Counter[31:16];
                                                vmip_readdata<=reg3adata;
                                                Generate_Ack<=1;
                                          end
```


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

```
                              if(REG3cCS)
                                    begin
//Clk 40 Mhz

reg3cdata[15:0]<=CLK40Counter[31:16];
                                          vmip_readdata<=reg3cdata;
                                          Generate_Ack<=1;
                                    end
                        end
            end
end

//memory read/write
always @ (posedge ERCLK or posedge reset_term)
begin
      if( reset_term )
            begin
                  access_done<=0;
                  state<=0;
                  EMEMCS1N<=1;
                  EMEMCS2N<=1;
                  EMEMRDN<=1;
                  EMEMWRN<=1;
                  ESCS1<=1;
                  ESCS2<=1;
                  memory_address<=0;
            end
      else if(access_memory==1)
            begin
                  case(state)
                        0:
                              begin
                                    case(mem_type)
                                          0:
//RAM1
                                                begin
//place address
                              memory_address[18:16]<=reg22data[2:0];
                              memory_address[15:0]<=reg24data[15:0];
//place data for write
                                    if(memory_read==0)
                              memory_data<=reg26data;
                                                end
                                          1:
//RAM2
                                                begin
//place address
                              memory_address[18:16]<=reg28data[2:0];
                              memory_address[15:0]<=reg2adata[15:0];
//place data for write
                                    if(memory_read==0)
                              memory_data<=reg2cdata;
                                                end
```


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

```
                                        2:
//sipex latch
                                              begin
//place data for write latch
                                        if(memory_read==0)

                                  memory_data<=reg2edata;
                                              end
                                        default:;
                                  endcase
                                  state<=1;
                                  end
                            1:
                                  begin
                                  case(mem_type)
                                        0:    EMEMCS1N<=0;
//place chip select for RAM
                                        1: EMEMCS2N<=0;
                                        2:
                                              begin
//place clock for sipex latch
                                                    ESCS1<=0;
                                                    ESCS2<=0;
                                              end
                                        default:;
                                  endcase
                                  state<=2;
                                  end
                            2:
                                  begin
//place write signal for write or
//read signal for read
                                  case(mem_type)
                                        0,1:
                                              begin
if(memory_read==1)
EMEMRDN<=0;
                                              else
EMEMWRN<=0;
                                              end
                                              default:;
                                  endcase
//remove clock from latch-data is stable now
                                  ESCS1<=1;
                                  ESCS2<=1;

                                  state<=3;
                                  end
                            3: state<=4;
```


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

```
                              4:
                                    begin
                                          case(mem_type)
                                                0,1:
      //read data for read
                                                      if(memory_read==1)

      memory_data<=EFID;
                                                default:;
                                          endcase

                                          EMEMCS1N<=1;
                                          EMEMCS2N<=1;
                                          EMEMRDN<=1;
                                          EMEMWRN<=1;
                                          ESCS1<=1;
                                          ESCS2<=1;
                                          state<=5;
                                    end
                              5:
                                    begin
                                          state<=6;
                                          access_done<=1;
                                    end
                              6:
                                    begin
      //state=6;
      //access_done=1;
                                    end
                              default:;
                        endcase
                  end
      end

endmodule
```


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

```
/*************************************************************************
***
 Module Name: addrdec

 Description
      This is the VXI register decoding. It generates one of the 15 register
      chip select(active high),internal register select signal, low byte
      and high byte select signals and read & write signals
*************************************************************************
**/
module      addrdec
(RCLK,reset_term,RD,A29,A,BSELN,ASN,DSN,SIZ0,SIZ1,CSPACEN,

      REG22CS,REG24CS,REG26CS,REG28CS,REG2aCS,REG2cCS,REG2eCS,REG30CS,
                  REG32CS,REG34CS,REG36CS,REG38CS,REG3aCS,REG3cCS,REG3eCS,
                  CSH,CSL,REG_ACCESS,RD_ACTIVE,WR_ACTIVE);

input       RCLK,reset_term,RD,A29,BSELN,ASN,DSN,SIZ0,SIZ1,CSPACEN;
input       [5:0]A;

output      REG22CS,REG24CS,REG26CS,REG28CS,REG2aCS,REG2cCS,REG2eCS,REG30CS,

      REG32CS,REG34CS,REG36CS,REG38CS,REG3aCS,REG3cCS,REG3eCS,CSH,CSL,
                  REG_ACCESS;

reg         REG22CS,REG24CS,REG26CS,REG28CS,REG2aCS,REG2cCS,REG2eCS,REG30CS,

      REG32CS,REG34CS,REG36CS,REG38CS,REG3aCS,REG3cCS,REG3eCS,CSH,CSL,
                  REG_ACCESS;

output      RD_ACTIVE,WR_ACTIVE;
reg         RD_ACTIVE,WR_ACTIVE;

// Address Decoding Logic Equations
      always @ (posedge RCLK or posedge reset_term)

      begin
            if( reset_term )
            begin
                  //clear all chip selects when reset or ASN is high
                  REG22CS<=0; REG24CS<=0; REG26CS<=0; REG28CS<=0; REG2aCS<=0;
                  REG2cCS<=0; REG2eCS<=0; REG30CS<=0; REG32CS<=0; REG34CS<=0;
                  REG36CS<=0; REG38CS<=0; REG3aCS<=0; REG3cCS<=0; REG3eCS<=0;
                  CSH<=0; CSL<=0; REG_ACCESS<=0;
                  RD_ACTIVE<=0; WR_ACTIVE<=0;
            end
            else if(ASN ==0 && DSN ==0 && CSPACEN== 1 && A29==1 && BSELN==0)

            begin

                  case(A)
                              2: REG22CS<=1;
                              4: REG24CS<=1;
                              6: REG26CS<=1;
                              8: REG28CS<=1;
```


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

```
                              'ha: REG2aCS<=1;
                              'hc: REG2cCS<=1;
                              'he: REG2eCS<=1;
                              'h10: REG30CS<=1;
                              'h12: REG32CS<=1;
                              'h14: REG34CS<=1;
                              'h16: REG36CS<=1;
                              'h18: REG38CS<=1;
                              'h1a: REG3aCS<=1;
                              'h1c: REG3cCS<=1;
                              'h1e: REG3eCS<=1;

                        default:;
                        endcase
      //set a signal to indicate  register is selected and
      //assign low and high byte signals
            if (REG22CS==1 || REG24CS==1 ||REG26CS==1 ||REG28CS==1
      ||REG2aCS==1 ||
                        REG2cCS==1 ||REG2eCS==1  ||REG30CS==1 ||REG32CS==1
                        ||REG34CS==1 ||REG36CS==1 ||REG38CS==1  ||REG3aCS==1
                        ||REG3cCS==1 ||REG3eCS==1 )
                        begin
                              REG_ACCESS<=1;
                              if( (SIZ1==1 && A[0]==0) || (SIZ0==1 && A[0]==1
) )
                                    CSH<=1;
                              if( (SIZ1==1 && A[0]==0) || (SIZ0==1 && A[0]==0
) )
                                    CSL<=1;
      //in motorola convention higher byte(d15-d8) is stored in lower
      //memory(A0 is low) and lower byte(d7-d0) is stored in higher
      //memory(A0 is high).
      //CSL is high when lower memory is selected and CSH is high when
      //higher memory is selected
      //select read or write signal
                              if(RD)
                                    RD_ACTIVE<=1;
                              else
                                    WR_ACTIVE<=1;
                        end
                  end
      end
endmodule
```


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

```
/*************************************************************************
***
 Module Name: vmip_bus_ack

 Description Generates DSACK signal to vmip bus
*************************************************************************
**/
module      vmip_bus_ack (RCLK,RESETN,ASN,dsack,Generate_Ack);

input       RCLK,RESETN,ASN,Generate_Ack;

output      dsack;
reg         dsack;

wire        r_reset = !RESETN | ASN ;

always @ (posedge RCLK or posedge  r_reset)
      begin
            if( r_reset)
                  begin
                        dsack<=0;
                  end
            else if(Generate_Ack==1)
                  begin
                        dsack<=1;
                  end
      end
endmodule

/*************************************************************************
***
 Module Name: CLK_Counter

 Description Counter
*************************************************************************
**/
module      CLK_Counter (CLK,CLRN,Q );

input       CLK,CLRN;

output      [31:0]Q;
reg         [31:0]Q;

always @ (posedge CLK or negedge CLRN)
      begin
            if (CLRN==0)
                  Q <= 0;
            else
                  Q <= Q + 1;
      end
endmodule
/*********************************END*************************************
**/
```


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

LOOPBACK CONNECTOR USED TO TEST VM6069

UUT P1	Output Signal Name	Connect Via	UUT P1	Input Signal Name
1	TXD1+	30awg Black wire (5cm)	3	RXD1+
2	TXD1-	30awg Black wire (5cm)	4	RXD1-
3	RXD1+			
4	RXD1-			
5	RTS1+	30awg Black wire (5cm)	7	CTS1+
6	RTS1-	30awg Black wire (5cm)	8	CTS1-
7	CTS1+			
8	CTS1-			
9	DTR1+	30awg Black wire (5cm)	11	DSR1+
10	DTR1-	30awg Black wire (5cm)	12	DSR1-
11	DSR1+			
12	DSR1-			
13	TXC1+	30awg Black wire (5cm)	15	RXC1+
14	TXC1-	30awg Black wire (5cm)	16	RXC1-
15	RXC1+			
16	RXC1-			
17	ST1+	30awg Black wire (5cm)	25*	RXD1+
18	ST1-	30awg Black wire (5cm)	26*	RXD1-
19	RL1+	30awg Black wire (5cm)	29*	CTS2+
20	RL1-	30awg Black wire (5cm)	30*	CTS2-
21	GND	N/C		
22	GND	N/C		
23	TXD2+	30awg Black wire (5cm)	25	RXD2+
24	TXD2-	30awg Black wire (5cm)	26	RXD2-
25	RXD2+			
26	RXD2-			
27	RTS2+	30awg Black wire (5cm)	29	CTS2+
28	RTS2-	30awg Black wire (5cm)	30	CTS2-
29	CTS2+			
30	CTS2-			
31	DTR2+	30awg Black wire (5cm)	33	DSR2+
32	DTR2-	30awg Black wire (5cm)	34	DSR2-
33	DSR2+			
34	DSR2-			
35	TXC2-	30awg Black wire (5cm)	37	RXC2-
36	TXC2+	30awg Black wire (5cm)	38	RXC2+
37	RXC2-			
38	RXC2+			
39	ST2-	30awg Black wire (5cm)	4	RXD1-
40	ST2+	30awg Black wire (5cm)	3	RXD1+
41	RL2-	30awg Black wire (5cm)	8	CTS1-

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

42	RL2+	30awg Black wire (5cm)	7	CTS1+
43	OCD0	30awg Black wire (5cm)	61	OCD8
44	OCD1	30awg Black wire (5cm)	62	OCD9
45	OCD2	30awg Black wire (5cm)	63	OCD10
46	OCD3	30awg Black wire (5cm)	64	OCD11
47	OCD4	30awg Black wire (5cm)	65	OCD12
48	OCD5	30awg Black wire (5cm)	66	OCD13
49	OCD6	30awg Black wire (5cm)	67	OCD14
50	OCD7	30awg Black wire (5cm)	68	OCD15
51	X	N/C		
52	X	N/C		
53	X	N/C		
54	X	N/C		
55	X	N/C		
56	X	N/C		
57	X	N/C		
58	X	N/C		
59	X	N/C		
60	X	N/C		
61	DOUT1			
62	DOUT2			
63	DOUT3			
64	DOUT4			
65	DOUT5			
66	DOUT6			
67	DOUT7			
68	DOUT8			

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

APPENDIX B - VM6069 SCHEMATIC

50-0110-000 - SCHEMATIC, VM6069, UNIVERSAL SERIAL INTERFACE

To view, or print, the schematic for the VM6069, open the VM6069.pdf file in the **Manuals** directory on the *Product Manuals and Drivers* CD supplied with this manual.

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

INDEX

***

*CLS 61
*ESE 62
*ESR? 63
*IDN? 64
*OPC 65
*RST 66
*SRE 67
*STB? 68
*TRG 69
*TST? 70
*WAI 71

B

backplane 16, 55
backplane jumpers 15, 16

C

CALibration:SECure:CODE 72
CALibration:SECure[:STATe] 73
command set 55
communication protocol 12
cooling 15

D

device dependant register 11, 12, 21
differential drivers 12
differential receivers 12
dynamic address configuration 16

F

FIFO 12, 21
flash memory 12, 21, 22
front panel 17

H

hand-shaking 22

I

interface 11, 12, 22
interface FPGA 22

K

keyword 56

L

logical address 15, 16, 17

M

message-based 55

O

open collector transceivers 12
operational status 80
OUTPut:TTLTrg 75
OUTPut[:STATe] 74
OUTPut[:TTLTrg]:POLarity 76

P

parameter 56
parameters 55
peripheral I/O 17
peripherals 13
power 15, 16, 63
PROGram:MODule 77
programming language 55
protocol FPGA 22

R

register-based 21

S

SCSI connector 17
serial transceivers 12
standard serial interface 12
STATus:OPERation:ENABle 82
STATus:OPERation[:EVENt]? 85
STATus:PRESet 83, 84, 86
STATus:QUEStionable:CONDition? 87
STATus:QUEStionable:ENABle 88
STATus:QUEStionable[:EVENt]? 89
syntax 55

T

tree-structured language 55
TRIGger:SOURce:TTLTrg 78
TRIGger:[STATe] 79

U

universal serial interface 11, 12
user FPGA 11, 12
USI 12

V

VHDL/VERILOG 21
VHDL/VERILOG design code 22
VMIP 11, 16, 17
VXIbus 11, 12, 15, 17, 21, 22, 55

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