KineticSystems' Model

V155 User's Manual

V155 User's Manual
Release 1.0

Table Of Contents

Chapter 1: Introduction

Description

The V155 is a single-width, C-size module that combines the performance of a Pentium M-based computer with the functionality of a VXI Slot-0 controller. Alternatively, on-board strap options enable this module to be used as a VXI-based processor in non-Slot-0 applications.

The V155's optional External CD-ROM drive allows for the installation of a number of popular programming environments and compilers, including Microsoft Visual C++, Borland C++, and Microsoft Visual Basic.

For soft real-time applications, the V155 coupled with the Windows XP Pro operating system is a cost-effective and high-performance Slot-0 controller solution for your data collection needs.

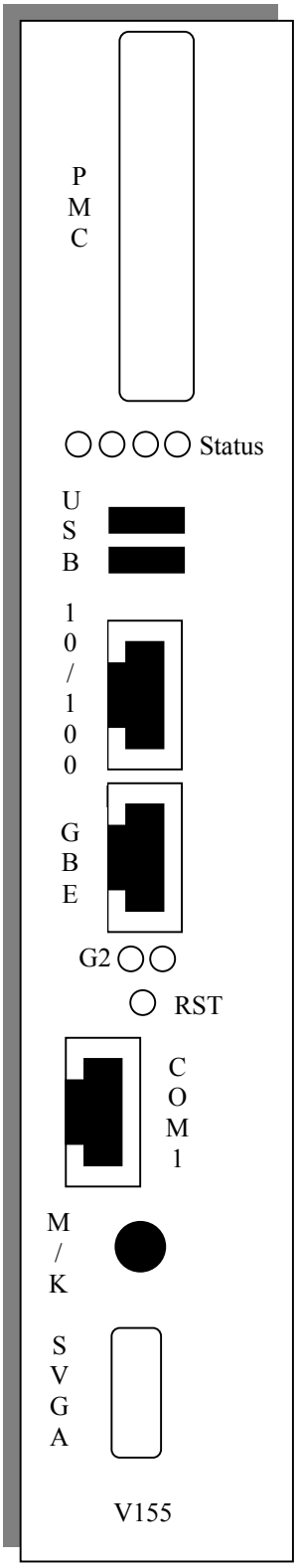
V155 Specifications

Item	Specification
Processor Options	Pentium M Processor 1.1 GHz Pentium M Processor 1.6 GHz Pentium M Processor 1.8 GHz
DDR SDRAM	Options available with 512Mbyte, 1Gbyte and 1.5Gbyte
Internal Hard Drive	40 GB IDE Hard Drive, contact KineticSystems for latest hard drive offerings
Timers and Watchdog Timer	Two 16-bit timers and two 32-bit timers and software programmable watchdog timer
Compact Flash Memory	Supports up to 2 GB of Compact Flash through secondary IDE port
RTC/NVSRAM	Real Time Clock and 32 KB of nonvolatile SRAM
Ethernet Interface	Dual Ethernet controllers supporting 10BaseT, 100BaseTX and 1000BaseT interfaces
PMC Slot	5 V signaling, types 1 and 2, 32-bit PCI bus, 66 MHz maximum
AGP Support	Internal AGP SVGA controller and Digital Visual Interface (DVI)
USB Port	Supports 2 USB Rev. 2.0
Environmental and Mechanical Temperature Range Operational Storage Relative Humidity	 0 to +50 degrees C -25 to 75 degrees C 0 to 86% non-condensing to 40 degrees C
Vibration and Shock Operating Vibration Operating Shock Non-Operating Shock	 9.8m/s^2 (1.0G), 5-500 Hz 1470m/s^2 (150G) 6860m/s^2 (700G)
Power Requirements +5V -5.2V -2V +12V -12V	 9500 mA 270 mA 96 mA 100 mA 100 mA

Technical specifications contained within this publication are subject to change without notice.

Table 1-1. Specifications

Front Panel



The V155 font-panel has all the standard external connections for monitor, mouse, keyboard, Ethernet, USB, and COM ports.

- 1 PMC site for high speed reflective memory or high speed digital I/O.
- The 4 Status LEDs (from left to right) indicate:
  - Boot LED—indicates BIOS boot in progress (red)
  - Power LED—indicates when power is applied; also indicates throttling by blinking slowly (green)
  - IDE LED—indicates when IDE activity is occurring (yellow)
  - VME LED—indicates VME SYSFAIL condition (red)
- 2 USB ports support standard USB Interface Rev 2.0
- 2 Ethernet ports:
  - 10/100—supports 10BaseT and 100BaseTX Ethernet options.
  - GBE—supports 10BaseT, 100BaseTX and Gigabit Ethernet options.
- Gigabit Ethernet LEDs
- The recessed RST button applies SYSRESET to the VME backplane and resets (reboots) the SBC microprocessor.
- 1 RJ45 connector16550-compatible serial port.
- 1 6-pin PS-2 combined keyboard and mouse connector.
- 1 standard SVGA video connector.

Product Ordering Information

Model V155-AB14 VMIC 7807 P4, 1.8 GHz, 1.5 GB, 40 GB Hard Disk
Model V155-AB13 VMIC 7807 P4, 1.8 GHz, 1 GB, 40 GB Hard Disk
Model V155-AB12 VMIC 7807 P4, 1.8 GHz, 512 MB, 40 GB Hard Disk

Related Products

V155-EXCD External CD-ROM USB
V155-KBMS Keyboard and Mouse
LX40-ALA6 KineticSystems VISA Library
DX10-ALA6 VMIC Plug-In Driver
V155-OSIN Windows XP Pro Installed on Hard Disk
V155-BSP VMIC Board Support Package

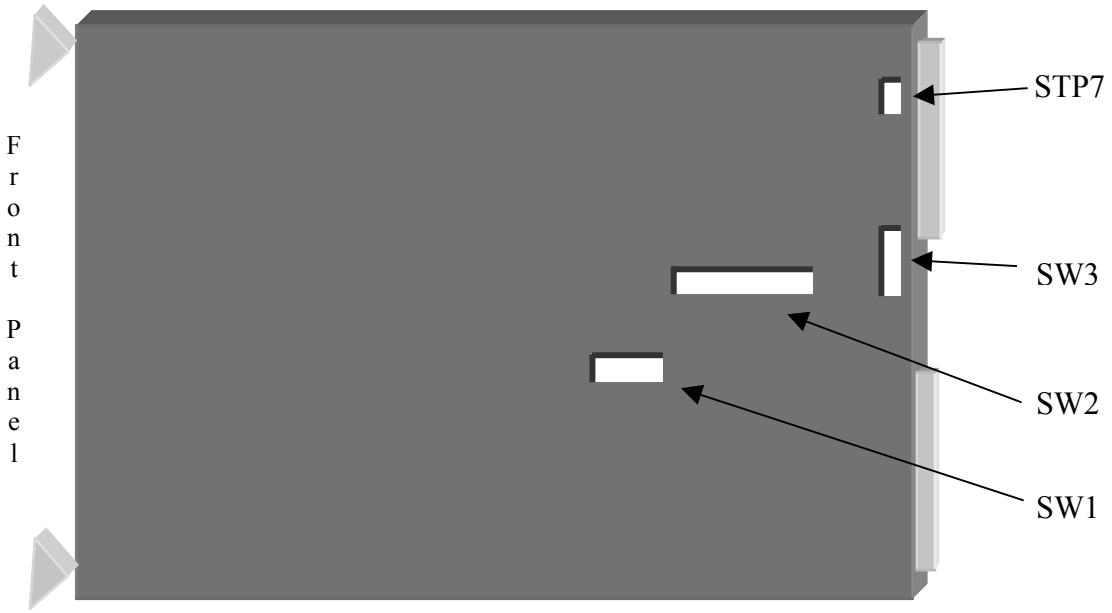
Chapter 2: Installation and Configuration

Unpacking the V155

The Model V155 is shipped in an anti-static bag with a Styrofoam packing container. Carefully remove the module from its anti-static bag and prepare to set the various options to conform to the operating environment. Make sure that all anti-static precautions are taken to avoid damaging the module. Please take the following precautions when unpacking the module:

- When handling module, use a grounding strap or touch a grounded object.
- Touch the anti-static package to a metal part of your VXI chassis before removing the module from the package.
- Remove the module from the package and inspect the module for damage.
- Do not install the module into the VXI chassis until you are satisfied that the module exhibits no obvious mechanical damage and is configured to conform to the desiring operating environment.

The V155 requires various strap and switch selections to be set before installing the module in the VXI chassis, located as follows:



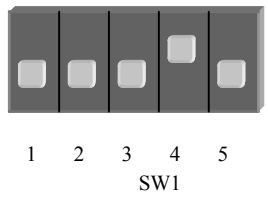
The V155 may be configured to operate as either a Slot0 controller or a non-Slot0 controller. When the V155 is the Slot0 controller, it must be located in the left-most slot (Slot0) and be set for Logical Address 0. If the V155 is not the Slot0 controller, it may be located in any other slot in the chassis and set for Logical Address 1 through 255. Selection of Slot-0/Non Slot-0 operation is made via settings of the SW1, SW3, and STP7 settings, as described below. By default, the V155 is shipped from the factory configured as a slot-0 controller.

Selecting Slot-0 Operation

For Slot-0 operation, SW1, SW2, SW3, and STP7 should be configured as:

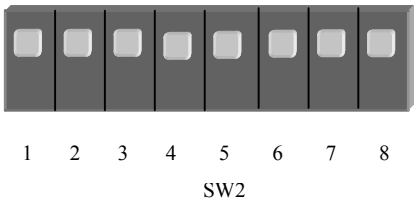
SW1

Switch	Position
1	down
2	down
3	down
4	up
5	down



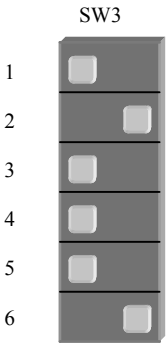
SW2

Switch	Position
1	up
2	up
3	up
4	up
5	up
6	up
7	up
8	up



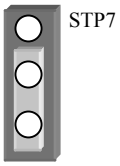
SW3

Switch	Position
1	left
2	right
3	left
4	left
5	left
6	right



STP7

Switch	Position
strap	down

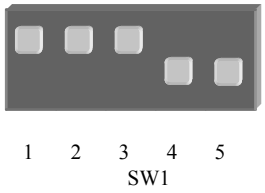


Selecting Non Slot-0 Operation

For Non Slot-0 operation, SW1, SW3, and STP7 should be configured as:

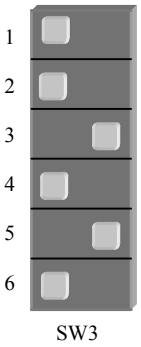
SW1

Switch	
1	up
2	up
3	up
4	down
5	down



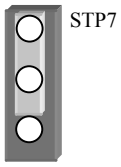
SW3

Switch	Position
1	left
2	left
3	right
4	left
5	right
6	left



STP7

Switch	Position
strap	up



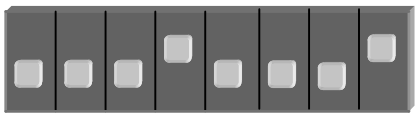
Note that the logical address switch (SW2) must be configured to a value *not* 0 (not all up).

Setting the Logical Address

The Logical address of the V155 module is set via SW2. The individual switches of SW2 represent a binary combination of numbers in the range of 0 to 255. The switch settings are made by depressing each Logical Address switch to the desired location. A switch that is in the DOWN position yields a bit set to a one. A switch that is in the UP position yields a bit set to a zero. The left-most switch corresponds to Logical Address bit 128 and the right-most switch corresponds to Logical Address bit 1, as shown below:

SW2

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



1 2 3 4 5 6 7 8

SW2

Setting shown: 2 + 4 + 8 + 32 + 64 + 128 =
Logical Address 238

To statically assign a Logical Address to the V155, simply set the 8-position DIP switch to the desired Logical Address in the range of 0 through 254. This sets the Logical Address of the V155 and may only be altered by changing the setting on the DIP switch.

The V155 may also be Dynamically Configured. A device that is Dynamically Configured must have its Logical Address set to 255 (FF_{16}). A device that is Dynamically Configured has its Logical Address set by the Resource Manager when the Logical Address Register of the V155 is written. Dynamic Configuration is used to avoid conflicts in setting up a device's Logical Address.

Installation

Module Installation

After all the user selectable configuration parameters have been setup, the module may then be inserted into the VXI chassis. If the V155 is configured for Slot0 operation, insert the V155 into the left-most slot (Slot0) of the VXI chassis. For a non-Slot0 configuration, insert the V155 into any slot in the range of 1 through 13.

CAUTION:	**TURN OFF MAINFRAME POWER BEFORE INSERTING OR REMOVING A VXI*bus* MODULE.**

WARNING:	**REMEMBER TO REMOVE THE INTERRUPT ACKNOWLEDGE AND BUS GRANT DAISY CHAIN JUMPERS BEFORE INSERTING A VXI MODULE**

The VXI*bus* backplane must be properly configured before inserting a VXI module and applying power. The Interrupt Acknowledge jumper must be removed from the slot in which the VXI module is to be inserted. The Bus Grant jumpers must also be removed from the slot in which the VXI module is to be inserted. All unoccupied slot locations must have the Interrupt Acknowledge and Bus Grant jumpers installed so that the interrupt and grant continuity is not disrupted by any open slots. When using backplanes that auto-configure, these steps are not necessary since the installation of a VXI module in the chassis makes the required configuration occur.

Operating System Installation

Installation of the Operating System is performed exactly as if the V155 where an ordinary laptop or desktop system.

OS installation requires an external USB CD drive.

At this point in time, only Windows XP is supported.

Software Installation

The V155 can be operated as a VME-based controller by simply installing the included VMIC IO Works software and using their VMIC VMEbus Access API; the VMIC installation disk contains full on-line documentation.

The V155 can also be operated as a full VXI-based controller by purchasing the optional KineticSystems VISA & VISA VMIC Plug-in.

VMIC VMEbus Access

The VMIC VMEbus Access Software contains all the drivers and support files needed for VME access on the V155 SBC, and should be installed first. Insert the VMIC disk and run the appropriate setup.exe.

KineticSystems VISA Installation

If slot-0 VISA-level access is desired, KineticSystems VISA and the VISA VMIC Plug-in will need to be purchased separately and installed. Since the VISA VMIC Plug-in is built on top of the VMIC Access Software, the VMIC layer must always be installed even if no application program intends to directly access the VMIC API.

VISA is installed first. If KineticSystems VISA was already installed on the system as part of a previous KineticSystems product installation, this step is omitted. Insert the KineticSystems VISA disk and run the appropriate setup.exe.

The VISA VMIC Plug-in is installed last. Insert the KineticSystems VISA VMIC Plug-in disk and run the appropriate setup.exe.

Directory Structure

The V155 installation creates the directory structure shown below.
Folders are represented by the (→) symbol. Files are represented by the (•) symbol.

→VXIPNP Root level (VPNPPATH)
- **→WINNT**
  - **→ Bin**
    - pisadevimpvmicvxi.dll
    - pisa.dll
    - resman.exe
  - **→ Include**
    - pisavxi.h
    - visa.h
    - visatype.h
    - vpptype.h
  - **→ Lib**
    - **→ Cvi** (*appropriate for use in CVI compiler projects*)
      - visa32.lib
    - **→ Msc** (*appropriate for use in Microsoft Visual C/C++ projects*)
      - visa32.lib

→WinNT/Windows (System Root)
- **→ system32**
  - visa32.dll

VMIC Plug-in Resman options

The KineticSystems VISA *resman* utility accesses theVISA device manager to probe for hardware installed on the system.

The VMIC Plug-in adds 1 new option to resman:

- `--vmicDelay <timeout>`

  This option specifies the number of seconds to wait after SYSRESET before probing for VXI modules controlled by a V155. By default, there is a 5 second delay between the de-assertion of SYSRESET and probing for hardware devices; this period allows devices time to perform their reset and self test routines. This option changes the length of the delay. timeout is in seconds.

Please consult your KineticSystems VISA manual for more information on *resman*.

Chapter3: VXI*bus* Configuration Registers and Operational Registers

The following table shows the various registers located in A16 space for the V155:

A16 Offset	Write Access	Read Access
00_{16}	Logical Address Register	Identification
02_{16}	Reserved	Device Type Register
04_{16}	Status/Control Register	Status/Control Register
06_{16}	Reserved	Reserved
08_{16}	MODID Register	MODID Register
$0A_{16}$	Reserved	Reserved
$0C_{16}$	Reserved	Reserved
$0E_{16}$	Reserved	Reserved
10_{16}	Reserved	Reserved
12_{16}	Reserved	Reserved
14_{16}	Reserved	Reserved
16_{16}	Reserved	Reserved
18_{16}	Reserved	Reserved
$1A_{16}$	Reserved	Reserved
$1C_{16}$	Reserved	Reserved
$1E_{16}$	Reserved	Reserved
20_{16}	Reserved	Suffix High Register
22_{16}	Reserved	Suffix Low Register
24_{16}	Reserved	Serial Number High Register
26_{16}	Reserved	Serial Number Low Register
28_{16}	Reserved	Reserved
$2A_{16}$	Reserved	Interrupt Status Register
$2C_{16}$	Interrupt Control Register	Interrupt Control Register
$2E_{16}$	Trigger Interrupt Mask	Trigger Interrupt Source
30_{16}	Trigger Interrupt Source Clear	Reserved
32_{16}	Trigger Source Register	Reserved
34_{16}	Trigger Timer Configuration Register	Reserved
36_{16}	Reserved	Reserved
38_{16}	SBC Slave Mode Configuration	Interrupt Status ID Register
$3A_{16}$	Reserved	Reserved
$3C_{6}$	Miscellaneous Control Register	Reserved

ID/Logical Address Register

The ID/Logical Address Register is a write/read register located at an offset of 00_{16} from the A16 Logical Base Address. A read operation to this register returns the Device Class, the addressing modes of the device's operational registers and the Manufacturers' Identification. A write operation to this register address is typically executed by the Resource Manager during a Dynamic Configuration allocation sequence. During the sequence, the Resource Manager allocates a Logical Address to the V155 by writing a logical address value to the least significant eight bits of this register. The format and bit assignments of this register are shown in the following diagram. Since this register has write-only and read-only bits, two bit patterns are shown.

On read transactions:

00h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	1	1	1	1	1	1	1	1	0	0	1	0	1	0	0	1

Class = Register based | **Addressing Mode = A16** | **KineticSystems' Manufacturer ID = $F29_{16}$ (3881)**

Bit(s)	Mnemonic	Meaning
15:14	Device Class	These bits are set to reflect the Device Class of the V155. This bit combination indicates that the V155 is a Register based Device.
13:12	Address Space	These bits are set to reflect the addressing mode(s) of the V155's operational registers. Since all the communication registers of the V155 appear in A16 address space, the bits in this field are both set to one.
11:0	Manufacturer	This field reflects the manufacturer of a VXI device. This value is 3881($F29_{16}$) for KineticSystems.

On write transactions:

00h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	Not Used								Logical Address							

Bit(s)	Mnemonic	Meaning
15:8	Not Used	These 8 bits are not used. A write operation to these bits has no effect on the V155.
7:0	128-LA1	Logical Address 128 through 1 are write-only bits used to set the V155's Logical Address during a Dynamic Configuration cycle executed by the Resource Manager. A Dynamic Configuration sequence is performed on a VXI module when its logical address has been set to 255 (FF_{16}).

Device Type Register

The Device Type Register is a read-only register located at an offset of 02_{16} from the A16 Logical Base Address of the V155. This register contains the Model Code of the V155. Since the V155 is an A16-only device, the entire 16-bits of this field is used for the Model Code.

Model Codes for VXI Slot0 devices must be in the range of 00_{16} to FF_{16}. Model Codes for non-Slot0 devices must be in the range of 100_{16} to $FFFF_{16}$. When the V155 is configured for non-Slot0 operation, the Model Code returned in this register is 155_{16}. When the V155 is configured for Slot0 operation, the 100_{16} bit is set to zero, yielding a Model Code of 55_{16}.

V155 Model Codes:

155_{16} for non-Slot0 configurations
55_{16} for Slot0 configurations

The following diagram shows the bit pattern for the Device Type Register for both Slot0 and non-Slot0 configurations.

For Slot0 Configurations:

02h

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

For non-Slot0 Configurations:

02h

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

Status/Control Register

The Status/Control Register is a write/read register located at an offset of 04_{16} from the A16 Logical Base Address of the V155. This register contains write-only, read-only and write/read bits. This register is used to monitor the Module ID VXI signal, control the assertion of SYSFAIL, control Soft Reset, and check the status of the Power-On Self Test. The following two diagrams show the Status/Control Register, one for read accesses and one for write.

For read operations executed to the Status/Control Register:

04h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Write	**0**	**MOD ID***	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**RDY**	**PASS**	**SYS INH**	**SFT RST**

Bit(s)	Mnemonic	Meaning
15	Not Used	This bit is not used and read as a zero.
14	MODID*	This bit is set to a one if the module is not selected with the MODID line on the VXI P2 connector. A zero indicates that the module is selected by the MODID signal.
13:4	Not Used	These bits are not used and read as ones.
3	RDY	Ready is a read-only bit that is set to a one indicating successful completion of register initialization.
2	PASS	Pass is a read-only bit that is set to a one when the V155 has completed its power-on self-test without any errors. If an error occurs, this bit is set to a zero and the SYSFAIL signal is asserted by the V155.
1	SYS INH	SYSFAIL INHIBIT. Reading this bit as a one indicates that the V155 is prevented from driving the backplane SYSFAIL line.
0	SFT RST	SFT RST This bit is read as a one when the V155 has been placed into the Soft Reset state. Writing this register with this bit set to a zero removes the V155 from the soft reset state.

For write operations executed to the Status/Control Register:

04h

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only				Not Used										SYS INH	SFT RST

Bit(s)	Mnemonic	Meaning
15:2	Not Used	These bits are not used for write operations.
1	SYS INH	SYSFAIL INHIBIT is a write/read bit used to inhibit the V155 from asserting the backplane signal SYSFAIL. Setting this bit to a one disables the assertion of SYSFAIL and a zero enables the signal.
0	SFT RST	SOFT RESET is a write/read bit used to reset the V155. Setting this bit to a one places the V155 in the soft reset state and writing the bit to a zero removes the V155 from the reset state.

MODID Register

The MODID Register is a read/write register located at an offset of 08_{16} from the A16 Logical Base Address of the V155. This register is only available when the V155 is configured as a Slot0 device. When the V155 is configured for non-Slot0 operation, the MODID Register is reserved.

The MODID Register is a write/read register used to control the MODID geographic addressing lines on the VXI P2 connector. Each of the 13 slots in the VXI chassis has an individual line that can be asserted and monitored through the MODID Register. Before any MODID lines can be asserted by the V155, the Output Enable bit (bit 13) of the register must be set to a one. When the outputs are enabled, setting a MODID bit location to a one asserts the associated MODID signal.

The data read from the MODID bits in this register do not necessarily reflect the state of the bits that were written to this register. Instead, a read of this register returns the actual state of each MODID line.

The following diagram shows the bit pattern for the MODID Register.

08h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Write	**1**	**1**	**MODID Enable**	**MID 12**	**MID 11**	**MID 10**	**MID 9**	**MID 8**	**MID 7**	**MID 6**	**MID 5**	**MID 4**	**MID 3**	**MID 2**	**MID 1**	**MID 0**

Bit(s)	Mnemonic	Meaning
15:14	Not Used	This bit is not used and read as a ones.
13	MODID Enable	OUTPUT ENABLE is a write/read bit used to enable or disable the V155 from driving the MODID signals. Setting this bit to a one enables the drivers and a zero disables them.
12:0	MID 12-0	MODULE ID 12 through 0 are write/read bits used to assert and monitor the MODID signals. Writing a bit with a one will assert (as indicated by a high state) the corresponding slot's MODID line.

Suffix High Register

The Suffix High Register is a read-only register located at an offset of 20_{16} from the A16 Logical Base Address of the V155. This register is used in combination with the Suffix Low Register to determine the module model number suffix. The Suffix High Register contains the first two ASCII characters of the suffix and the Suffix Low Register contains the last two characters. The suffix shown is for the V155-AA11 module.

The bit pattern for the Suffix High Register is as follows:

For Read transactions:

20h

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

Suffix Low Register

The Module Suffix Low Register is a read-only register located at an offset of 22_{16} from the A16 Logical Base Address of the V155. This register is used in combination with the Suffix High Register to determine the module model number suffix. The Suffix Low Register contains the last two ASCII characters of the suffix and the Suffix High Register contains the first two characters. The suffix shown is for the V155-AA11 module.

The bit pattern for the Suffix Low Register is as follows:

For Read transactions:

22h

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

Serial Number High Register

The Serial Number High Register is a read-only register located at an offset of 24_{16} from the A16 Logical Base Address of the V155. This register is used in conjunction with the Serial Number Low Register to define the serial number of the V155. The following diagram shows the bit pattern of the Serial Number High Register.

For Read transactions:

24h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Only	**SN 31**	**SN 30**	**SN 29**	**SN 28**	**SN 27**	**SN 26**	**SN 25**	**SN 24**	**SN 23**	**SN 22**	**SN 21**	**SN 20**	**SN 19**	**SN 18**	**SN 17**	**SN 16**

Serial Number Low Register

The Serial Number Low Register is a read-only register located at an offset of 26_{16} from the A16 Logical Base Address of the V155. This register is used in conjunction with the Serial Number High Register to define the serial number of the V155. The following diagram shows the bit pattern of the Serial Number Low Register.

For Read transactions:

26h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Only	**SN 15**	**SN 14**	**SN 13**	**SN 12**	**SN 11**	**SN 10**	**SN 9**	**SN 8**	**SN 7**	**SN 6**	**SN 5**	**SN 4**	**SN 3**	**SN 2**	**SN 1**	**SN 0**

Interrupt Status Register

The Interrupt Status Register is a read-only register located at an offset of $2A_{16}$ from the A16 Logical Base Address of the V155. The contents of this register are enabled onto the VME*bus* during an interrupt acknowledge cycle. This register contains the Logical Address of the V155 in the lower 8-bits of the register and the upper 8-bits contains the cause/status of the interrupt. The lower 8-bits of this register return the Logical Address of the V155 only for interrupt acknowledges cycles. An I/O read of this field returns all 8-bits set to ones.

The V155 has two interrupt sources. One of the sources is from a pre-selected VXI Trigger input and the other source is from Location Monitors. The VXI interrupt sources are enabled through the Trigger Interrupt Mask Register located at offset $2E_{16}$. The Location Monitor interrupt sources are enabled through the Location Monitor Interrupt Control Register located at an offset of $3A_{16}$. These two registers must be appropriately enabled before the V155 can generate an interrupt source. The interrupt source(s) may then generate a VXI interrupt request when interrupts are enabled in the Interrupt Control Register located at an offset of $2C_{16}$.

The interrupt acknowledge cycle executed by the Interrupt Handler reads a 16-bit value from the V155. The lower 8-bits of this data reflects the Logical Address of the device generating the interrupt. The upper 8-bits reflects the cause of the interrupt. Of the upper 8-bits, only 2 of them are used by the V155. Once an interrupt acknowledges cycle occurs, the interrupt source bits that were set in this register when the interrupt vector was read are reset to zero. This will also occur when the Interrupt Status Register is read.

The format of the Interrupt Status Register is as follows:

2Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Only	0	0	0	0	0	0	LOC MON	TRG IN	LA 128	LA 64	LA 32	LA 16	LA 8	LA 4	LA 2	LA 1

Bit(s)	Mnemonic	Meaning
15:10	Not Used	These bits are not used and read as zeros.
9	LOC MON	LOCATION MONITOR INTERRUPT SOURCE is a read-and-clear bit that is set when an interrupt source is generated by one of the Location Monitor Interrupts enabled in the Location Monitor Interrupt Control Register. To find out the actual cause of the location monitor interrupt, the Location Monitor Interrupt Control Register must be consulted.
8	TRG IN	TRIGGER IN INTERRUPT SOURCE is a read-and-clear bit that is set when an interrupt source is generated by one of the enabled trigger input interrupt sources in the Trigger Interrupt Mask Register. To find out the actual cause of the trigger input interrupt, the Trigger Interrupt Source Register must be consulted.
7:0	LA128:1	LOGICAL ADDRESS 128 through 1 return the Logical Address of the V155 during an interrupt acknowledge cycle to the V155. An I/O read of these bits return all ones.

Interrupt Control Register

The Interrupt Control Register is a write/read register located at an offset $2C_{16}$ from the A16 Logical Base Address of the V155. This register is used to configure the V155 for interrupt sourcing. The Interrupt Request Level, Interrupt Enable, and Interrupt Source Mask are contained in this register.

The format and description of the Interrupt Control Register are shown in the following diagram.

2Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Write	**1**	**1**	**1**	**1**	**1**	**1**	**LOC MON***	**TRG IN***	**IR ENA***	**1**	**IRQ S3**	**IRQ S2**	**IRQ S1**	**1**	**1**	**1**

Bit(s)	Mnemonic	Meaning
15:10	Not Used	These bits are not used and read as ones.
9	LOC MON*	LOCATION MONITOR INTERRUPT ENABLE is a write/read bit used to enable and disable the generation of a VXI interrupt when one of the Location Monitor interrupt sources are enabled in the Location Monitor Interrupt Control Register. Setting this bit to a one disables the Location Monitor interrupts and a zero enables the interrupt.
8	TRG IN*	TRIGGER IN INTERRUPT ENABLE is a write/read bit used to enable and disable the generation of a VXI interrupt when one of the enabled interrupt sources in the Trigger Interrupt Mask is generated. Setting this bit to a one disables the interrupts and a zero enables the interrupt.
7	IR ENA*	INTERRUPT REQUEST ENABLE is a write/read bit used to enable/disable the V155 from generating an interrupt request to the VME*bus*. Setting this bit to a one disables the V155 from generating an interrupt request and a zero enables the interrupt request.
6	Not Used	This bit is not used and read as a one.
5:3	IRQS3:1	INTERRUPT REQUEST SELECT 3 through 1 are write/read bits used to select the desired interrupt request level that the V155 asserts when an interrupt is sourced.

The following chart shows the interrupt request level selections.

IRQ S3	IRQ S2	IRQ S1	Interrupt Request Level
0	0	0	IRQ7
0	0	1	IRQ6
0	1	0	IRQ5
0	1	1	IRQ4
1	0	0	IRQ3
1	0	1	IRQ2
1	1	0	IRQ1
1	1	1	Disconnected

2:0 Not Used These bits are not used and read as ones.

Trigger Interrupt Mask/Trigger Interrupt Source Register

The Trigger Interrupt Mask/Trigger Interrupt Source Register is located at an offset of $2E_{16}$ from the A16 Logical Base Address of the V155. This register serves two purposes, depending on the direction of the transfer. A write operation to this register address accesses the Trigger Interrupt Mask Register. This register is used to enable and disable interrupts to the VXI bus on the occurrence of a trigger condition. Trigger conditions include the 8 VXI TTL Trigger lines and the two ECL VXI Trigger lines. A mask bit is set to a one to enable the interrupt source and set to a zero to disable the source.

The second register at this address is the Trigger Interrupt Source Register. This read-only register is used to determine which trigger event caused the interrupt source. Each bit read as a one was involved in generating the trigger interrupt source. After an interrupt has been generated and acknowledged, the Trigger Interrupt Source Clear Register must be written with data to clear the individual interrupt source.

The following two diagrams show the two registers.

Trigger Interrupt Mask Register (Write-Only):

2Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	**Not Used**						**ECL TG1**	**ECL TG0**	**TTL TG7**	**TTL TG6**	**TTL TG5**	**TTL TG4**	**TTL TG3**	**TTL TG2**	**TTL TG1**	**TTL TG0**

Bit(s)	Mnemonic	Meaning
15:10	Not Used	These bits are not used and setting them to ones does not have any effect on the V155.
9:8	ECL TG1:0	ECL TRIGGER 1 and 0 are write-only bits used to enable the generation of a VXI interrupt when the corresponding VXI ECL Trigger line is asserted. A bit set to a one enables the interrupt source and a zero disables the interrupt source.
7:0	TTL TG1:0	TTL TRIGGER 7 and 0 are write-only bits used to enable the generation of a VXI interrupt when the corresponding VXI TTL Trigger line is asserted. A bit set to a one enables the interrupt source and a zero disables the interrupt source.

Trigger Interrupt Source Register (Read-only):

2Eh

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Only	Not Used						ECL TG1	ECL TG0	TTL TG7	TTL TG6	TTL TG5	TTL TG4	TTL TG3	TTL TG2	TTL TG1	TTL TG0

Bit(s)	Mnemonic	Meaning
15:10	Not Used	These bits are not used and read as zeros.
9:8	ECL TG1:0	ECL TRIGGER INTERRUPT 1 and 0 are read-only bits that are read as a one when the V155 has received the assertion of the corresponding VXI ECL Trigger line and the Interrupt Mask bit was enabled. Reading this bit as a zero indicates that the ECL Trigger line is not generating an interrupt source.
7:0	TTL TG7:0	TTL TRIGGER INTERRUPT 7 through 0 are read-only bits that are read as a one when the V155 has received the assertion of the corresponding VXI TTL Trigger line and the Interrupt Mask bit was enabled. Reading this bit as a zero indicates that the TTL Trigger line is not generating an interrupt source.

Trigger Interrupt Source Clear Register

The Trigger Interrupt Source Clear Register is a write-only register located at an offset of 30_{16} from the A16 Logical base Address of the V155. This register is used to clear the Interrupt Source bits in the Trigger Interrupt Source Register once they have been set by the receipt of a preselected trigger input. Any bit location set to a one when writing to this register clears the corresponding Interrupt Source bit. Any bit set to a zero has not effect on the Interrupt Source.

The following diagram shows the bit layout for the Trigger Interrupt Source Clear Register.

30h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	**Not Used**						**ECL TG1**	**ECL TG0**	**TTL TG7**	**TTL TG6**	**TTL TG5**	**TTL TG4**	**TTL TG3**	**TTL TG2**	**TTL TG1**	**TTL TG0**

Bit(s)	**Mnemonic**	**Meaning**
15:10	Not Used	These bits are not used. Any write to these bit locations have no effect on the V155.
9:8	ECL TG1:0	CLEAR ECL TRIGGER INTERRUPT SOURCE 1 and 0 are write-only bits used to clear the corresponding ECL trigger interrupt source once set by the assertion of the signal.
7:0	TTL TG7:0	CLEAR TTL TRIGGER INTERRUPT SOURCE 7 through 0 are write-only bits used to clear the corresponding TTL trigger interrupt source once set by the assertion of the signal.

Trigger Source Register

The Trigger Source Register is a write-only register located at an offset of 32_{16} from the A16 Logical Base Address of the V155. This register is used to source the VXI ECL and VXI TTL. This register allows the trigger signals to be either asserted, negated or pulsed. The binary combination of bits 15 and 14 of this register determine what action is to be taken on the selected trigger signals. The following chart shows the binary combination of the control bits and the effect they have on the selected trigger signals.

CNTL1	CNTL0	Effect On Trigger Signal
0	0	Assertion
0	1	Negation
1	0	Pulse
1	1	Reserved

When a trigger is asserted through the Trigger Source Register, it remains asserted until either a reset condition occurs or the Trigger Source Register is written to negate the trigger signal. A pulsed output lasts for approximately 1.5 microseconds.

The following diagram shows the bit pattern for the Trigger Source Register.

32h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	CNTL1	CNTL0	Not Used				ECL TG0	ECL TG0	TTL TG7	TTL TG6	TTL TG5	TTL TG4	TTL TG3	TTL TG2	TTL TG1	TTL TG0

Bit(s)	Mnemonic	Meaning
15:14	CNTL1:0	CONTROL 1 and 0 are write-only bits used to define the operation to be performed on the requested trigger signal. The binary combination of these bits determine what action to take on the selected trigger signals. The previous chart shows the required binary combinations to set, clear and pulse the trigger signals.
13:10	Not Used	These bits are not used and may be written with any data pattern.
9:8	ECL TG1:0	ECL TRIGGER 1 and 0 are set to a one when writing to this register to allow the selected operation specified by the control bits to occur to the VXI ECL Trigger lines. Any trigger bit set to a zero when writing to this register has no effect on the trigger signal itself.
7:0	TTL TG7:0	TTL TRIGGER 7 through 0 are set to a one when writing to this register to allow the selected operation specified by the control bits to occur to the VXI TTL Trigger lines. Any trigger bit set to a zero when writing to this register has no effect on the trigger signal itself.

Trigger Timer Configuration Register

The Trigger Timer Configuration Register is a write-only register located at an offset of 34_{16} from the A16 Logical Base Address of the V155. This register is used to configure the timer interval and specify the trigger signals to assert once the Trigger Timer expires. The Trigger Timer, which is a 32-bit modulo-n type counter, can be tied to any or all of the trigger signals. At a predetermined interval, the enabled trigger signals are pulsed for a period of approximately 1.5 microseconds.

The actual register accessed through this A16 address offset is determined by the four most significant bits of the Miscellaneous Control Register at offset $3C_{16}$. The binary combination of these four bits specify the register to be accessed as shown in the following table.

RSEL3	RSEL2	RSEL1	RSEL0	Register Accessed
0	0	0	0	Trigger Timer Low
0	0	0	1	Trigger Timer High
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	0	0	0	Trigger Timer Control
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

A Trigger Timer is configured by first loading the Trigger Timer High Register and Trigger Timer Low Register. The Trigger Timer Low Register is used in conjunction with the Trigger Timer High Register for establishing the timer interval. This 32-bit counter is programmable from 2 microseconds to 429 seconds in 100 nanosecond increments. The data value loaded into the combination of the Trigger Timer Low and High Registers is the number of 100 nanosecond increments between trigger assertions. For example, to obtain an interval of 1 millisecond, the 32-bit timer must be loaded with data set to 10000 (2710_{16}). Therefore, the Trigger Timer High Register is loaded with 0 and the Trigger Timer Low Register is loaded with 10000 (2710_{16}).

The following diagram shows the bit pattern for the Trigger Timer High Register.

34h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	TMR 31	TMR 30	TMR 29	TMR 28	TMR 27	TMR 26	TMR 25	TMR 24	TMR 23	TMR 22	TMR 21	TMR 20	TMR 19	TMR 18	TMR 17	TMR 16

Bit(s)	Mnemonic	Meaning
15:0	TMR31:16	TIMER DATA 31 through 16 are write-only bits used to establish the interval at which trigger signals are asserted. This register is used in combination with the Trigger Timer Low Register to determine the number of 100 nanosecond increments between trigger assertions.

The following diagram shows the bit pattern for the Trigger Timer Low Register.

34h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	TMR 15	TMR 14	TMR 13	TMR 12	TMR 11	TMR 10	TMR 9	TMR 8	TMR 7	TMR 6	TMR 5	TMR 4	TMR 3	TMR 2	TMR 1	TMR 0

Bit(s)	Mnemonic	Meaning
15:0	TMR15:0	TIMER DATA 15 through 0 are write-only bits used to establish the interval at which trigger signals are asserted. This register is used in combination with the Trigger Timer High Register to determine the number of 100 nanoseconds increments between trigger assertions.

The Trigger Timer Control Register contains an enable bit that allows the timer to operate. This register also contains the 10 trigger source bits which determine the trigger signals to assert once the timer expires. Any trigger signal bit set to a one in this register is asserted once the timer expires. The following diagram shows the bit layout for the Trigger Timer Register.

34h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	**TMR ENA**	Not Used					**ECL TG1**	**ECL TG0**	**TTL TG7**	**TTL TG6**	**TTL TG5**	**TTL TG4**	**TTL TG3**	**TTL TG2**	**TTL TG1**	**TTL TG0**

Bit(s)	**Mnemonic**	**Meaning**
15	TMR ENA	TIMER ENABLE is a write-only bit used to enable and disable the timer from operating. Setting this bit to a one enables the timer and a zero disables the timer.
14:10	Not Used	These bits are not used and may be written with any data pattern.
9:8	ECL TG1:0	VXI ECL TRIGGER1 and 0 are write-only bits used to enable the assertion of the corresponding VXI ECL Trigger signal once the timer expires. A zero in a bit location prevents the signal from being asserted once the timer expires.
7:0	TTL TG7:0	VXI TTL TRIGGER7 through 0 are write-only bits used to enable the assertion of the corresponding VXI TTL Trigger signal once the timer expires. A zero in a bit location prevents the signal from being asserted once the timer expires.

SBC Slave Mode Enable Register

The SBC (Single Board Computer) Slave Mode Enable Register is a write-only register located at an offset of 38_{16} from the A16 Logical Base Address of the V155. This register is used to enable accesses to the SBC slave mode functions. Please refer to the SBC manual for additional information on enabling these accesses on the SBC.

The V155's SBC can respond as a slave on the VXI*bus*. The V155 can respond to 255 megabyte block of extended space addresses (A32 address space). The RAM on the SBC can be accessed in both standard and extended address space.

This register provides access to setup and enable the VXI*bus* transfers to the SBC as a slave. Under VXI, there are 3 possible address spaces. The binary combination of the most significant two bits of this register are used to select which address space is written. Since the V155 only supports slave operations in A32 space, only 1 setting is possible. The following chart shows the address configuration selection.

CNTL1	CNTL0	Address Configuration Register
0	0	A32 Address
All Others		Reserved

The following diagram shows the bit patterns for the Address Configuration Register. The two most significant bits must be set as shown. The ENABLE bit enables or disables the specified slave address space. Setting the bit to a one enables the address space and disabled by setting this bit to a zero. The address bit specifications represents the VME address bits that are used during the compare to determine the address match in the A32 address space.

The following diagram shows the bit pattern for the Address Configuration Register.

38h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	**0**	**0**	**Not Used**					**ENA**	**A31**	**A30**	**A29**	**A28**	**A27**	**A26**	**A25**	**A24**

Interrupt Status ID Register

The Interrupt Status ID Register is a read-only register located at an offset of 38_{16} from the A16 Logical Base Address of the V155. This register is used to read the 16-bits of data received from the V155 during an interrupt acknowledge cycle on the VXI*bus*. Since the SBC only supports an 8-bit interrupt vector, an external mechanism has been provided to latch the entire 16-bits of interrupt vector information. The following diagram shows the bit pattern for the Interrupt Status ID Register.

38h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-Only	ISRC 7	ISRC 6	ISRC 5	ISRC 4	ISRC 3	ISRC 2	ISRC 1	ISRC 0	LA 128	LA 64	LA 32	LA 16	LS 8	LA 4	LA 2	LA 1

Bit(s)	Mnemonic	Meaning
15:8	ISRC7:0	INTERRUPT SOURCE 7 through 0 are read-only bits which reflect the interrupt source bits set by the interrupting VXI module during the interrupt acknowledge cycle.
7:0	LA128:1	LOGICAL ADDRESS 128 through 1 are read-only bits used to determine the Logical Address of the interrupting VXI module.

Miscellaneous Control Register

The Miscellaneous Control Register is a write-only register located at an offset of $3C_{16}$ from the A16 Logical Base Address of the V155. This register is used to control which buried register is accessed through the Trigger Timer Configuration Register address. The following diagram shows the bit pattern for the Miscellaneous Control Register.

3Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-Only	RESL 3	RESL 2	RESL 1	RESL 0	Not Used											

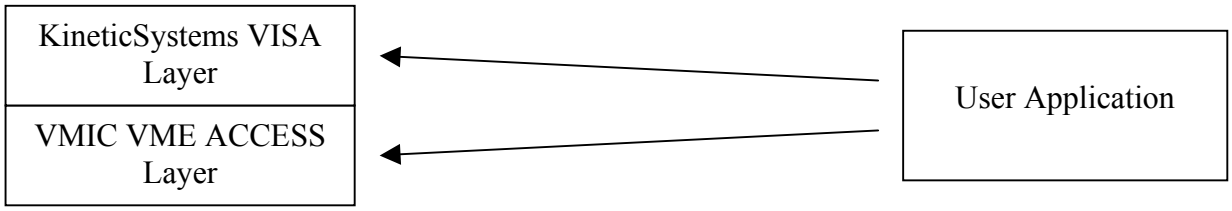
Bit(s)	Mnemonic	Meaning
15:12	RSEL3:0	REGISTER SELECT3 through 0 are write-only bits used to specify which buried register is to be accessed when writing to the Trigger Timer Configuration Register as shown in the table below.
11:0	Not Used	These bits must be set to a zero when writing to this register.

RSEL3	RSEL2	RSEL1	RSEL0	Register Accessed
0	0	0	0	Trigger Timer Low
0	0	0	1	Trigger Timer High
0	0	1	0	Reserved
0	0	1	1	Reserved
0	1	0	0	Reserved
0	1	0	1	Reserved
0	1	1	0	Reserved
0	1	1	1	Reserved
1	0	0	0	Trigger Timer Control
1	0	0	1	Reserved
1	0	1	0	Reserved
1	0	1	1	Reserved
1	1	0	0	Reserved
1	1	0	1	Reserved
1	1	1	0	Reserved
1	1	1	1	Reserved

Chapter 4: Programming

Software Layers

A user application can access either the VXI or VME layer software. Each layer is tailored to a different level of access, so an application designer has the freedom to choose whatever layer is required.



User application can link/dynamically load to either software layer to access the services required.

VMIC VME Layer

The VMEBus software comes with its own documentation and help files. Please consult this documentation for more details.

KineticSystems VISA Layer

VISA provides the highest level of device type/platform independent access. Please consult the *KinteicSystems VISA Application Programming Interface* document that accompanied the VISA distribution for details on access provided by the VISA layer.

Special Note: Because of limitations of the VMIC library, block moves via the VISA layer (e.g., `viMoveIn32()`, `viMoveOut16()` etc.) can achieve optimal performance by acquiring DMA buffers, as detailed below. Without accessing through an appropriately acquired DMA buffer, block moves under VISA will be transformed by the VMIC layer to a sequence of single moves.

A DMA buffer is acquired by opening a session to the controller memory resource ("VXI::MEMACC"), and calling viMemAlloc(). The pointer returned will be a pointer to memory on the local machine, allocated from the DMA buffer heap. Data should be placed in this buffer before a move out and read from here on a read in. In addition, the session of the instrument doing the move must have its VI_ATTR_DMA_ALLOW_EN attribute set to VI_TRUE; the underlying hardware will not perform a true DMA operation if this attribute is not set. Note that if VI_ATTR_DMA_ALLOW_EN is VI_TRUE and a VISA move operation is called with a buffer *not* allocated from the controller memory resource, the operation will fail.

Example

The following code illustrates the use of acquiring and releasing a DMA buffer.

```
/*
 * This code fragment will open an instrument, copy a 1K block of single word data from
 * offset 0x1000 of its extended memory region, modify it, and write it back out.
 */

ViInt16 *pAddr;
ViSession iInstrHandle,memHandle;

/*
 * Open the instrument at logical address 5
 */
viOpen(rmSesn, "VXI::5::INSTR",VI_NULL,VI_NULL,&iInstrHandle);

/*
 * Open the controller memory resource
 */
viOpen(rmSesn, "VXI::MEMACC",VI_NULL,VI_NULL,&memHandle);

/*
 * Allocate a DMA buffer
 */
viMemAlloc(memHandle,1024 * sizeof(short),&pAddr);

/*
 * Set DMA attribute to allow DMA operations
 */
viSetAttribute(iInstrHandle, VI_ATTR_DMA_ALLOW_EN,VI_TRUE);

viMoveIn16(iInstrHandle,VI_A32_SPACE,
           0x1000,2048,pAddr);

/* code here to update/modify data in pAddr */

viMoveOut16(iInstrHandle,VI_A32_SPACE,
            0x1000,2048,pAddr);

/*
 * Free the buffer
 */
viMemFree(memHandle,pAddr);

/*
 * Close memory resource
 */
viClose(memHandle);
```

Building an application

Building an application will require including the appropriate header file(s) and linking to the appropriate lib file(s), as shown in the following table:

Software Layer	Header to include	Lib file to link to
KineticSystems VISA	visa.h	visa32.lib
VMIC Layer	vmemgr.h	vmemgr.lib

Chapter 5: VXI Triggers

The V155 supports the eight VXI TTL trigger lines as well as the two VXI ECL trigger lines. These signals operate in the same manner as the VXI trigger lines. The V155 hardware supports the Stop/Start protocol and the Synchronous protocol for asserting the VXI trigger lines.

The Start/Stop protocol provides a mechanism to assert a trigger line under computer control and maintain the signal level until the trigger line is negated by programmed control. This can be useful for generating a trigger signal completed under computer control. The Synchronous protocol permits the V155 to generate a pulse on a trigger line for a duration of approximately 1.5 microseconds. The computer merely writes to the Trigger Source Register with the trigger signals to be asserted.

Synchronous Trigger Example:

As an example, assume it is desired to apply a 1.5 microsecond pulse to VXI trigger line 2. The V155 is set for Logical Address 0, which results in a A16 Logical Base Address of $C000_{16}$. This can be accomplished by writing to the Trigger Source Register, at an offset of 32_{16} from the A16 Logical Base Address of the V155, with the data set to 8804_{16}.b

Shown in pseudocode, the trigger may be sourced as follows:

```
/* 16-bit A16 write to address 0xc032 with data of 0x8004 */

viOut16(ViSession, VI_A16_SPACE, 0x32, 0x8004);
```

KineticSystems Visa also directly supports this operation via viAssertTrigger():

```
viSetAttribute(session,VI_ATTR_TRIG_ID,VI_TRIG_TTL2);
viAssertTrigger(session,VI_TRIG_PROT_SYNC);
```

Start/Stop Trigger Example:

As an example, assume it is desired to apply assert VXI trigger line 5 and the ECL trigger line 0, wait for a period of time, negate ECL trigger 0, wait for a period of time, and then negate VXI trigger line 5. The V155 is set for Logical Address 0, which results in a A16 Logical Base Address of $C000_{16}$.

Shown in pseudocode, the trigger sequence can be sourced as follows:

```
/*
 * assert TTL trigger line 5 and ECL trigger line 0
```

```
 */

viOut16(ViSession, VI_A16_SPACE, 0x32, 0x120);

/*
 *delay for a period of time
 */
taskDelay ( 1 );

/*
 * negate ECL trigger line 0
 */
viOut16(ViSession, VI_A16_SPACE, 0x32, 0x4100);

/*
 * delay for a period of time
 */
taskDelay ( 1 );

/*
 * negate TTL trigger line 5
 */
viOut16(ViSession, VI_A16_SPACE, 0x32, 0x4020);
```

In Visa, the same operation:

```
/*
 * Turn on trigger ECL0
 */
viSetAttribute(session,VI_ATTR_TRIG_ID,VI_TRIG_ECL0);
viAssertTrigger(session,VI_TRIG_PROT_ON);

/*
 * Turn on trigger TTL5
 */
viSetAttribute(session,VI_ATTR_TRIG_ID,VI_TRIG_TTL5);
viAssertTrigger(session,VI_TRIG_PROT_ON);

/*
 * Delay for a period of time
 */
taskDelay ( 1 );

/*
 * negate ECL trigger line 0
 */
viSetAttribute(session,VI_ATTR_TRIG_ID,VI_TRIG_ECL0);
viAssertTrigger(session,VI_TRIG_PROT_OFF);

/*
```

```
 * Delay for a period of time
 */
taskDelay ( 1 );

/*
 * negate TTL trigger line 5
 */
viSetAttribute(session,VI_ATTR_TRIG_ID,VI_TRIG_TTL5);
viAssertTrigger(session,VI_TRIG_PROT_OFF);
```

The trigger lines may also be connected to a hardware timer to assert them at a predetermined interval. The interval counter (timer) is based off of the CLK10 10 megahertz VXI*bus* clock and contains 32 bits. When the timer expires, a 1.5 microsecond pulse is applied to the preselected trigger line(s). The 32-bit counter yields an interval from 2 microseconds to 429 seconds in 100 nanosecond increments. The 32-bit timer value is split into two 16-bit values that are loaded into the Trigger Timer High and Trigger Timer Low Registers. Please refer to the Trigger Timer Registers for additional information on timer operation.

As an example, assume it is desired to setup VXI TTL trigger line 4 to be pulsed every 1 millisecond. For this example, the Trigger Timer High Register must be loaded with 0, the Trigger Timer Low Register must be loaded with 2710_{16}, and the Trigger Timer Control is loaded with 8010_{16}. The V155 is set for Logical Address 0, which results in a A16 Logical Base Address of $C000_{16}$.

Shown in pseudocode, the trigger sequence can be setup as follows:

```
/*
 * set the register select bits to zero in the misc. control register
 */
viOut16(ViSession, VI_A16_SPACE, 0x3C, 0x00);

/*
 * load the timer data into the timer data low register
 */
viOut16(ViSession, VI_A16_SPACE, 0x34, 0x2710);

/*
 * set register select bits to point to timer data high register
 */
viOut16(ViSession, VI_A16_SPACE, 0x3C, 0x1000);

/*
 * load the timer data into the time data high
 */
viOut16(ViSession, VI_A16_SPACE, 0x34, 0x00);

/*
 * set register select bits to point to timer control register
```

```
 */
viOut16(ViSession, VI_A16_SPACE, 0x3C, 0x8000);

/*
 * load enable timer and TTL line 4
 */
viOut16(ViSession, VI_A16_SPACE, 0x34, 0x8010);
```

After the timer is setup and enabled, a 1.5 microsecond pulse is generated on VXI TTL trigger line 4 every 1 millisecond. To stop the timer, the Timer Enable bit in the Timer Control Register must be set to zero.

Along with the ability to assert VXI/Front Panel trigger lines, the V155 can also respond to the assertion of these signals asserted by other devices. The V155 can respond to these signals by either polling or by an interrupt. Once an enabled trigger source is received by the V155, it is latched and 'held' until cleared by programmed control. To enable a specific trigger line source to be received by the V155, it must first be enabled in the Trigger Interrupt Mask Register located at an offset of $2E_{16}$ from the A16 Logical Base Address of the V155. This register contains individual bit positions for each of the 10 trigger sources. The trigger sources include the eight VXI TTL trigger lines, two VXI ECL trigger lines. Each bit set to a one enables the trigger source to be latched by the V155. Once an enabled trigger source has been latched by the V155, it may be read through the Trigger Interrupt Source Register, located at an offset of $2E_{16}$ from the A16 Logical Base Address. Any bit set to a one in this register may generate an interrupt request, if enabled. An interrupt source is any event that may generate an interrupt, if it is enabled in the Interrupt Control Register. The TRIGGER IN INTERRUPT ENABLE, the INTERRUPT REQUEST ENABLE and the INTERRUPT REQUEST SELECT bits must be set appropriately in order for an interrupt to be generated on the VXI*bus*. Please refer to the Interrupt Control Register section of this manual for additional information.

Once a trigger event has been latched and read through the Trigger Interrupt Source Register, it must be cleared before subsequent trigger events may be seen on that trigger line. The latched trigger source is cleared through the Trigger Interrupt Source Clear Register located at an offset of 30_{16} from the A16 Logical Base Address. Any bit set to a one when the register is written causes the corresponding trigger source to be reset to zero and ready for additional captures. This same routine must be followed regardless of the mechanism used to determine that a trigger event occurred.

As an example, assume it is desired to respond to the assertion of VXI trigger line 0 by asserting trigger line 1. This can be accomplished by setting up the V155 to enable VXI trigger line 0 in the Trigger Interrupt Mask Register and waiting for the source to be set in the Trigger Interrupt Source Register. This routine is using the polling technique instead of an interrupt driven mechanism. For this example, the V155 is set for Logical Address 0, which results in a A16 Logical Base Address of $C000_{16}$.

The pseudocode for this example is as follows:

```
/*
 * load interrupt mask register to enable VXI TTL trigger 0
 */
viOut16(ViSession, VI_A16_SPACE, 0x2E, 0x01);
```

```
/*
 * loop until trigger received (read of non zero)
 */
do
{
  /*
   * read the trigger interrupt source register
   */
  viIn16(ViSession, VI_A16_SPACE, 0x2E, &rdata );


} while (rdata == 0);

/*
 * write trigger interrupt source register to clear TTL trigger
 * 0 bit.
 */
viOut16(ViSession, VI_A16_SPACE, 0x30, 0x01);

/*
 * write trigger source register to pulse TTL trigger line 1
 */
viOut16(ViSession, VI_A16_SPACE, 0x32, 0x8002);
```

Appendix A

Technical Support and Warranty

DynamicSignals warrants its standard hardware products to be free of defects in workmanship and materials for a period of one year from the date of shipment to the original end user. DynamicSignals warrants its software products to conform to the software description applicable at the time of purchase for a period of ninety days from the date of shipment. Products purchased for resale by DynamicSignals carry the original equipment manufacturer's warranty.

DynamicSignals will, at its option, either repair or replace products that prove to be defective in materials or workmanship during the warranty period.

Transportation charges for shipping products to DynamicSignals are prepaid by the purchaser, while charges for returning the repaired product to the purchaser, if located in the United States, are paid by DynamicSignals. Return shipments are made by UPS, where available, unless the purchaser requests a premium method of shipment at his expense. The selected carrier is not the agent of DynamicSignals, and DynamicSignals assumes no liability relating to the services provided by the carrier.

The product warranty may vary outside the United States and does not include shipping, customs clearance or any other charges. Consult your local authorized representative for more information regarding specific warranty coverage and shipping details.

Product specifications and descriptions in this document subject to change without notice.
DynamicSignals specifically makes no warranty of fitness for a particular purpose or any other warranty either expressed or implied, except as is expressly set forth herein. This warranty does not cover product failures created by unauthorized modifications, product misuse or improper installation.

Products are not accepted for credit or exchange without prior written approval. If it is necessary to return a product for repair replacement or exchange, a Return Authorization (RA) Number must first be obtained from the Repair Service Center before shipping the product to DynamicSignals.

Please take the following steps if you are having a problem and feel you may need to return a product for service:

Contact DynamicSignals and discuss the problem with a Technical Service Engineer.
Obtain a Return Authorization (RA) Number.
Initiate a purchase order for the estimated repair charge if the product is out of warranty.
Include with the product a description of the problem and the name of the technical contact person at your facility.
Ship the product prepaid with the RA Number marked on the outside of the package to:

DynamicSignals LLC
Repair Service Center
900 North State Street
Lockport, IL 60441

Telephone: (815) 838-0005
Fax: (815) 838-4424

Feedback

The purpose of this manual is to provide you with the information you need to make the V155 as easy as possible to understand and use. It is very important that the information is accurate, understandable and accessible. To help us continue to make this manual as "user friendly" as possible, we hope you will fill out this form and Fax it back to us at (815) 838-4424. Or mail a copy to: DynamicSignals LLC 900 N. State St., Lockport, IL 60441. Your input is very valuable.

Please rate each of the following.

The information in this manual is:

	Yes									No
Accurate	10	9	8	7	6	5	4	3	2	1
Readable	10	9	8	7	6	5	4	3	2	1
Easy to find	10	9	8	7	6	5	4	3	2	1
Well organized	10	9	8	7	6	5	4	3	2	1
Sufficient	10	9	8	7	6	5	4	3	2	1

We would appreciate receiving any thoughts you have about how we can improve this user's manual:

(Include additional sheets if needed)

Name Phone Company