Model V207

16-bit, High-speed ADC Subsystem

User's Manual

March 6, 2003

This manual describes operation of current versions of the V207:

- **V207-ZA13 –** Basic V207 unit: 16-bit, 500 ksamples/second ADC with Ping-Pong output buffer and 16-bit data transfer capability.
- **V207-ZB13 –** Basic unit plus 1 Mbyte linear Multi-buffer with transient capture and 32-bit data transfer capability.
- **V207-ZB23 –** Basic unit plus 4 Mbyte linear Multi-buffer with transient capture and 32-bit data transfer capability.
- **V207-ZC13 –** Basic unit plus Digi-bus interface.
- **V207-ZD23 –** Basic unit plus 4 Mbyte circular Multi-buffer with transient capture and 32-bit data transfer capability.
- **V207-ZD33 –** Basic unit plus 16 Mbyte circular Multi-buffer with transient capture and 32-bit data transfer capability.

Appendix A describes operational differences between the V207-w**B**yz and V207-w**D**yz units. **Appendix B** lists differences between earlier versions and current versions of some of the above models. The primary difference between version-2 (V207-...2) and version-3 (V207-...3) models is that the version-2 units only support 16-bit data transfer (**D16**) capability. Version-3 units support both **D16** and **D32** data tranfer.

Table of Contents

Table of Figures

Table of Tables

About This Manual

Organization

Chapter 1*,* ***Introduction***, gives you a brief overview of the Model V207, lists items you need to get started, and explains how to safely unpack your module.

Chapter 2*,* ***Installation and Configuration***, explains how to configure the V207 and correctly insert it into a C-size VXIbus mainframe.

Chapter 3*,* ***Understanding the V207/V208 Family***, provides information about the V207, associated signal-conditioning modules, and interconnection methods.

Chapter 4, ***Understanding the V207***, describes the performance of the V207.

Chapter 5, ***Configuration and Operational Registers****,* explains how to access and control the V207.

Chapter 6, ***Programming Information****,* gives you example setup procedures for preparing the V207 to acquire data.

The ***Appendices*** provide additional information that may be helpful in the use of the Model V207, in learning more about KineticSystems and its products, and in quickly reaching us.

Glossary

Following is a glossary of some of the terms and conventions used throughout this manual:

*****	An indicator that a register bit contains low-true data. For example, writing a **"0"** to a bit labeled Enable* would cause a function to be enabled.
A16 Space	The first 64 kbytes of address space, accessible with 16-bit addressing. The configuration registers of VXI devices occupy 64-byte blocks of this address space. The Logical Address of a device determines which 64-block block is associated with that device.
A32 Space	The 4 Gbyte address space, accessible with 32-bit addressing. A module can request a block of this address space via information contained in its *Configuration* registers. *Operational* registers, if present, reside in this space.
ADC Clock	The rate at which data is presented to the ADC for digitization. The ADC clock rate is also equal to the maximum rate of aggregate data throughput for all digitized channels.
Configuration Registers	Setup registers located in **A16** space. Some are mandatory; some are optional.
D16	A single 16-bit data transfer.

D16 BLK A block transfer of 16-bit words.

D32 A single 32-bit data transfer. Not all Slot-0 controllers support **D32**.

D32 BLK A block transfer of 32-bit words. Not all Slot-0 controllers support **D32** BLK.

Device One of 255 devices that a VXIbus system can support. The term is often used interchangeably with "module." The distinction is that a VXIbus module can consist of more than one device.

Digi-bus™ A high-speed digital bus protocol developed by KineticSystems and implemented on the VXI Local Bus. The **Digi-bus** protocol allows an ADC to transmit digital data to DSP and memory devices at high speeds without external wiring.

DSP Digital Signal Processor. Usually refers to VXIbus devices that perform autonomous digital signal processing of digital data.

Dynamic Addressing The VXIbus addressing mode in which the address of a device is stored in a write-able register. See also Static Addressing.

hexadecimal A base-16 number. The suffix, "h," indicates that a number is hexadecimal. For example, 1Ah = 26_{10}; FFh = 255_{10}; 1000h = 4096_{10} .

Local Bus A user-definable 12-line bus implemented on the P2 connectors of the VXIbus.

Logical Address A VXIbus module's unique address. A VXIbus system has 254 logical addresses that are available. **"0"** is the address of the Slot-0 controller. **"255"** specifies that dynamic addressing be used to address that module.

MUX-bus™ A four-channel analog bus protocol developed by Kinetic Systems and implemented on the VXI Local Bus. The **MUX-bus** protocol allows signal-conditioning modules to transmit analog data to an ADC module at high speeds with high accuracy without external wiring.

Operational Registers Setup and data-transfer registers that are located in A32 address space.

RAM Random Access Memory. RAM refers to a memory block that has direct addressable access, as opposed to sequential access.

Resource Manager Software that sets logical addresses and optimally configures Operational register addresses and memory-block addresses in a system. The manufacturer of the Slot-0 controller provides this software, often referred to as "RESMAN."

Sample Clock or Scan Clock The rate at which each channel is digitized. Each tick of the sample clock initiates digitization of the channels in the order defined in Scan RAM. Digitization occurs at the rate of the ADC clock. Note that, in this context, "sample" and "scan" are interchangeable, as in "sample clock" and "scan clock" or as in "sample rate" and "scan rate."

Static The VXIbus addressing mode in which the address of a device is stored in a

Addressing switch register. See also Dynamic Addressing.

Chapter 1: Introduction

About the V207

The V207 is a single-width, C-size, register-based VXIbus module that provides high-speed, high-resolution, analog-to-digital conversion. This module is intended for use with VXIbus signal conditioning modules that incorporate the analog **MUX-bus** (models V241, V246, V252 and V253) to provide flexible system configurations without the need for inter-module cabling. The 96-channel MUX-bus compatible V243 is intended for use with the V208. It will operate with the high-speed V207, but with reduced accuracy.

The V207 provides four instrumentation amplifier inputs to increase throughput with large numbers of input channels. When used with appropriate signal-conditioning modules, up to 256 channels can be digitized. If no **MUX-bus** signal-conditioning modules are used, the V207 can digitize up to four preconditioned signals that connect to front-panel isolated BNC connectors.

The V207 provides a high degree of flexibility for configuring analog sampling systems. With the optional Multi-buffer memory, data can fill the buffer at high sample rates and then be emptied in bursts over the VXIbus. Multi-buffers can be operated in *Continuou*s or *Transient* modes. If the **Digi-bus** option is used, the digital data can stream to the V165 Digital Signal Processor, the V110 RAM memory module or to other modules supporting **Digi-bus** protocol.

The V207 supports both static and dynamic configuration. Access to digitized data and to operational registers is via **A32** memory space. Data transfers use **D16** and **D32** protocol.

The V207 is available with several digital-output options and Multi-buffer size options. The part number is V207-*wxyz*, where:

w specifies the sampling rate:

- Z = 500 ksamples/s

x selects the digital output option:

- A = Ping-Pong buffer
- B = Linear Multi-buffer with transient capture option without pre-trigger capability
- C = **Digi-bus**
- D = Circular Multi-buffer with transient capture option with pre-trigger capability

y specifies the Multi-buffer size:

- 1 = 1 Mbyte Multi-buffer
- 2 = 4 Mbyte Multi-buffer
- 3 = 16 Mbyte Multi-buffer

z specifies the version number.

Version 3 of a V207 with sampling rate to 500 ksamples/s and 16 Mbyte Multi-buffer with transient capture pre-trigger capability is specified as V207-ZD33, for example.

Getting Started

To set up and use your V207 VXIbus module, you will need most or all of the following:

- The V207, configured with any appropriate options, and its User Manual
- One or more of the following **MUX-bus**-compatible signal-conditioning modules and their User Manuals:
  - V241 - 96-channel, high-level, scanning MUX
  - V246 - 8-channel bridge signal conditioner
  - V252 - 8 or 16-channel, 8-pole analog filter
  - V253 - 16-channel, programmable gain / analog filter
- One or more of the following software packages:
  - VXI*plug&play* instrument driver(s)
  - VISA software associated with a computer interface or Slot-0 controller that supports the VXI *plug&play* standards.
  - LabVIEW for Windows
  - LabWindows/CVI
  - MS Visual Basic
  - KSC *Reality* software
- "VXI Data Acquisition Handbook" by Dr. J. W. Tippie
- Your VXIbus system with its Resource Manager and high-level test and/or application software

Unpacking the V207

The V207 comes in an anti-static bag to avoid electrostatic damage. Electrostatic discharge to the module can damage components on it. Please take the following precautions when unpacking the module:

- Ground yourself with a grounding strap or by touching a grounded object.
- Touch the anti-static package to a metal part of your VXIbus 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 VXIbus chassis until you are satisfied that the module exhibits no obvious mechanical damage and is configured to conform to the desiring operating environment. The next chapter describes installation and configuration.

Chapter 2: Installation and Configuration

Setting the Logical Address Switches

A VXI system can have as many as 255 devices, with each having a unique number in the range from zero to 254. Eight bits represent the number, which is the Logical Address of the device. VXIbus defines two concepts of addressing: "static" and "dynamic." All VXIbus devices *must* allow static addressing, in which the address is determined by the setting of a switch register. VXIbus devices may, but are not required to, support dynamic addressing. In dynamic addressing, the Logical Address is stored in a software-addressable register. For reasons discussed in Chapter 5, all KineticSystems VXIbus devices support dynamic as well as static addressing.

Before installing the V207 in the VXIbus chassis, you must set the switch register to an appropriate value. If you wish to employ static addressing you must make sure you set the switch register to a unique value other than 0 or 255. It is a good idea to note module addresses in an accessible log, because if you replace a module, it is very important that the new module have the same address as the replaced one.

If your system employs dynamic addressing, which delegates the task of assigning device addresses to the Resource Manager software, then make sure the address switch is set to 255 (all **"1"**s).

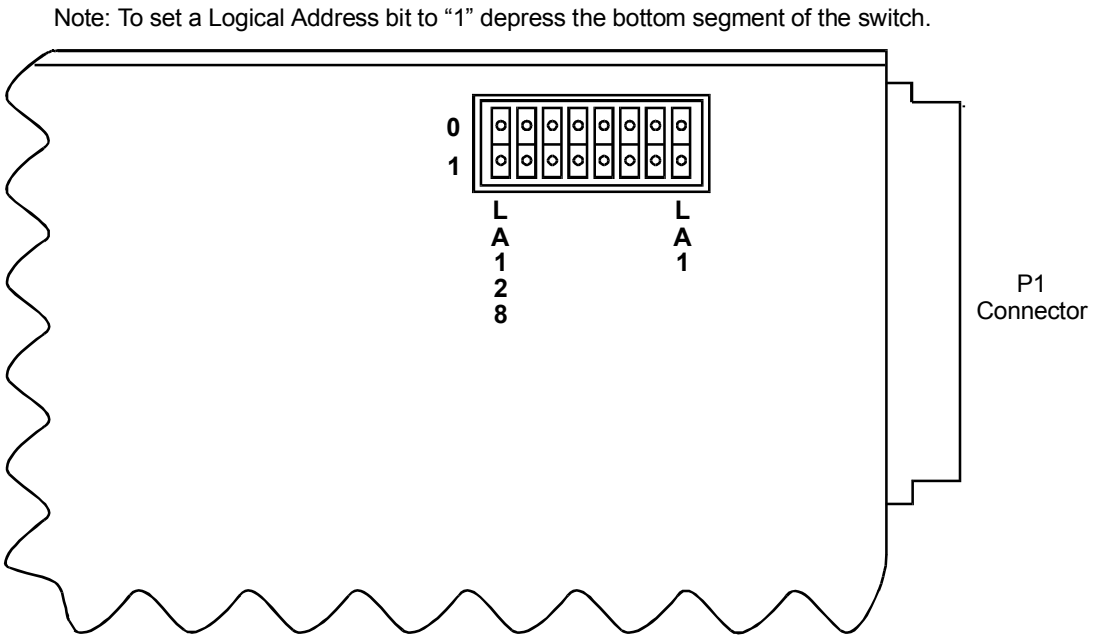


Figure 2-1. V207 Switch Locations

Module Insertion

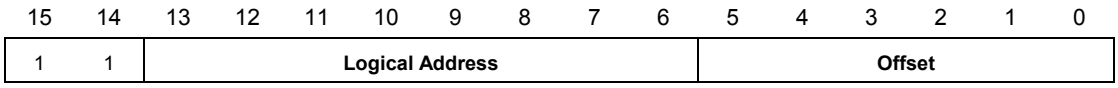
Before inserting your VXIbus module into the chassis, make sure that the chassis is plugged into electrical power but *not turned on*. The power cord provides a ground connection for the mainframe and protects the equipment and you from electrical harm.

In a VXI system, the Bus Grant and IACK signals are received and transmitted by each of the modules. These signals must be jumpered around any vacant slots in the mainframe. Most

current mainframes, including our V194 and V195, contain jumperless backplanes, where the Bus Grant and IACK signals are automatically jumpered when a slot is empty.
If your mainframe does not contain a jumperless backplane, you must position certain jumpers correctly on the chassis backplane to assure that the V207 acknowledges interrupts properly. Remove the Interrupt Acknowledge (IACK) jumper from the slot selected for the V207 and install daisy-chain jumpers in any empty slots between the V207 and the Slot 0 Controller.
You can now insert the V207 into the chassis. Slowly push it in until its plug connectors are resting against the backplane connectors. Then, using evenly distributed pressure, press the module straight in until it seats in the slot and the module front panel is even with the chassis front panel. Tighten the top and bottom screws.
You may now safely apply power to the V207.

Module Configuration

You, or your software, must perform two types of module configuration. The first has to do with VXIbus-related items and involves communication with V207 *configuration* registers. The second deals with setting parameters related to data acquisition and involves communication with V207 *operational* registers.
VXIbus-related configuration includes setting the logical address, specifying the amount of memory space required, specifying where in memory the V207 registers and memory blocks are located, and setting interrupt levels.
When you configure the V207 for data-acquisition, you must select a sample-clock source and clock rate, load the scan list in Scan RAM, and perhaps define Multi-buffer or **Digi-bus** parameters.
VXIbus devices occupy system memory space. The configuration registers for each VXIbus device have 64 bytes of memory space in the upper 16 kbytes of the 64-kbyte **A16** memory space. Whether you set the 8-bit Logical Address statically in the switch register or dynamically in the Logical Address register, those eight bits determines the base address of the 64-byte block of memory as follows:



Each 64-byte block contains several registers that supply information about the module, such as the manufacturer, the module identifier (i.e., "207h"), its class (register-based or message-based), serial number, and the amount of memory space it requires.
In addition to **A16** addressing, a VXIbus device can also support **A24** or **A32** addressing. If a Multi-buffer option is chosen, the V207 requires either 16 or 32 Mbytes of memory space. The V207 therefore supports **A32** addressing since the maximum size of **A24** memory is 16 Mbytes. The operational registers are all in **A32** space. To access them, one must first write a proper offset value to the Offset register in **A16** space. Refer to Chapter 5 for details relating to the *configuration* and *operational* registers.

Chapter 3: Understanding the V207/V208 Family

Overview

The V207 is a member of a versatile family of analog-to-digital converter (ADC) modules and signal conditioning modules. You can use from one to eleven signal-conditioning modules, in a VXIbus mainframe, with a single V207 or V208 scanning ADC module. The signal-conditioning modules transmit their analog data to the host ADC over the VXI Local Bus using **MUX-bus** protocol.

The V207 transmits its digitized data to a slot-0 controller via the VXIbus backplane *or* to one or more companion modules via the Local Bus using **Digi-bus** protocol. The flow of data is from right to left. Signal conditioning modules using **MUX-bus** protocol are positioned to the right of the ADC; slot-0 controllers and modules receiving digitized data using **Digi-bus** protocol are positioned to the left.

The V207 provides aggregate sampling rates to 500 ksamples/s. It supports up to 256 **MUX-bus** analog input channels. The V208 provides sampling rates to 100 or 200 ksamples/s. It supports up to 2048 **MUX-bus** channels.

The **MUX-bus** family of analog front-end modules includes:

- **V241** **96-channel multiplexer** for high-level signals that do not require conditioning;
- **V243** **96-channel signal conditioner** with programmable gain, programmable 2-pole active filters and precision calibration reference (V208 only);
- **V246** **8-channel, bridge-input, signal conditioner** with programmable gain, programmable 6-pole filters, per-channel excitation source and optional simultaneous sampling;
- **V252** **16-channel signal conditioner** with factory-installed 8-pole filters, optional programmable gain and optional simultaneous sampling; and
- **V253** **16-channel signal conditioner** with programmable gain, optional 6-pole filters and optional simultaneous sampling.

You can "mix and match" these front-end modules with a single V207.

The **Digi-bus** family of modules includes:

- **V110** **RAM memory module** with capacity to 128 Mbytes, on-board DSP for data manipulation, and programmable pre-trigger and post-trigger sample sizes for transient recorder applications;
- **V165** **TI 'C30-based DSP** with 40 MFLOP processing power;
- **V168** **SHARC-based DSP** with up to 1.44 GFLOP processing power;
- **V285** **16-channel waveform generator** with programmable clock rates and programmable Bessel filters on each channel;
- **V387** **128-channel digital input/output module**.

Data streams from the ADC to **Digi-bus** family modules over the VXI Local Bus using **Digi-bus** protocol and extends as far as adjacent modules in the VXIbus backplane propagate the bus. Brief descriptions of the **MUX-bus** and **Digi-bus** concepts and protocols follow.

The MUX-bus

The **MUX-bus** is a four-channel, differential, analog bus implemented on the 12-line VXI Local Bus. Under control of the V207, signal-conditioning modules multiplex up to 256 analog signals onto the **MUX-bus**. The V207 provides the signals to start a new scan (or "frame") and to step to a new channel.

The **MUX-bus** operates as a four-phase bus to provide ample settling time before digitizing occurs. The four analog buses are referred to as A, B, C and D. Figure 3-1 below illustrates the **MUX-bus** cycle for an 8-channel system.

The V207 and its signal conditioning modules each contain a scan table called "Scan RAM." These tables contain the channel scanning order. On the signal-conditioning modules, channels 0, 4, 8, 12, etc., are associated with **MUX-bus** channel A. Channels 1, 5, 9, 13, etc., are associated with channel B, and so forth. The order of the channels in the scan list must be such that analog signals are presented to the **MUX-bus** channels in the order A, B, C, D, A, B, C, D, etc. In other words, the two least significant bits must follow the sequence 00, 01, 10, 11, 00, 01, etc. Other than this restriction, you can select any order of channel scanning that you wish within a group of signal conditioning modules on a common **MUX-bus**. An example of proper Scan RAM setup is given in the next chapter.

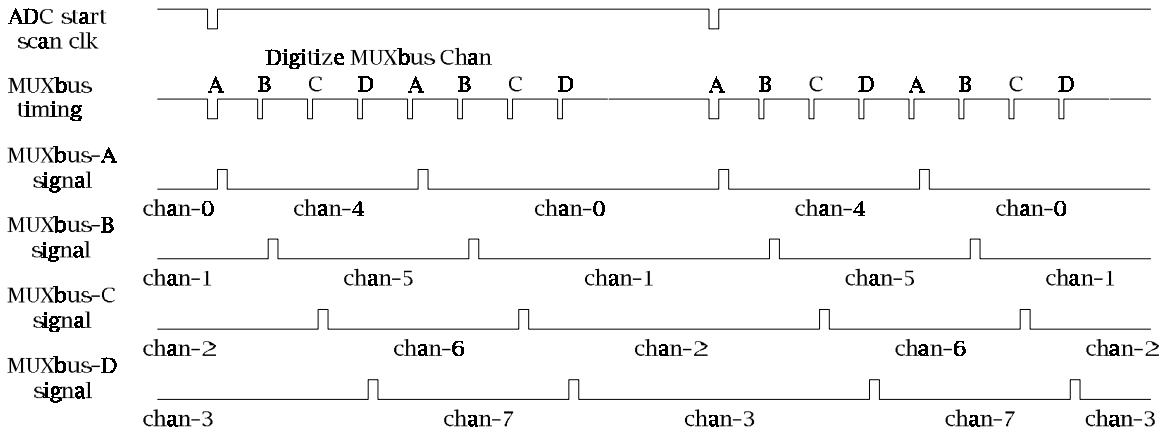


Figure 3-1. MUX-bus timing

The V207 Family of Signal Conditioning and Multiplexing Modules

A number of multiplexing and signal-conditioning modules use the **MUX-bus** protocol to connect to the V207 ADC. The family consists of the V241 high-level scanning multiplexer, the V246 bridge signal conditioner, and the V252 and V253 gain and filter modules. All are single-width, C-size, register-based VXIbus modules that support both static and dynamic addressing. The V243 low-level, low-noise module is intended for use with the V208.

V241

The V241 acquires data economically from pre-conditioned channels and is intended for medium to high channel count applications with high-level signals (± 10.24 V full scale). The V241 provides up to 96 high-level, differential input channels.

Two calibration channels are provided for each block of six input channels on a common multiplexer. One of each pair is internally set to analog ground (0 V differentially) while the other receives the **MUX-bus** system calibration voltage from the ADC card (+ 10 V differentially). This method allows end-to-end calibration of the V241 and V207.

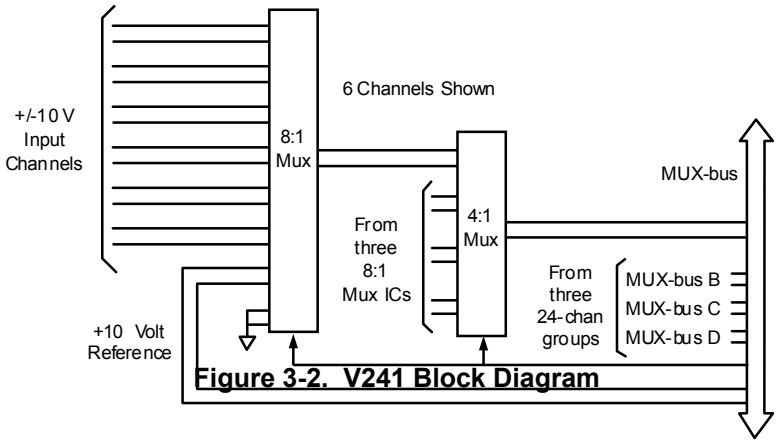


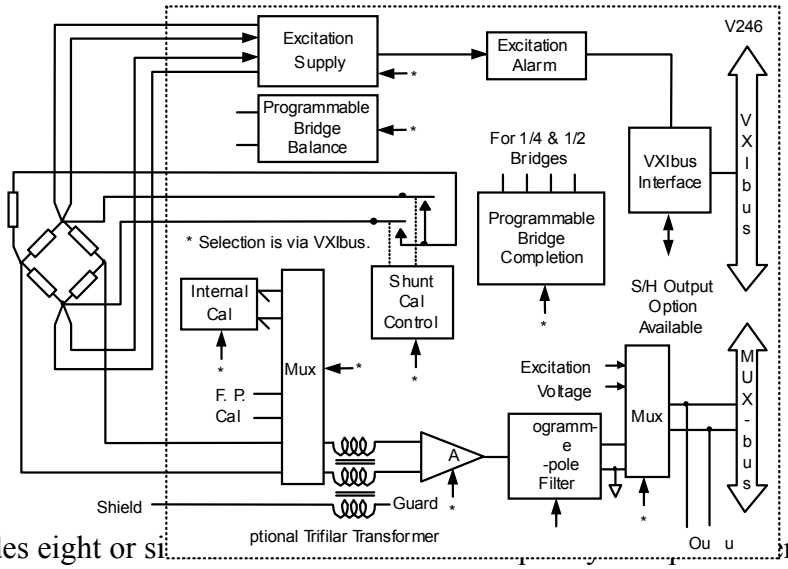
Figure 3-2. V241 Block Diagram

V246

The V246 provides eight channels of bridge signal conditioning. It accommodates transducers that represent one, two, or four active arms of a bridge circuit and permits the digitization of properly conditioned inputs from high-frequency strain gages, RTDs and other bridge-type sensors.

The V246 provides bridge completion, excitation, anti-aliasing filtering, and amplification for bridge-type inputs. It contains provisions for 2-point shunt calibration, automatic voltage and excitation calibration, as well as excitation alarms. A removable termination housing is available for convenient wiring of sensor leads.

Optional trifilar transformers are available for applications where high frequency electro-magnetic interference is a concern. Simultaneous sampling is also an option.



V252

The V252 provides eight or si ring. Gain is programmable on a channel-by-channel basis. You can select an eight-pole, low-pass, Bessel, Butterworth, Elliptic, or Constant-delay filter for each channel to provide high roll-off for excellent anti-alias filtering. Plug-in filter blocks provide cutoff frequencies that range from 10 Hz to 100 kHz.

Simultaneous sampling on all channels is available as an option. For calibration, the V207 provides a reference via the VXI Local Bus. An on-board reference can also provide a calibration input. The calibration signals can be applied to any channel.

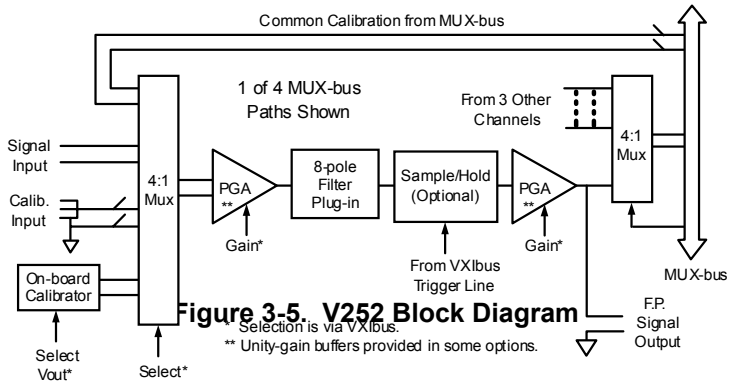


Figure 3-5. V252 Block Diagram

V253

The V253 contains sixteen channels of programmable-frequency, low-pass filtering. Gain is programmable on a channel-by-channel basis as well. Options provide low-pass 6-pole Bessel or Chebyshev filters for all channels.

Simultaneous sampling on all channels is available as an option. Calibration inputs are provided via the front-panel connector, a mainframe reference from the **MUX-bus**, or from an on-board reference. A removable termination housing is available for convenient I/O wiring.

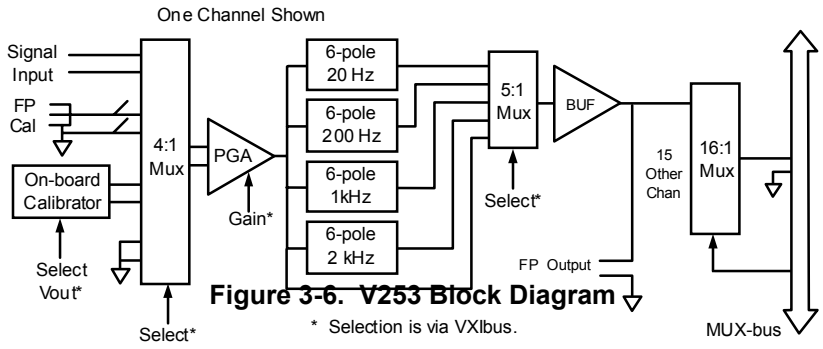


Figure 3-6. V253 Block Diagram

The Digi-bus

In multi-channel systems, the V207 generates a digital stream of data with each tick of the sample clock. For applications that require further buffering and processing of the data, the **Digi-bus** option allows the digital data from the V207 to flow to other modules without being subject to the timing uncertainties of the VXI backplane.

Figure 3-7 illustrates the basic **Digi-bus** concept. The V207 is the source of the data stream and **Digi-bus** timing signals. The V387 digital input module can provide additional discrete data to the stream. The V110 memory module provides multi-buffered access to the data via the VXIbus. The data stream terminates at the V165 DSP, and the DSP software determines what passes on to the left. The V285 waveform generator and the V387 digital input/output module select items from the data stream for output.

The **Digi-bus** supports multiple data sources as well as multiple data sinks. The left-most module receiving digital data also must terminate the bus. The **Digi-bus** supports transfer rates up to 10 Mbytes/second.

The **Digi-bus** and Multi-buffer options are mutually exclusive. If you plan to use more than one V208 or V207 with **Digi-bus** output in the same mainframe, note that **Digi-bus** and MUX-bus use the VXIbus Local Bus. Therefore, if more than one ADC module has access to a **Digi-bus**, only an ADC module positioned at the right end of the **Digi-bus** group of modules can make use of the MUX-bus protocol.

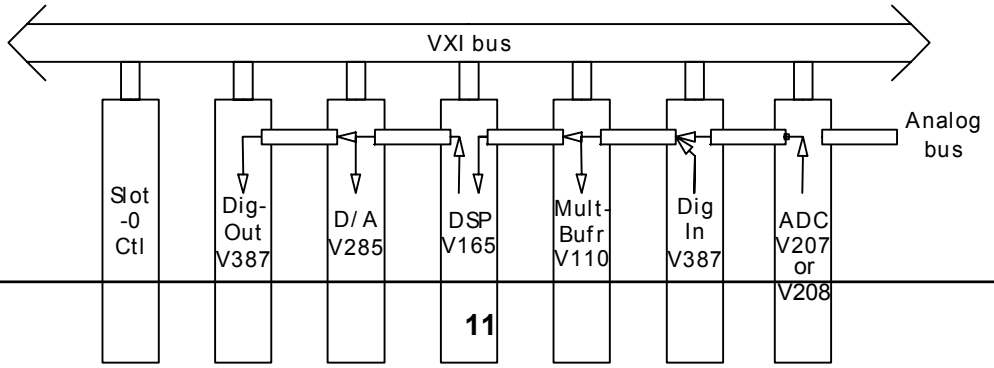


Figure 3-7. Digi-bus concept

Chapter 4: Understanding the V207

Overview

The V207 provides 16-bit analog-to-digital conversion at rates of 500 ksamples/s. It is used with signal-conditioning multiplexing modules that use the analog **MUX-bus** protocol to pass analog signals to the V207.

The V207 provides four differential inputs that allow high data throughput with large numbers of input channels connected via the **MUX-bus**. The V207 also accepts inputs via front-panel connectors. Note that you cannot use inputs from the **MUX-bus** and front-panel connectors at the same time.

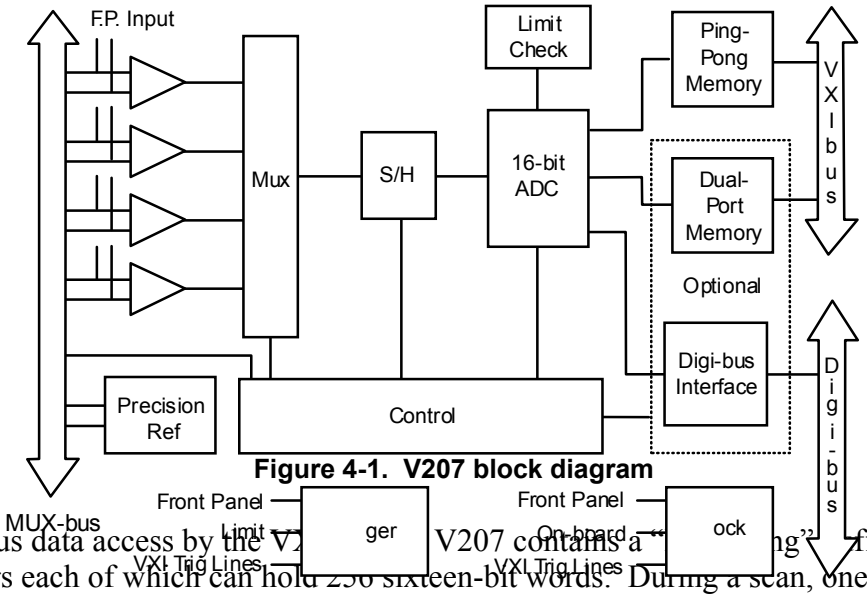


Figure 4-1. V207 block diagram

For continuous data access by the V2 V207 contains a “ ng” ffer, which is a pair of buffers each of which can hold 256 sixteen-bit words. During a scan, one buffer is being filled with ADC data from the multiplexed channels while the other contains the data from the previous scan and is available for readout from the VXIbus. At the start of the next scan, the roles of the buffers swap. The processor has one scan period to read the data from the V207. For example, if the scan clock rate is 100 Hz, the processor must recognize that the data is available and read it within 10 msec.

When the scan rate is above 100 Hz, the processor may be unable to respond quickly enough to empty the Ping-Pong buffer. The problem is that most operating system environments have worst-case latencies that are well over 1 ms. The solution is to use the **Digi-bus** to pass the data to a DSP or memory module; to use a large on-board buffer, a “multi-buffer,” that can hold data from a number of scans and be read more efficiently with large DMA block transfers; or to use a Slot-0 controller, such as the V160 Grand Interconnect controller, with Multi-buffer option.

The V207-.D.. Multi-buffer option is available in two sizes: 4 Mbytes and 16 Mbytes. You use the operational registers in **A32** address space to specify the size of the buffer most suitable for your application and to divide the buffer into a number of segments. The ADC treats the memory buffer as a circular buffer. That is, when it reaches the end it resets its address to the start of the buffer. When the ADC fills a segment, the Multi-buffer sets a “segment-full” flag

and signals the user application with an interrupt. The operating system responds and reads data from the filled segment(s) while the ADC is filling a subsequent segment.

An important feature of a V207 with a Multi-buffer is *transient capture* mode. Transient capture refers to the ability of this module to store a block of data related to an "event" in its local Multi-buffer, and then transmit that data over VXIbus, 32 bits at a time. When this mode is activated, a V207-.D.. begins recording its data into the circular Multi-buffer. The size of the buffer and the number of pre-trigger and post-trigger samples are programmable. The V207 can receive a trigger from its front-panel *Trigger In* connector or a VXIbus trigger line, or it can be triggered from software. Once triggered, it begins post-trigger countdown. When this countdown is complete, the Multi-buffer—containing pre-trigger and post-trigger samples—can be read via VXIbus. Therefore, VXIbus and host-computer transfer rates are not limiting factors when using this mode. Data storage can be at the maximum rate for the V207, with the recording time only limited by the size of the Multi-buffer.

If the **Digi-bus** option is present, the V207 acts as data source and “master” for the **Digi-bus**. It determines the **Digi-bus** clocking time, which relates to the ticks of the ADC conversion clock. It also provides a data “frame” signal, which is equivalent to the scan period. The **Digi-bus** is intended for use in continuous, rather than transient, data-taking applications.

V207 Specifications

Item	Specifications
Input Channels	
Number	4, expandable to 256 via the **MUX-bus**
Source	Front-panel or **MUX-bus** on the P2 connector
Analog Signal Input Type	Differential
Analog Input Range	± 10.24 V
Input Impedance	Greater than 20 MΩ
Large Signal Analog Bandwidth (- 3 dB)	Greater than 100 kHz
Resolution	16 bits, monotonic over operating temperature range
DC Accuracy	0.02% FSR (FSR = 20.48 volts)
Differential non-linearity	0.006% of FSR
Integral linearity	14 bits
No missing codes	Guaranteed
Dynamic Performance	
Maximum conversion rate	500,000 samples/second
Cross-talk	-70 dB
Sample Clock	
Internal source	Derived from the VXIbus precision 10 MHz clock
Frequency choices	12 steps from 100 Hz to 500 kHz (1, 2, 5 … sequence)
External source	TTL signal to 500 kHz
Duty cycle	45 to 55%
Backplane source	1 of 8 trigger lines on the VXIbus P2 connector
External Trigger	
Source	Negative-going TTL signal
Minimum value	50 nsec
Limit Checking	Level, ± slope
Resolution	8 bits
Maximum value	± 10.16 V
Calibration	Precision 10 V reference on board, bussed to adjacent signal conditioning modules via the **MUX-bus** for end-to-end calibration
Front-panel Connectors	BNCs and SMBs
Power Requirements	
+5 V	3.2 A without Multi-buffer, 4.65 A with Multi-buffer
-5.2 V	40 mA
+24 V	130 mA
-24 V	130 mA
Environmental and Mechanical	
Temperature range	
Operational	0° C to + 50° C
Storage	-25° C to + 75° C
Relative humidity	0 to 85%, non-condensing to 40° C
Cooling requirements	10 CFM
Dimensions	340 mm x 233.35 mm x 30.48 mm (C-sized VXIbus card)
Front-panel potential	Chassis ground

Table 4-1. Performance Characteristics of the V207.

Setup Memories

Scan RAM

You can multiplex up to 256 analog-input channels into the V207. The contents of the scan memory (called "Scan RAM") determine the sampling sequence of the channels, and you must write the desired sequence into the 256-word Scan RAM before acquisition can begin. As each channel is converted, an address register increments, and the next location in Scan RAM selects the channel to be digitized. The same address register routes the converted data to the appropriate sequential location in Ping-Pong, Multi-buffer, or **Digi-bus** memory.

Scan RAM exists in *all* **MUX-bus** modules. You must write the *same* sampling sequence into each Scan RAM, with an important exception. You must set bit 14 to a "**1**" in a module's scan RAM to indicate when the sequence applies to that particular module. In other words, when bit 14 is set, that module gates its selected analog signal onto the **MUX-bus**. Obviously, you must take care to not set bit 14 in more than one signal-conditioning module at the same address location.

Following is an example of loading the Scan RAM for the following module configuration:

- One V207 host ADC module and two V246 8-channel bridge signal conditioning modules.
- All 8 channels in each V246 are included in the scan list.
- V246 #1 is to be first in the list, followed by V246 #2.
- The channels in each V246 are to be scanned in order, 1 through 8.

The organization of the Scan RAM is as follows:

- Bit 15 is set to **"1"** to define the end of the list.
- Bit 14 is set to **"1"** in a signal conditioning module to connect a channel in *that module* to the **MUX-bus** at that point in the scan list.
- Bits 13 and 12 are not used in any Scan RAM and should be written as **"0."**
- Bits 11 through 0 define the channel to be scanned.
- Bits 1 and 0 *also* define the **MUX-bus** channel (A, B, C or D) being used.
- Bits 14 through 2 are ignored in the V207 ADC. Only the end-of-list flag (bit 15) and the **MUX-bus** channel selection (bits 1 and 0) are used in the V207's Scan RAM.

The Scan RAM setup considerations are:

- Bit 14 must be set to **"1"** in *only one* signal conditioning module at any Scan RAM address. Otherwise, more than one signal conditioning module will be connected to the MUX bus at the same time. This error will cause an "overlap" indication in one or more signal conditioning modules.
- Four **MUX-bus** channels are used to provide sufficient data settling time. If a sequence other than A, B, C, D (as defined by the two least significant bits in the channel address) is used, this sequence must be repeated throughout the scan list to produce predictable results. This also means that the number of channels in a scan list must be divisible by 4 (4, 8, 12, 16, etc.).

- The end-of-list flag (bit 15 = **"1"**) must be in the same list location in the ADC and in all signal conditioning modules connected to it. This flag is placed in the list location that represents the last channel scanned in the scan list.
- There are many bits that are ignored in the V207's Scan RAM, and the channel address bits related to one signal conditioning module are ignored by others in the subsystem. However, a recommended convention is to write the same data pattern for bits 13-0 in the Scan RAM for the ADC and all signal conditioning modules, while writing "**0**" for bit 14 for all locations in the V207 Scan RAM and the appropriate values for bit 14 in the signal conditioning Scan RAM memories.
- All Scan RAM data is ignored for addresses beyond that which contains the end-of-list flag. Good convention would have those data words written with all "**0**'s."
- When using multibuffer memory, only an even number of channels is allowed.

The following charts show the bit patterns to be written in the Scan RAM memories for this V207-V246 example:

Scan RAM - V207 ADC

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Data	MUX	Selection
200h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
202h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
204h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
206h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
208h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
20Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
20Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
20Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0007h	D	
210h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
212h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
214h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
216h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
218h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
21Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
21Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
21Eh	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	8007h	D	End List
220h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
3Feh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		

Scan RAM - V246 Signal Conditioner #1

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Data	Mux	Selection
100h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h	A	Channel 1
102h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001h	B	Channel 2
104h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002h	C	Channel 3

106h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4003h	D	Channel 4
108h	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4004h	A	Channel 5
10Ah	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	4005h	B	Channel 6
10Ch	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	4006h	C	Channel 7
10Eh	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	4007h	D	Channel 8
110h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0000h	A	
112h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
114h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
116h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
118h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0004h	A	
11Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
11Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
11Eh	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	8007h	D	End List
120h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
10Feh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		

Scan RAM - V246 Signal Conditioner #2

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Data	Mux	Selection
100h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
102h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
104h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
106h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
108h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
10Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
10Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
10Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0007h	D	
110h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h	A	Channel 1
112h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001h	B	Channel 2
114h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002h	C	Channel 3
116h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4003h	D	Channel 4
118h	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4004h	A	Channel 5
11Ah	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	4005h	B	Channel 6
11Ch	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	4006h	C	Channel 7
11Eh	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	C007h	D	Channel 8 + End List
120h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
10Feh	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		

Limit Memory

You can assign to each channel a unique limit value that is stored in Limit Memory as an eight bit digital count and a slope bit. The eight bits correspond to the eight most significant bits of the converted data word. As the V207 digitizes each channel, the value is checked to see if the limit is exceeded. If it is, it asserts a VXIbus interrupt if Global Limit Enable—bit 3 in the Setup register—is enabled. A VXIbus trigger line is also asserted if Limit Enable—bit 11 in Trigger Register 2—is asserted. You can route this event trigger *from* the VXIbus trigger line to the front-panel Trigger Out connector, if you desire.

Output Data Buffers

Ping-Pong Memory

Between ticks of the sample clock, the ADC converts one 16-bit data point each from the list of active channels in Scan RAM. The number of words generated between each clock tick can be as high as 256 in the V207. System design must address how to transfer the data to external storage before the ADC overwrites it with subsequent conversions. Some form of buffering is usually needed since required sample rates may prohibit the processor from keeping up on a conversion-by-conversion basis.

"Ping-Pong" buffering is the simplest form of buffering. The Ping-Pong buffer in the V207 consists of two banks of 256 sixteen-bit words each. During a period of the sample clock, the ADC puts its data into one bank of this memory. Following the next sample clock, the ADC puts its data into the other bank. While the ADC loads one bank, the processor unloads the other via the VXIbus. The user has one sample clock period to transfer the data out of the V207. It therefore is important to synchronize the reading of the data with the sample clock. One way to do this is to route the sample clock to a VXIbus trigger line via bits 3 – 0 of Trigger Register 1 (offset 02h in **A32** space). The processor can then monitor the selected trigger line and synchronize V207 data transfers.

Multi-buffer Memory Option

If the sample clock period is 10 ms or less, the processor may not respond in time to empty the buffer. In this case, a buffer that can hold data from a number of sample-clock periods is needed. The Multi-buffer option provides you with a 4 or 16 Mbyte block of memory for such purpose. You can configure this memory block in various ways to suit your application. Multi-buffering is a scheme that lets you define from one to eight buffer segments of equal size within all or a portion of the memory block. You use the nine **A32** registers at offset addresses from 20h to 36h to configure the memory and segments. The ADC fills the buffer segments in order, signaling the processor by interrupt or by a flag in the Full Flag register (offset 28h) as each is filled. The processor can begin reading data from Buffer #1 as soon as the Full Flag #1 is set. As the processor is reading the Buffer #1 data, Buffer #2 is being filled with new data. This process continues for the number of buffers desired. It is the processor's responsibility to empty each buffer before the ADC is ready to fill it again. Failure to empty the contents of a buffer within this time span will cause the Overrun bit to be set in the Full Flag register, signifying that data has been lost.

You can also configure the Multi-buffer for transient applications that need a large memory buffer. In transient mode, an "event" trigger starts post-trigger countdown. This trigger can be

an event generated by a Limit Violation, an external input, or an event on one of the VXIbus Trigger lines. The buffer fills with interleaved channel data until the post-trigger countdown reaches zero. The contents of this buffer will remain valid until the Transient mode is re-armed by disabling and re-enabling the Transient Enable bit in the Setup register (offset 06h in **A32** Space).

NOTE: When using multibuffer memory, only an even number of channels is allowed.

Timing Parameters: the sample clock, triggers and interrupts

Selecting the Sample Clock Source

As you can see from Figure 4-2, you can select the sample clock from three sources: a selected VXIbus trigger line, an external clock via the front-panel connector, or the internal clock derived from the 10 MHz VXIbus system clock. The sample clock can also *drive* a selected VXIbus trigger line. You make sample-clock selections via the Sample Clock register (at 00h offset in **A32** space). Trigger line selections are made via the Trigger Register 1 (at 02h offset in **A32** space).

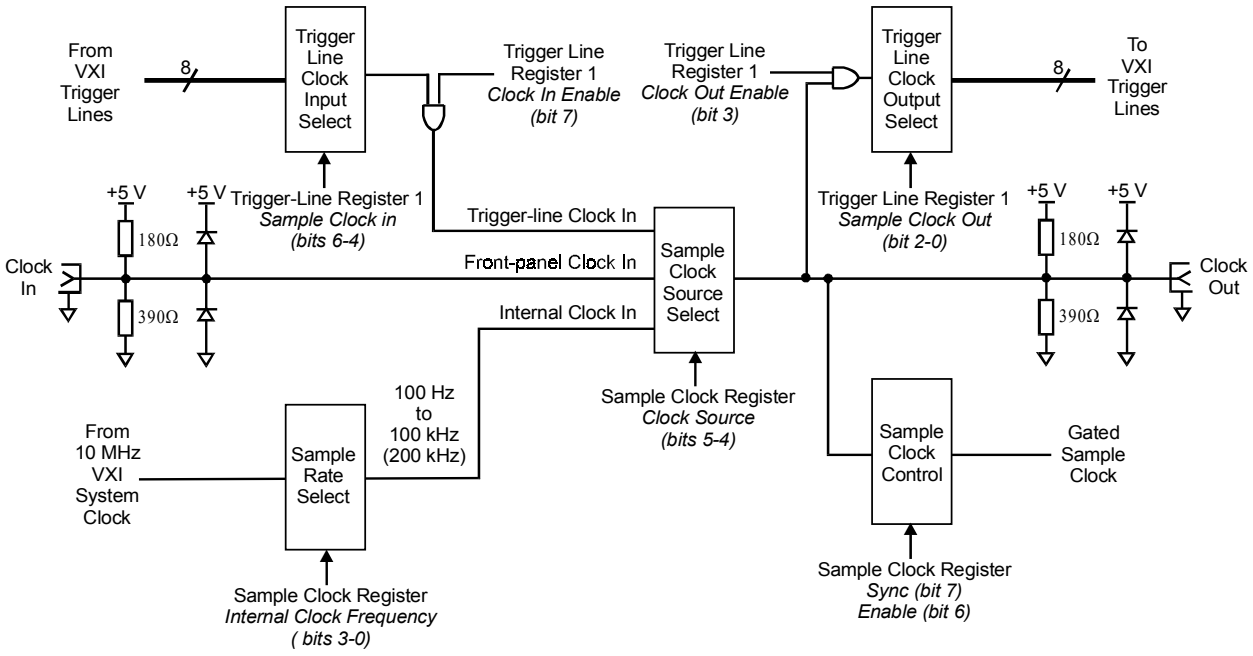


Figure 4-2. Sample Clock Block Diagram

Selecting the Sample Clock Rate

The frequency of the sample clock is the rate at which samples are digitized from each channel. The ADC digitizes at the maximum rate of 500 kHz regardless of the sample clock frequency. The *theoretical* maximum sample clock rate, therefore, is 500 kHz divided by the number of channels you are digitizing. However, the *actual* maximum rate may be less than the theoretical maximum since the number of channels being digitized is assumed to be divisible by four and the available sample-clock frequencies form a 1, 2, 5 sequence. The resulting product of the number of channels and the sample-clock frequency may well be less than the theoretical maximum rate.

A V207 ADC module can address up to 256 channels of signal conditioning, with the modules coupled to the V207 via the backplane **MUX-bus**. The following definitions aid in understanding the V207 scanning process:

- The *sample clock* determines the rate that the active channels (the channels selected by the Scan RAM) are sampled. For example, if 96 channels are included in the scan, and a 1 kHz sample clock is chosen, all 96 channels are scanned once every millisecond (at a 1 ksamples/second rate).
- The *ADC clock* operates at the V208 module's maximum rate of 500 kHz, regardless of the sample clock selection. This clock is derived from the 10 MHz CLK10 on the VXI backplane. Analog-to-digital conversion occurs at this ADC clock rate.

Figure 4-3 shows a V207-Zxyz with its ADC clock rate of 500 kHz, 16 active signal conditioning channels and a sample clock rate of 20 kHz. This provides a per-channel sample rate of 20,000 samples per second and a data throughput rate of 320 ksamples/s.

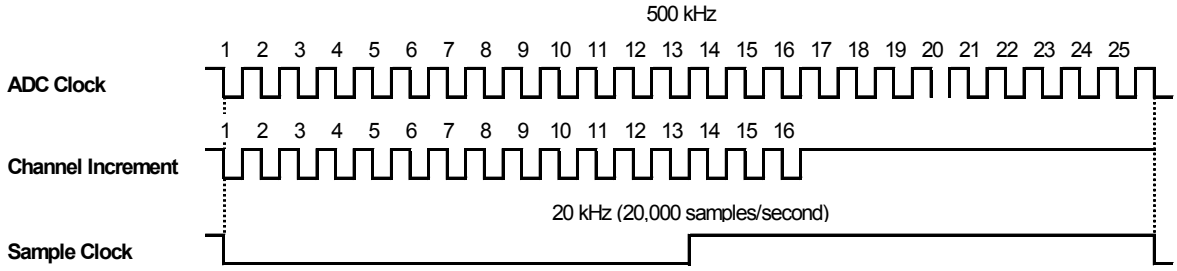


Figure 4-3. Relationship of ADC Clock to Sample Clock

One of three sources can be selected for the sample clock: the internal sample clock, the front-panel clock input or a selected VXI trigger line. The internal sample clock is programmable from 100 Hz to 500,000 Hz in a 1, 2, 5 progression. This clock is derived from the CLK10 source. The maximum rate that can be used for the sample clock depends upon the number of active channels in the scan. The formula is:

V207 Maximum Sample Clock Rate (Hz) = 500,000 Hz / Number of Active Channels

For example, a V207 with 64 channels has a maximum sample clock rate of 7812.5 Hz. If an external clock source is used, the frequency tolerance of CLK10 (± 0.01%) and that of the external clock must be evaluated to assure that the actual ratio of the two clocks will not exceed the limit. Table 4-2 shows the maximum number of active channels that can be supported by the various internal sample clock rates. This table also takes into consideration the need to have the number of channels in a scan divisible by four to provide proper analog signal settling of the four **MUX-bus** segments.

Sample Clock (Hz)	Maximum Number of Channels		
	V207-Zxyz	V208-Yxyz	V208-Zxyz
500,000	(Note 3)		
200,000	–	–	(Note 2)
100,000	4	(Note 1)	–
50,000	8	–	4
20,000	24	4	8
10,000	48	8	20
5,000	100	20	40
2,000	248	48	100
1,000	500	100	200
500	1,000	200	400
200	2,500	500	1,000
100	5,000	1,000	2,000

Table 4-2. Maximum Number of Channels with Internal Sample Clock

Note 1: When using the V208-Yxyz module front-panel signal input connectors (with no signal conditioning modules), the maximum sample clock rate is 100,000 Hz with one channel active and 20,000 Hz with four channels active.

Note 2: When using the V208-Zxyz module front-panel signal input connectors (with no signal conditioning modules), the maximum sample clock rate is 200,000 Hz with one channel active and 50,000 Hz with four channels active.

Note 3: When using the V207-Zxyz module front-panel signal input connectors(with no signal conditioning modules), the maximum sample clock rate is 500,000 Hz with one channel active and 50,000 Hz with four channels active.

Trigger Lines

The V207 makes use of the VXIbus trigger lines in a number of ways.

As Outputs

Using Trigger Registers 1 and 2, you can select VXIbus trigger lines to be driven by (if enabled):

- the sample clock selected within the V207; (Trigger Register 1, bits 3 – 0)
- the front-panel *Trigger In* signal; (Trigger Register 1, bits 11 – 8)
- a Limit Violation event trigger occurring in the V207. (Trigger Register 2, bits 11 – 8)

As Inputs

Using Trigger Registers 1 and 2, you can select trigger lines to provide (if enabled):

- the sample clock to be used by the V207; (Trigger Register 1, bits 7 – 4)
- a front-panel *Trigger Out* signal; (Trigger Register 1, bits 15 – 12)
- a trigger to start post-trigger countdown for Multi-buffer transient data capture. (Trigger Register 2, bits 3 – 0)

Figure 4-4 illustrates the trigger line organization in the V207.

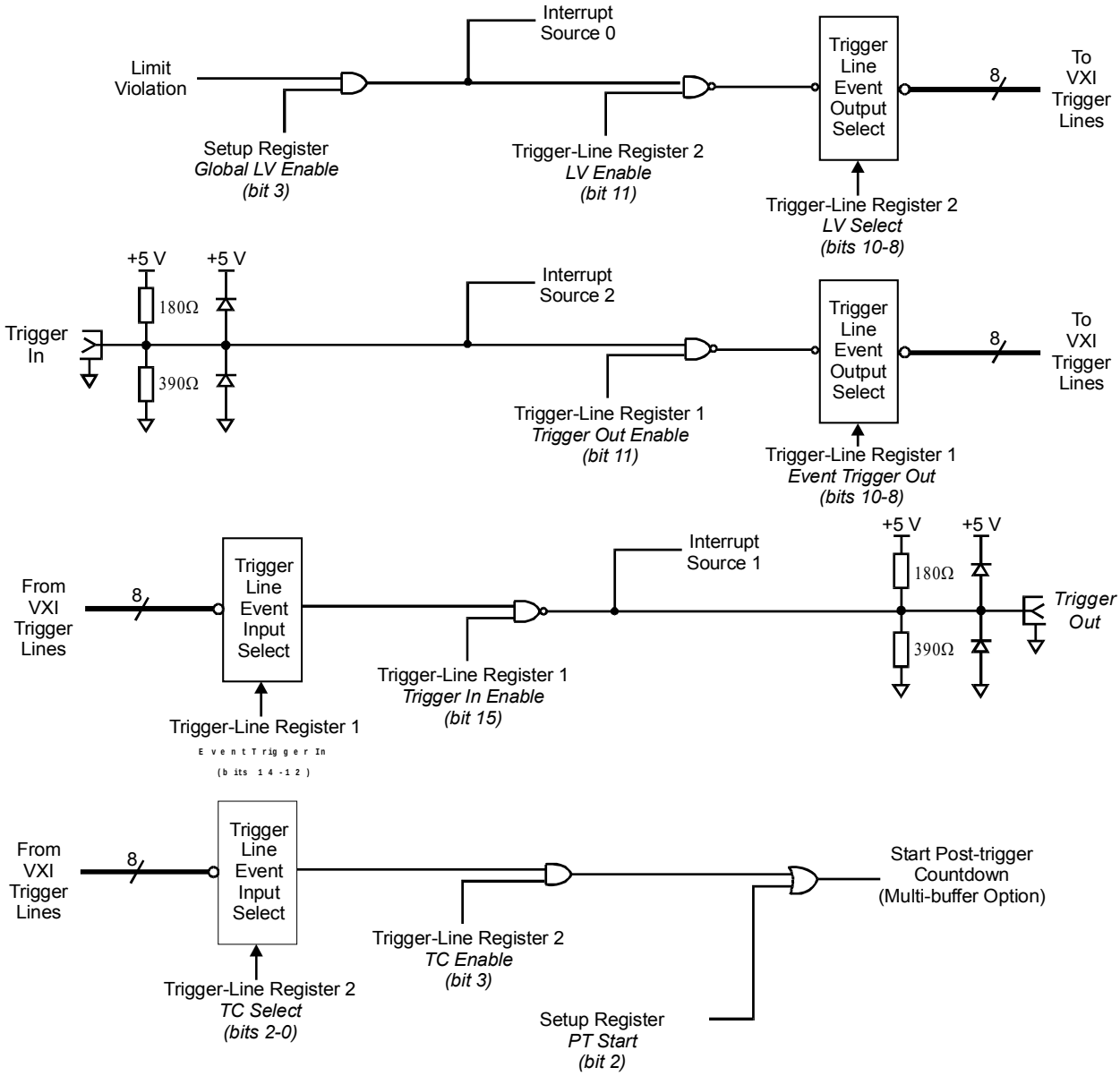


Figure 4-4. Event-Trigger Routing Diagram

Event Triggers

An event trigger that starts the post-trigger countdown for the capturing of transient data originates either from a VXIbus trigger line or by software trigger via bit 2 in the Setup register. An event trigger on a VXIbus trigger line comes from a limit violation on a V207 data channel or from the front-panel *Trigger-In* connector.

An internal limit violation and/or a front-panel trigger can assert an interrupt on the VXIbus. A VXIbus trigger that appears at the front-panel *Trigger-Out* connector can also assert an interrupt.

Timing Considerations in the V207

In the V207 all timing signals are derived from the 10 MHz clock provided by the Slot-0 controller and distributed across the backplane. This guarantees synchronous operation for multiple V207s in a common mainframe.

In many applications you may wish to compare data from different channels, and maybe different ADCs, at a known time. Since the data sampling in a channel is a function of the ADC sample clock, time is essentially represented by the number of samples from the start of data acquisition. If the ADC sample clocks were derived from different crystal oscillators, for example, the sampling time among ADCs would tend to drift slowly due to slightly different oscillator frequencies. Deriving all clocks from a common, precision frequency source guarantees that all data sampling is in a locked phase relationship. The VXI 10 MHz clock is a good choice since the VXIbus specification *requires* that the clock meet certain accuracy requirements and is available at each module slot.

Synchronization of Multiple V207s

The V207 includes frequency dividers to derive the 500 kHz ADC clock and the programmable sample clock that initiates a scan of the input channels. At system startup there is no guarantee that the dividers in different V207s are in phase and, thus, no guarantee of the phase relationship between channels of different V207s.

The V207 provides for the synchronization of V207s in a single mainframe to ensure that the timing for the sampling of given channels among V207s is the same. One uses the *Sync* and *Enable* bits in the Sample Clock register (00h, **A32** space) to accomplish synchronization. For *Sync* = **"1"** and *Enable* = **"0,"** a VXIbus trigger, as selected via Trigger Register 1 (bits 14 – 12), clears the internal counters. For *Sync* = **"1"** and Enable = **"1,"** a trigger transition on the selected VXIbus trigger line begins synchronous sampling. Note that the trigger event on the VXIbus trigger line must be synchronous with the 10 MHz clock.

Interrupts

The V207 has eight interrupt sources, shown in Table 4-3. You can enable or disable individual interrupts via mask bits in the read/write Interrupt Control register (at 1Ch in **A16** space). This register also allows you to collectively enable or disable the V207 interrupts as well as to set the Interrupt Request Level. You can determine the status of the eight interrupt sources by reading the Interrupt Status register (at 1Ah). An "interrupt pending" LED is on the front panel.

Interrupt	Interrupt Source #	Bit positions in the Interrupt Status and Interrupt Control registers
Limit Violation	0	8
External Event from the VXIbus Trigger Lines	1	9
External Event from the Front-panel Trig In	2	10
Multi-buffer Full Flag #1	3	11
Multi-buffer Full Flag #2	4	12
Multi-buffer Full Flag #3	5	13
Multi-buffer Full Flag #4	6	14
Transient Complete or Multi-buffer Full Flag >#4	7	15

Table 4-3. Interrupt Sources

Front Panel

LEDs

Add Rec

Illuminated when the module is being accessed.

Failed

Illuminated if the V207 has failed its self-test.

Int Src

Illuminated as long as the V207 has an interrupt pending.

Active

Illuminated whenever the ADC is converting data.

SMB Connectors

Sample Clock In

You can connect an external sample clock that initiates sampling if selected by software. Signal levels must be TTL-compatible. The rising edge of the signal triggers sampling.

Sample Clock Out

You can monitor the internal sample clock as well as use it as a timing source. The signal is TTL. The rising edge signals start of sampling.

Trig In

You can connect an external trigger source that triggers an action in the V207 if enabled by software. Signal levels must be TTL-compatible. The falling edge of the signal initiates the action.

Trig Out

You can monitor the event trigger as well as use it as a timing trigger. Signal levels must be TTL-compatible. The falling edge of the signal initiates the action.

Other Connectors

Reference Monitor

The internal 10-volt reference can be monitored by connecting a digital voltmeter with high input impedance across the +/- terminals. You can adjust the reference via the "Ref Adj" potentiometer.

Channel Inputs

If you are not using the **MUX-bus** to multiplex input signals to the V207, you can use the four isolated BNC connectors to connect up to four input signals to this module. The shell and center contact of each BNC connector represent the two lines of a differential pair. The external signal source must be grounded, and that ground must have some connection to the mainframe ground. Otherwise, the common-mode limit may be exceeded, resulting in poor data quality.

Note that these inputs are in parallel with the **MUX-bus** input signals. You should take care to avoid connecting a signal source to any of these connectors when modules that drive the **MUX-bus** are connected to the **MUX-bus.**

Product specifications and descriptions in this document subject to change without notice.

V207 Adjustment procedure

Required equipment

DC voltage source capable of ± 10 V and stable to ± 100 μV
8 ½ digit multimeter. Use this to verify the DC source.
Cable to connect DC source to DVM then to all four BNC connectors on the V207.
Potentiometer adjustment tool.

Purpose

The purpose of this calibration is to set the V207 to an optimum gain and offset for all four channels. The adjustment ensures that the V207s ADC is not saturated and that the gain is not so low that it decreases dynamic range and resolution.

Definitions

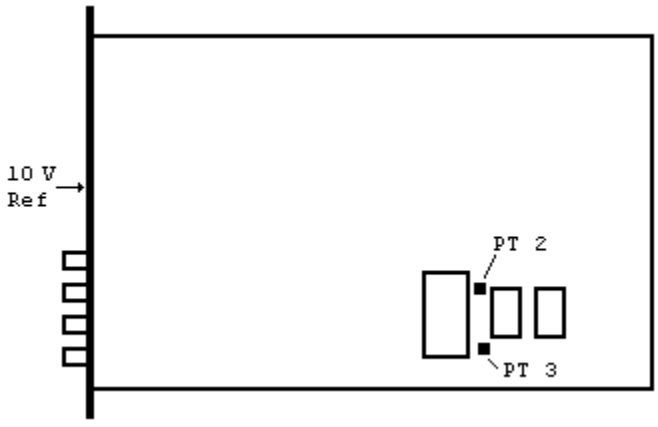
Two definitions are important:
1. The gain of the ADC is defined as 3200 counts/Volt. 1/gain gives 312.5 μV per bit.
2. The ADC counts with zero volts input is defined as 32768.

Preliminary adjustment procedure

Allow the module to warm up for 30 minutes.
Ensure that there is no module in the slot to the right of the V207.
Use the following procedure to take readings:
1. Connect the DC source to all four channels.
2. Setup the V207 to take samples on all four channels.
3. Take the average of the four channels output. Repeat the averaging 500 times (2000 total samples). The average of the 500 four channel averages will be referred to as the *reading* for the voltage input on all four channels.

$$reading = \frac{1}{500}\sum_{n=1}^{500}\frac{ch1 + ch2 + ch3 + ch4}{4}$$

Adjustment procedure



1. Connect the DC source to the DVM and to all four BNC inputs on the V207.
2. Set the DC source to 0.0 V ± 100 μV according to the DVM.
3. Adjust the offset pot (PT 3) so that the *reading* of the zero volt input is 32,768 ± 2 counts.
4. Set the DC source to +10.0 V ± 100 μV according to the DVM.
5. Record the *reading* (according to step 3 of preliminary adjustment) as $Fs_{positive}$.
6. Set the DC source to -10.0 V ± 100 μV according to the DVM.
7. Record the *reading* as $Fs_{negative}$.
8. Calculate the gain: $gain = Fs_{positive} - Fs_{negaitve}$
9. Adjust the gain pot (PT 2) so the *gain* (difference between +10 V and –10 V input) is 64,000 ± 5 counts.

10. Connect the DVM to the +10 V reference on the front panel. Adjust the Ref Adj pot on the front panel so the reference is 10.0 V ± 50 μV.

Chapter 5: Configuration and Operational Registers

Address Space

VXIbus uses the VMEbus protocol for data transfer and therefore supports 32-bit addressing to access I/O slave devices. 32-bit addressing provides direct access to memory space of four Gigabytes.

Slave devices such as VXIbus data acquisition modules exist for a variety of purposes and can be simple or very complex. Communication between host and slave can require access to several registers in one device or access to many Mbytes of memory in another. ("Devices" and "modules" are terms often used interchangeably. The distinction is that more than one VXIbus device *can* reside in a VXIbus module. However, there is generally one device per module.)

To minimize the amount of address-decoding hardware needed, simpler slave devices use addressing modes that fully decode only 16 or 24 address lines rather than 32. Therefore, there are three defined addressing modes…**A16**, **A24** and **A32**…having address spaces of 64 kbytes, 16 Mbytes and 4 Gbytes, respectively.

All VXIbus devices have registers located within 64-byte blocks in **A16** address space and therefore support **A16** addressing. Devices requiring no more than 64 bytes of address space need only support **A16** addressing. Devices needing more than the 64 bytes to accommodate additional registers or blocks of memory *must* also support **A24** or **A32** addressing, but not both. VXIbus devices that use **A24** or **A32** addressing modes are required to have four registers in **A16** space for parameter definition. One such parameter is Required Memory, which uses four bits (*m*) to specify the size of the memory in **A24** or **A32** space required by the device. A device *may not* use more than one-half of the memory space, and it *should not* use more than one-fourth. Table 5-1 shows the relationship between the four-bit parameter, *m,* and the memory required by the device. Note that *m = 0* defines the case for maximum usage, i.e., half of the memory space. Required Memory is specified in bits 15 – 12 in the Device Type register at offset 02h.

m	Required Memory				*m*	Required Memory			
	A24		A32			A24		A32	
0	8	Mbytes	2	Gbytes	**8**	32	kbytes	8	Mbytes
1	4	"	1	"	**9**	16	"	4	"
2	2	"	512	Mbytes	**10**	8	"	2	"
3	1	"	256	"	**11**	4	"	1	"
4	512	kbytes	128	"	**12**	2	"	512	kbytes
5	256	"	64	"	**13**	1	"	256	"
6	128	"	32	"	**14**	512	bytes	128	"
7	64	"	16	"	**15**	256	"	64	"

Table 5-1. Relationship between the "*m*" Parameter and Required Memory

One of the four registers is the Offset register, which is needed only for devices using **A24** or **A32** address space. This 16-bit read/write register defines the base address of the device's **A24** or **A32**

operational registers. The *m+1* most significant bits of the Offset register provide the values of the *m+1* most significant bits of the device's **A24** or **A32** register addresses, where *m* is as defined in Table 5-1 above.

Static and Dynamic Configuration

A VXIbus system can have up to 255 devices. Therefore, eight bits define the device address, which is called the "Logical Address." The Logical Address can be "static" or "dynamic." A static address resides in an 8-bit switch register; a dynamic address resides in a write-only register. Setting the switch register to 255 (all **"1"**s) causes dynamic addressing to be enabled. Any other setting enables static addressing, in which case the value held in the switch register is the Logical Address.

With the Logical Address set to 255, a device responds to accesses at address 255 only when the MODID line is asserted as a qualifier by the Slot-0 controller. After a new Logical Address is written to the device, the device responds to the new address independent of the state of the MODID line.

For data acquisition and control applications, dynamic configuration is an important concept. A system often contains more than one module of a given type, and it can be easy, and sometimes desirable, to swap positions of two modules after removing them from the mainframe. If dynamic configuration is not employed, one must make sure that the switch register is correctly set when inserting or re-inserting a device. Dynamic configuration greatly simplifies system setup, since the software can assure that the devices are located in the desired slots. Dynamic configuration also allows a system's Resource Manager to configure memory usage optimally in a system.

Communication Protocol

VXIbus allows communication over the backplane by either register-based or message-based protocols. With register-based protocol, the communication is via an 8-, 16- or 32-bit parallel path directly to I/O registers within the modules. With message-based protocol, an ASCII interpreter is included on each module, and the binary representations of ASCII characters are transmitted over the backplane. The advantage of message-based protocol is that English-like commands and responses can be used.

High-performance data acquisition and control modules are usually register-based because the data throughput is usually several orders of magnitude greater than with message-based devices. All KineticSystems VXIbus devices are register-based.

Register Addressing

The user assigns each device in a VXIbus system a unique number between 1 and 254. This 8-bit number, called the Logical Address, defines the base address for the VXIbus device registers located on the module. Each device has a 64-byte block of memory reserved for these registers. The memory blocks, called configuration space, are located in the upper 16 kbytes of the 64-kbyte **A16** address space.

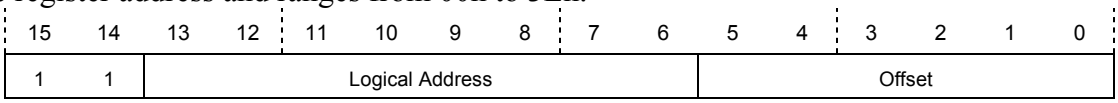
Every device has at least three configuration registers: ID / Logical Address, Device Type, and Status / Control. Modules using **A24** or **A32** addressing must also have an Offset register. The rest of the 64-byte block can contain registers or memoryappropriate for the operation of the specific device.

A device's Logical Address occupies bits 13 - 6 of the register address. Bits 15 and 14 of the address are both **"1"**s,**"** and the base address of the register block is therefore:

***V**40h+C000h**

where *V* is the Logical Address of the device and C000h is the starting address of the top 16-kbyte block.

The address of a specific register is the base address plus an offset address. The offset is bits 5 - 0 of the register address and ranges from 00h to 3Eh.



The V207 also uses *operational* registers in **A32 space; therefore,** it is an "Extended" register-based device.

Required Configuration Registers

The four required VXIbus registers are ID / Logical Address, Device Type, Status / Control, and Offset. You can access these registers by **D16** transfers only.

ID Register *00h*

This read-only register returns 5F29h.
Fields are Device Classification, Addressing Mode and Manufacturer ID.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	0	1	0	1	1	1	1	1	0	0	1	0	1	0	0	1
	Class = Extended		**Addressing Mode = A32**		**KineticSystems' Manufacturer ID = F29h (3881)**											

Device Classification *Bits 15 and 14*

00 Memory device
01 Extended device
10 Message-based device
11 Register-based device

The V207 is an *Extended device.*

Addressing Mode *Bits 13 and 12*

00 **A24**
01 A32
10 Reserved
11 **A16**

The V207 uses *A32 addressing*.

Manufacturer ID *Bits 11 through 0*

KineticSystems' Manufacturer ID is *3881*, which corresponds to F29h.

Logical Address Register *00h*

This write-only register holds the Logical Address. In systems using Dynamic Configuration, the system Resource Manager uses this register to set the Logical Address of the device.

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

Logical Address *Bits 7 through 0*

Device Type Register *02h*

This read-only register contains the Required Memory and Model Code for the V207. It returns 6207h or 7207h, depending on the size of the Multi-buffer option.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	x	x	x	x	0	0	1	0	0	0	0	0	0	1	1	1
	Required Memory				**Model Code = 207h**											

Required Memory ***Bits 15 – 12***

Examples

A value of **6h** indicates that the V207 uses 32 MB of **A32** memory space. **V207-ZD33**
A value of **8h** indicates that the V207 uses 16 MB of **A32** memory space. **V207-ZD23**
A value of **15h** indicates that the V207 needs 64 kbytes or less of **A32** memory space. **V207-ZC13**

Model Code ***Bits 11 – 0***

The model code for the V207 is 207h.

Status Register *04h*

This read-only register provides binary information about the status of the V207.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	A32 Active	MODID*	Not Used										Ready	Passed	Sysfail Inhibit	Soft Reset

A32 Active ***Bit 15***

"**1**" in this field indicates that the **A32** registers of the V207 can be accessed. This bit reflects the state of the Control register's **A32** Enable bit.

MODID* ***Bit 14***

"**1**" in this field indicates that the V207 is *not* selected via the P2 MODID line. A "**0**" indicates that the device is selected by a high state on the P2 MODID line. The Resource Manager uses this bit to configure the V207 dynamically.

Ready ***Bit 3***

"**1**" in this field indicates that the registers have been successfully initialized. The V207 is ready for access.

Passed ***Bit 2***

"**1**" in this field indicates that the device self-test has passed. A "**0**" indicates that the V207 has failed—or is currently executing—its self-test.

Sysfail Inhibit ***Bit 1***

"**1**" in this field indicates that the V207 is disabled from driving the SYSFAIL* line. This bit reflects the state of the Sysfail Inhibit line in the Control register.

Soft Reset ***Bit 0***

"**1**" in this field indicates that the V207 is in a reset state. While in this state, the V207 will allow access only to its Configuration registers.

Control Register *04h*

This write-only register causes execution of specific actions by the V207.

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

Write-only	A32 Enable	Not Used	Sysfail Inhibit	Soft Reset

A32 Enable ***Bit 15***

Setting this bit to "**1**" enables access of the **A32** registers of the V207.

Sysfail Inhibit ***Bit 1***

Setting this bit to "**1**" disables the V207 from driving the SYSFAIL* line.

Soft Reset ***Bit 0***

Setting this bit to "**1**" forces the V207 into a reset state.

Offset Register *06h*

This read/write register determines and reports the device base address in **A32** memory space.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Base Address in A32 memory space								0	0	0	0	0	0	0	0

Depending upon the options chosen, the V207 requires either 16 kB, 8 MB or 32 MB of the 4 GB **A32** memory space. The Required Memory parameter, *m*, in the Device Type register is therefore 15, 8 or 6, respectively. The *m*+1 most significant bits of the Offset register are the values of the *m*+1 most significant bits of the device **A32** register addresses. The 15 - *m* (i.e., 0, 7, or 9) least significant bits of the Offset register are meaningless.

Additional Configuration Registers

Additional configuration registers are:

- Attribute register
- Serial Number High & Low registers
- Version Number register
- Interrupt Status register
- Interrupt Control register
- Subclass register
- Suffix High & Low registers, and
- fourteen EEPROM user registers.

Note that you can access these registers by **D16** transfers only.

Attribute Register *08h*

This read-only register provides low-true information about the V207's interrupt handling capabilities. A read of this register returns FFFAh.

	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	1	1	1	1	1	0	1	0
	Reserved													**Interrupt Capability***	**Interrupt Handler Control***	**Interrupt Status Reporting***

Reserved ***Bits 15 - 3***

These bits are reserved for future use and return **"1"**s when read.

Interrupt Capability* ***Bit 2***

"**0**" signifies that the V207 is capable of generating interrupts.

Interrupt Handler Control* ***Bit 1***

"**1**" indicates that the V207 is *not* capable of Interrupt Handler Control.

Interrupt Status Reporting* ***Bit 0***

"**0**" indicates that the V207 has Interrupt Status Reporting capability.

Serial Number Register *0Ah, 0Ch*

The read-only Serial Number registers (high and low words) store the 32-bit hexadecimal value of the V207's decimal serial number.

0Ah

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	SN7				SN6				SN5				SN4			

Low word

0Ch

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	SN3				SN2				SN1				SN0			

High word

Version Number Register *0Eh*

This read-only register gives the hardware and firmware revision numbers of the module.

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

Firmware Version ***Bits 15 - 12***

Firmware Revision ***Bits 11 - 8***

Hardware Version ***Bits 7 - 4***

Hardware Revision ***Bits 3 - 0***

Each field is a four-bit integer indicating the version or revision number.

Reserved *10h, 12h, 14h, 16h, 18h*

These five read-only registers are reserved for future use. They return FFFFh.

	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	1	1	1	1	1	1	1	1

Interrupt Status Register *1Ah*

This read-only register provides information about the states of the eight interrupt sources. It can be used for polling of interrupts. Interrupt bits are cleared by a read of this register or by an interrupt-acknowledge cycle.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	TC	MBF4	MBF3	MBF2	MBF1	FPT	TTLT	LimV	1	1	1	1	1	1	1	1
	Interrupt Sources								**Reserved / Logical Address**							

Interrupt Sources ***Bits 15 - 8***

A "**1**" in any of these bit locations indicates that an action has occurred that can generate an interrupt.

Bit		Interrupt Source Description
15	TC	Transient Complete or Multi-buffer Full Flag #5 & up
14	MBF4	Multi-buffer Full Flag #4
13	MBF3	Multi-buffer Full Flag #3
12	MBF2	Multi-buffer Full Flag #2
11	MBF1	Multi-buffer Full Flag #1
10	FPT	External Event from Front-panel Trigger In
9	TTLT	External Event from TTL Trigger Lines
8	LimV	Limit Violation

Table 5-2. Sources of Interrupts

Reserved / Logical Address ***Bits 7 - 0***

During a read, these bits return all "1's." During an interrupt acknowledge cycle, these bits return the V207's Logical Address.

Interrupt Control Register *1Ch*

The read/write register contains mask bits for each of the eight interrupt sources, a bit for disabling of interrupts and three bits that determine interrupt level. The Interrupt Source chart above describes the mask bits.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	TC*	MBF4*	MBF3*	MBF2*	MBF1*	FPT*	TTLT*	LimV*	EN*	1	Interrupt Req. Level			1	1	1
	Interrupt Mask bits															

Interrupt Mask Bits ***Bits 15 - 8***

Writing a "**1**" one of these bits *prevents* the corresponding interrupt source from generating an interrupt request. Writing a **"0"** enables an interrupt source to generate an interrupt request.

Interrupt Enable ***Bit 7***

Writing a "**1**" to this bit disables interrupt generation. Writing a **"0"** enables interrupt generation.

Interrupt Request Level ***Bits 5 - 3***

These bits determine the interrupt request level.

Bits 5	Bits 4	Bits 3	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

Table 5-3. Interrupt Request Levels

Not Used *Bits 2 – 0*

Bits 2 – 0 are not used and return **"1"**s when read.

Subclass Register *1Eh*

This read-only register provides information about the Subclass of the VXIbus device.

	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	1	1	1	1	1	1	1	0
	Extended Device	**Extended Register-based Device**														

Bit 15 indicates that the V207 is a VXIbus-defined Extended Device.
Bits 14 through 0 indicate that the V207 is an Extended Register-based Device.

Suffix Register *20h, 22h*

The Suffix read-only register (high and low words) hold the ASCII codes for the four characters of the V207's suffix. The suffix defines the optional characteristics of the module.

20h

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

ASCII code for Z = 5Ah (1st character) **ASCII code for A, B, C or D = 41h, 42h, 43h or 44h (2nd character)**

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	x	x	0	0	1	1	0	0	x	x

ASCII code for 1, 2 or 3 = 31h, 32h or 33h (3rd character) **ASCII code for 1, 2 or 3 = 31h, 32h or 33h (4th character)**

The suffix options for the V207 are:

1st character: Digitizing rate
"Z" indicates the maximum digitizing rate is 500 ksamples / second.
For the V207, there are no other maximum-digitizing-rate options

2nd character: Digital Output option
"A" indicates that the Ping-Pong buffer is the only output option present.
"B" indicates that the Multi-buffer is a *linear* buffer; i.e., the Multi-buffer and transient capture option is present *without* pretrigger capability.
"C" indicates that the **Digi-bus** option is present.
"D" indicates that the Multi-buffer is a *circular* buffer; i.e., the Multi-buffer and transient capture option is present *with* pretrigger capability.

3rd character: Multi-buffer capacity
"1" indicates that the Multi-buffer capacity is 1 Mbyte.
"2" indicates that the Multi-buffer capacity is 4 Mbytes.
"3" indicates that the Multi-buffer capacity is 16 Mbytes.

4th character: Revision level
The number, **"1," "2" or "3,"** etc., gives the revision level.

User Defined Registers *24h - 3Eh*

Fourteen read/write registers allow you to store any data you wish. The registers are EEPROM, so the data will not disappear until you over-write it.

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

Note that when a register is written, a minimum of 3 ms must pass before *any* User-Defined register may be read or written.

Operational Registers in A32 Space

The following operational registers are in **A32** space beginning at a starting address specified by the Offset register in **A16** configuration space. The size of the block of memory is either 16 or 32 MB, depending on the chosen V207 options. The hexadecimal addresses shown are the offset from the starting address. You can access these registers by **D16** transfers only, except for the Multi-buffer memory, which can be accessed by **D16** or **D32** transfers.

Sample Clock Register *00h*

This read/write register selects the clock source and sets the internal clock frequency.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	1	1	1	1	1	1	1	1	Sync	Enable	Clock Source		Internal Clock Frequency			

Sync *Bit 7*

You can use this bit to synchronize sample clocks in multiple V207s. Setting this bit to **"1"** enables *Sync* mode. In *Sync* mode, with Enable reset to **"0"** a trigger will clear the clock generator's internal counters. If Enable is set to **"1,"** a trigger will synchronously start the internal counters, *providing that the trigger event is synchronized to the 10 MHz clock*.

Enable *Bit 6*

Set this bit to **"1"** to enable the sample clock. Note that if the *Sync* bit also is set to **"1,"** the sample clock will not start until the next trigger (selected by Trigger Register 1 at 02h).

Clock Source *Bits 5 & 4*

You can select the Sample Clock from one of three sources: Internal Clock Source, VXIbus Trigger Lines, or External Clock supplied through the Front-panel SMB connector. If you select the internal sample clock, the lowest four bits in this register will determine the clock frequency. If you select Trigger Line source, the contents of Trigger Register 1 determine *which* trigger line you select. The Front-panel Clock Input is TTL-compatible. Setting the enable bit to "**1**" enables the selected clock.

Bits 5	Bits 4	Clock Source
0	0	Internal
0	1	Trigger Line
1	0	External (via Front Panel)
1	1	NOT VALID

Table 5-4. Clock Sources

Internal Sample Clock Bits 3 - 0

Bits				Clock Frequency
3	2	1	0	
0	0	0	0	500 kHz
0	0	0	1	200 kHz
0	0	1	0	100 kHz
0	0	1	1	50 kHz
0	1	0	0	20 kHz
0	1	0	1	10 kHz
0	1	1	0	5 kHz
0	1	1	1	2 kHz
1	0	0	0	1 kHz
1	0	0	1	500 Hz
1	0	1	0	200 Hz
1	0	1	1	100 Hz
1	1	0	0	NOT VALID
1	1	0	1	NOT VALID
1	1	1	0	NOT VALID
1	1	1	1	NOT VALID

Table 5-5. Internal Sample Clock Frequency Selection

Trigger Register 1 (Front panel & Sample clock) 02h

This read/write register selects up to four VXIbus trigger lines for transmitting the following:

- event triggers from a VXIbus trigger line to the front panel Trigger-Out connector (bits 15 – 12);
- event triggers from the front-panel Trigger-In connector to a VXIbus trigger line (bits 11 – 8);
- the sample clock from a VXIbus trigger line to this V207 (bits 7 – 4);
- the sample clock from this V207 to a VXIbus trigger line (bits 3 – 0).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Trig In Enable	Event Trigger In			Trig Out Enable	Event Trigger Out			Clock In Enable	Sample Clock In			Clock Out Enable	Sample Clock Out		

Trigger In Enable Bit 15

Setting this bit to "**1**" allows an event trigger occurring on the selected VXIbus trigger line (A) to appear at the front-panel *Trigger Out* SMB connector and (B) to assert Interrupt Source 1. Setting this bit to "**0**" prevents either of these from happening.

Event Trigger In ***Bits 14 - 12***

These bits select a VXIbus trigger line that, when enabled, is routed logically to the front-panel SMB connector. VXIbus trigger lines are selected as follows for all functions that use the VXIbus trigger lines:

Bits			VXIbus Trigger Line
n	n-1	n-2	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Table 5-6. VXIbus Trigger Line Selection

Trigger Out Enable ***Bit 11***

Setting this bit to "**1**" allows an event trigger occurring on the front-panel *Trigger In* SMB connector to appear at the specified VXIbus trigger line. The front-panel event trigger asserts Interrupt Source 2 independent of the state of this bit. Setting this bit to "**0**" prevents the front-panel trigger from being routed to the selected VXIbus trigger line.

Event Trigger Out ***Bits 10 - 8***

This field selects the VXIbus trigger line, as above.

Clock In Enable ***Bit 7***

Setting this bit to "**1**" allows the clock on a VXIbus trigger line to be used as the sample clock if the Sample Clock register selects a trigger line as the clock source.

Sample Clock In ***Bits 6 - 4***

This field selects the VXIbus trigger line to be used if the Sample Clock register selects trigger-line source.

Clock Out Enable ***Bit 3***

Setting this bit to "**1**" outputs the V207 clock source to the selected VXIbus trigger line.

Sample Clock Out ***Bits 2 - 0***

This field selects the VXIbus trigger line to receive the clock source specified in the Sample Clock register. The clock source in the V207 in this way is made available to other system elements.

Trigger Register 2 *04h*

This read/write register allows a Limit Violation (LV) event to drive a trigger line and allows trigger lines to control transient data capture (TC).

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Not Used				LV Enable	LV Select			Stop	Stop Multi-buffer Input			TC Enable	TC Select		

LV Enable ***Bit 11***

When set, transmits Limit Violation status to a selected VXIbus trigger line.

Limit Violation ***Bits 10 - 8***

Selects the VXIbus trigger line that transmits the Limit Violation status.

Stop ***Bit 7***

Not used in current V207 versions.

Stop Multi-buffer Input ***Bits 6 - 4***

Not used in current V207 versions.

TC Enable ***Bit 3***

Setting this bit to **"1"** enables the selected VXIbus trigger line to start post-trigger countdown for transient data capture in the Multi-buffer. *If this bit is set, the Stop bit (bit 7) must be set to zero for proper operation.*

TC Select ***Bits 2 - 0***

Selects the VXIbus trigger line that starts post-trigger (PT) countdown for transient data capture.

Setup Register *06h*

V207-.A.., -.B.. & -.C..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	**Not Used**								Not Used				Global LV Enable	Data Stop/ Start	Transient Mode	Run/ setup

V207-.D..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	**Not Used**								Transient trigger Enable	Not Used			Global LV Enable	Post-trigger Start	Multi-buffer Start	Run/ setup

Transient Trigger Enable Bit 7

V207-.D..

Setting this bit to **"0"** disables using a VXIbus trigger line to trigger transient data capture.

Setting this bit to **"1"** enables the start of post-trigger countdown for capture of transient data. Countdown then starts when an event trigger occurs on a pre-selected VXIbus trigger line. The trigger-line selection is made in Trigger Register 2, bits 2 – 0. Bit 3 (TC

Enable) in Trigger Register 2 (04h) *must be set after* bit 7 in the Setup register (Transient Trigger Enable) is set.

V207-.A.., -.B.. & -.C..

Not applicable

Global LV Enable Bit 3

V207-.B.. & -.D..

Setting this bit to **"0"** disables limit checking.

Setting this bit to **"1"** enables limit checking. A limit violation triggers start of post-trigger countdown.

V207-.A.. & -.C..

Not applicable

Post-trigger Start Bit 2

V207-.B..

Setting this bit to **"0"** halts the storing of data in the multi-buffer (which is linear in the **-.D..** version).

Setting this bit to **"1"** starts the storing of data in the multi-buffer.

V207-.D..

Setting this bit to **"0"** causes no action to occur.

Setting this bit to **"1"** starts post-trigger countdown for transient data capture. This bit should be set only after you have configured all Multi-buffer registers and the module is in *Run* mode. This bit is the "software trigger" for post-trigger countdown.

V207-.A.. & -.C..

Not applicable

Multi-buffer Start Bit 1

V207-.B..

Setting this bit to **"0"** places the V207 in *Normal* (non-transient) mode.

Setting this bit to **"1"** places the V207 in *Transient* mode. In this mode, the linear multi-buffer automatically stops filling when it is full.

V207-.D..

Setting this bit to **"0"** halts the storing of data in the circular multi-buffer.

Setting this bit to **"1"** starts the storing of data in the circular multi-buffer.

V207-.A.. & -.C..

Not applicable

Run/setup Bit 0

V207-.A.., -.B.., -.C.. & -.D..

Setting this bit to **"0"** places the V207 in *Setup* mode. In *Setup* mode you may configure, or examine the contents of, the Scan RAM and Limit Memory.

Setting this bit to **"1"** places the V207 in *Run* mode. In *Run* mode, current data is available in the Ping-Pong buffer and on the Digi-bus. Data availability in a multi-buffer depends upon other bits in this register.

Channel Limit-Violation Register 08h

This read-only register indicates which channel (0 to 255) has encountered a limit violation.

	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	Limit Violation Channel Number

Channel Number ***Bits 7 - 0***

Total Samples-Per-Frame Register (Digi-bus option) *10h*

This write-only register specifies the total number of time slots (one sample per time slot) in the **Digi-bus** frame. The maximum number of time slots is 256. The number in the register is one less than the number of time slots. This includes all samples from *every **Digi-bus** source module* in this set. If there is only one **Digi-bus** source module in this set, this value should equal the samples-per-frame value at offset 12h. If there are multiple **Digi-bus** source modules, the Total Samples Per Frame register on each module must be set to the same value.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	Not used								Total Number of Samples per Frame from all modules							

Samples per Frame ***Bits 7 - 0***

Samples-Per-Frame from this Source Register (Digi-bus option) 12h

This write-only register specifies the number of time slots occupied by *this* **Digi-bus** source module within the **Digi-bus** frame. Setting the value to zero specifies one sample/frame. This register contains the number of active channels on this module minus one.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	Not Used							Strobe Enable	Number of Samples per Frame from this module							

Strobe Enable Bit 8

When set, **Digi-bus** timing strobes are enabled onto the Local Bus.

Samples per Frame Bits 7 - 0

Start Address Register (Digi-bus option) 14h

This write-only register specifies the Starting Time Slot within a **Digi-bus** frame for *this* **Digi-bus** source module to begin inserting its data. A value of 0 specifies the first location within the frame.

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

Starting Time-Slot Bits 7 - 0

Digi-bus Example 1 - One source module

One V207 with **Digi-bus** option with 4 active channels

V207 registers	Data
Total Samples Per Frame (10h)	3
Samples Per Frame from This Source (12h)	3
Start Address (14h)	0

Digi-bus Example 2 - Two source modules

One V207 with **Digi-bus** option with 4 active channels
One V208 with **Digi-bus** option with 2 active channels

V207 registers	Data
Total Samples Per Frame (10h)	5
Samples Per Frame from This Source (12h)	3
Start Address (14h)	0
V208 registers	**Data**
Total Samples Per Frame (10h)	5
Samples Per Frame from This Source (12h)	1
Start Address (14h)	4

Total Buffer-Size Register (Multi-buffer option) 20h, 22h

The Total Buffer-Size register (high word and low word) holds the 32-bit value that represents the buffer size you need for your application. To accommodate **D32** reads, the buffer is 32 bits wide with each "long word" holding 2 samples. The value stored is actually one less than the

number of long words needed, i.e., (# of samples)/2 – 1. For example, the value "0" indicates one 32-bit long-word (two samples); the value "FFFh" indicates 4096 long-words (8192 samples); and the value "3FFFFFh" indicates 4,194,304 long-words (8,388,608 samples or 16 Mbytes).

20h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	**Total Buffer Size – Low Word**															

22h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	**Total Buffer Size – High Word**															

Individual Buffer-Size Register (Multi-buffer option) 24h, 26h

The Individual Buffer-Size register (high word and low word) holds the 32-bit value that represents the number of 32-bit long words (each holding two data samples) desired in *each segment* of the Multi-buffer memory. As above, the value stored is actually *one less* than the number of long words needed, i.e., (# of samples)/2 – 1.

The values you store in the Total- and Individual Buffer-Size registers must be consistent with the number of segments you desire to have. For example, if you wish to store a total of four Msamples using four segments, the Total Buffer-Size register must hold the value "1FFFFF" (2,097,152 long words) and the Individual Buffer-Size register "7FFFF" (524,288 long words, or 1 Msample). If you wished to have eight segments in this example, the Individual Buffer-Size register would hold the value "3FFFF."

24h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	Individual Buffer Size – Low Word															

26h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Write-only	Individual Buffer Size – High Word															

Buffer-Full Flag Register (Multi-buffer option) *28h*

You can configure the Multi-buffer memory with up to eight segments, as specified in the Total Buffer and Individual Buffer Size registers. Each segment has an associated status bit called a Full flag which indicates when the segment is full. It is necessary to clear each flag bit as the data is read from the corresponding buffer. Failure to clear the flag bit will result in the Overrun bit being set. *Note that write operations to this register result in selective clears, such that writing a **"1"** to a bit position resets that bit to **"0."***

V207-.B..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / selective clear	Not Used								Not Used			Overrun	Full4	Full3	Full2	Full1/ Transient Complete

V207-.D..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / selective clear	Transient Complete	Not Used						Overrun	Full8	Full7	Full6	Full5	Full4	Full3	Full2	Full1

Transient Complete *Bit 15 (Bit 1 for V207-.B..)*

The Transient Complete bit is set to **"1"** when the transient acquisition is complete; i.e., the post-trigger countdown reached zero. This bit resets to **"0"** when the V207 is placed into "Run" mode *and* the "Multi-buffer Start" bit is set to **"1"** (see Setup register, at offset 06h).

Overrun *Bit 8 (Bit 4 for V207-.B..)*

The Overrun bit indicates a buffer has been over-written before its contents were read. Writing **"1"** to bit 8 results in a selective clear of that bit.

Buffer Full Flags *Bits 7 - 0 (Bits 3 – 0 for the V207-.B..)*

When each buffer becomes full it asserts its Full flag. Writing a **"1"** to one or more bit positions results in a selective clear of those bits.

Countdown Register (Multi-buffer transient option) 30h, 32h

The Countdown registers are written with a 32-bit value corresponding to desired number of ″Post Trigger″ Samples/Channel. These registers are used when using the Multi-buffer in the Transient mode. The number of pre-trigger samples equals the buffer size minus the number of post-trigger samples.

30h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Post-trigger Count – Low Word															

32h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Post-trigger Count – High Word															

Trigger Address Register (Multi-buffer transient option) 34h, 36h

The Trigger Address registers contain the memory address at the time the post-trigger countdown is initiated. This address points to the data for the first channel in the Scan List.

34h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	Memory Address Pointer – Low Word															

36h

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	Memory Address Pointer – High Word															

Scan RAM 200h - 3FEh

The Scan RAM contains the scan table which must be set up properly on all **MUX-bus** modules. The Scan RAM is 256 words long, and the Scan Table is a subset of the Scan RAM. Each module scans through its Scan Table sequentially at the ADC clock rate. The currently addressed data on the list specifies which module is to connect a channel to the **MUX-bus**.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	End of List	Module Select	Not Used						Channel Numbers							

End of List Bit 15

You set this bit to a "**1**" to signify the end of the list.

Module Select Bit 14

This bit is not actually used in the V207. However, you use this bit the in signal-conditioning MUX modules to indicate which module applies its analog channel to the **MUX-bus** in that time slot.

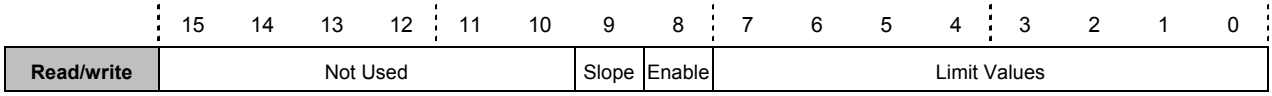
Channel Numbers Bits 7 - 0

You use these bits to define the active channels and the order of scanning.
Before you can write the Scan Table into Scan RAM, you must put the V207 in "Setup" mode. The V207 (or any other module receiving the MUX-bus) should be the first device put in "Setup" mode and the last put in "Run" mode. The Scan RAM is organized the same on all modules, with the exception of the Module Select bit. Bits 0 - 7 specify the channel number (0-255). Bits 8 - 14 are not used in the V207 Scan RAM and should be filled with zeroes. Bit 15 is the End-of-List bit. This bit should be set in the last location of the Scan Table (i.e., the last active channel) on all MUX-bus modules.
Scan RAM reflects the list of active channels for a given application. There can be up to 256 ADC time slots for each period of the sample clock. The channels, each representing an ADC time slot, are multiplexed into four paths, with the channels being assigned to the paths as follows (Note that physical Channel 1 is defined as 0000 0000 in Scan RAM.):
MUX-bus Path A -- Channels 1, 5, 9…
MUX-bus Path B -- Channels 2, 6, 10…
MUX-bus Path C -- Channels 3, 7, 11…
MUX-bus Path D -- Channels 4, 8, 12…
This physical mapping of the channels causes certain restrictions on the organization of the Scan RAM to attain maximum system performance. The Scan RAM must be organized to allow acquiring consecutive samples from Path A, Path B, Path C, and Path D respectively. For example, consecutive entries in the Scan RAM should appear as follows:

Address	Data (bits 7 - 0)	Comment
200h	xxxx xx00	These channels are assigned to MUX-bus path A.
202h	xxxx xx01	These channels are assigned to MUX-bus path B.
204h	xxxx xx10	These channels are assigned to MUX-bus path C.
206h	xxxx xx11	These channels are assigned to MUX-bus path D.
208h	xxxx xx00	Repeat back to MUXbus path A, etc.

Limit Memory *400h - 5FEh*

The Limit Memory contains the data for limit checking (using level and slope) in the V207. Bits 7-0 contain the level that is compared against the ADC data bits 15-8 respectively. Bit 8 is a channel-by- channel enable and should be set to a **"1"** on all channels in which limit checking is desired. Bit 9 should contain the desired Slope direction. A one-to-one mapping is present between the Scan RAM and Limit Memory where the limit conditions specified in address 0 of the Limit Memory are associated with the Channel Number as specified in address 0 of the Scan RAM.



Slope *Bit 9*

Slope Selection bit

0 Negative slope
1 Positive slope

Enable *Bit 8*

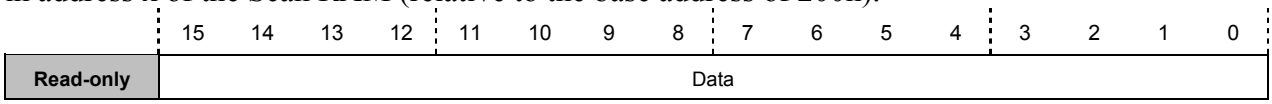
When set, enables limit checking

Limit Values *Bits 7 - 0*

These bits specify the Limit Values.

Ping-Pong Memory *600h - 7FEh*

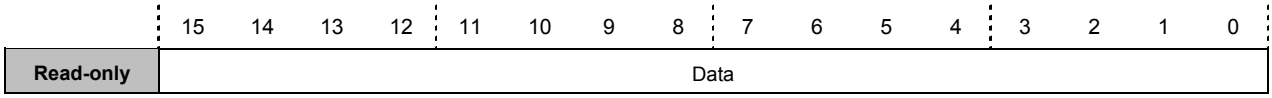
Ping-Pong memory provides a double buffering scheme for all active channels. The data bits in these RAM locations represent the active-channel data from Sample Clock (n-1). There is a one-to-one mapping between the Scan RAM and Ping-Pong memory. The data in Ping-Pong RAM address x (relative to the base address of 600h) is associated with the channel number specified in address x of the Scan RAM (relative to the base address of 200h).



The Ping-Pong memory only supports **D16** (16-bit) transfers. A **D32** transfer will contain only 16 bits of data.

Multi-buffer Memory (optional)

1 MB option	***10 0000h – 1F FFFEh***
4 MB option	***40 0000h – 7F FFFEh***
16 MB option	***100 0000h – 1FF FFFEh***



The Multi-buffer memory supports both **D16** and **D32** transfers. A **D16** transfer will read one 16-bit word, while a **D32** transfer will read two 16-bit words.

NOTE: When using multibuffer memory, only an even number of channels is allowed.

Chapter 6: Programming Information

Example procedures that use the Multi-buffer to capture transient data

You can use either the V207-.B.. or the V207-.D.. to capture transient data. Part of the operational setup is identical for the two V207 versions. However, there are also some differences, since the V207-.B.. uses a linear buffer and the V207-.D.. uses a circular buffer to store the transient data. The following procedures illustrate the similarities and differences. The procedures illustrate the capture of 2 ksamples of transient data in each of 4 channels at a sampling rate of 50 kHz per channel. A software command initiates the data capture.

<table>
<tr><th>Using a V207-.B..
containing a 4 MB linear Multi-buffer</th><th>Using a V207-.D..
containing a 16 MB circular Multi-buffer</th></tr>
<tr><td colspan="2">Get the base address for the V207 Operational registers, which are all in A32 space.
Read the Offset register at 06h (A16 space) to get the A32 offset.
Logical shift the result 16 places to the left to get the V207 base address.</td></tr>
<tr><td colspan="2">The registers from here on are all in A32 space.
Select the clock frequency and start the clock running.
Write 0043h to the Sample Clock register at 00h.</td></tr>
<tr><td colspan="2">Put the V207 in Setup mode.
Write 0000h to the Setup register at 06h.</td></tr>
<tr><td colspan="2">Load the Scan RAM.
Write 0000h to 200h.
Write 0001h to 202h.
Write 0002h to 204h.
Write 8003h to 206h.</td></tr>
<tr><td colspan="2">Load the Total Buffer-Size register. The value = total samples/2 – 1 = 4*2048/2 – 1 = 4095 = FFFh.
Write 0FFFh to the low word and 0000h to the high word of the Total Buffer-Size register at 20h and 22h, respectively.</td></tr>
<tr><td colspan="2">Load the Individual Buffer-Size register. The value = the above value + 1 = 4096 = 1000h.
Write 1000h to the low word and 0000h to the high word of the Individual Buffer-Size register at 24h and 26h, respectively.</td></tr>
<tr><td></td><td>Load the Countdown register with the number of post-trigger samples desired. If in this example you want 75% pre-trigger and 25% post-trigger samples, then the value</td></tr>
</table>

	loaded is 512 (= 200h). Write 0200h to the low word and 0000h to the high word of the Countdown register at 30h and 32h, respectively.
Put the V207 in *Run* mode and enable *Transient* mode. Write 0003h to the Setup register at 06h.	**Put the V207 in *Run* mode and begin acquiring data into the circular buffer.** Write 0003h to the Setup register at 06h.
Initiate transient data capture. Write 0007h to the Setup register at 06h. Data capture stops automatically when the buffer is full; the V207 remains in *Run* mode.	**Initiate transient data capture.** Write 0007h to the Setup register at 06h. Data capture stops when the post-trigger samples are captured. The V207 remains in *Run* mode.
Determine when the transient capture has ended. Poll on bit 0 of the Buffer-Full Flag register at 28h. Continue when set.	**Determine when the transient capture has ended.** Poll on bit 15 of the Buffer-Full Flag register at 28h. Continue when set.
	Determine where in A32 address space the data is located. Read the low and high words of the Trigger Address register at 34h & 36h, respectively. The 32-bit value is the Multi-buffer address of the data from the first channel in the Scan list. Since there are 2 ksamples of post-trigger data and 6 ksamples of pre-trigger data, the *offset* to the beginning of pre-trigger data is the value read minus 1800h.
Read the data. With a 4-Mbyte linear Multi-buffer, the data starts at address 0040 0000h. Data is in consecutive locations, with channels interleaved. Read location 40 0000h. Data = Smpl 1, Chnl 1 Read location 40 0002h. Data = Smpl 1, Chnl 2 Read location 40 0004h. Data = Smpl 1, Chnl 3 Read location 40 0006h. Data = Smpl 1, Chnl 4 Read location 40 0008h. Data = Smpl 2, Chnl 1 Read location 40 000Ah. Data = Smpl 2, Chnl 2	**Read the data.** With a 16-Mbyte circular Multi-buffer, the data starts at address *n*, which is 0100 0000h adjusted by the *offset* determined in the previous step. Data is in consecutive locations, with channels interleaved. Read location (n)h. Data = Smpl 1, Chnl 1 Read location (n+2)h. Data = Smpl 1, Chnl 2 Read location (n+4)h. Data = Smpl 1, Chnl 3 Read location (n+6)h. Data = Smpl 1, Chnl 4 Read location (n+8)h. Data = Smpl 2, Chnl 1

Read location 40 000Ch. Data = Smpl 2, Chnl 3 Read location 40 000Eh. Data = Smpl 2, Chnl 4 etc . . . Read location 40 3FFEh. Data = Smpl 2000, Chnl 4	Read location (n+A)h. Data = Smpl 2, Chnl 2 Read location (n+C)h. Data = Smpl 2, Chnl 3 Read location (n+E)h. Data = Smpl 2, Chnl 4 etc . . . Read location (n+3FFE)h Data = Smpl 2000, Chnl 4

Appendices

Appendix A: Differences between the V207–.B.. and the V207–.D..

Introduction

The "B" and "D" versions of the V207 have multi-buffers that support both transient and continuous data acquisition. Most operational parameters are identical in these two versions. However, some significant differences exist that affect the number of operational registers needed as well as the bit definitions.

The major differences are (1) multi-buffer size, (2) number of multi-buffer segments, and (3) pre-trigger capability, as shown in the following table:

Version	Multi-buffer size	Number of multi-buffer segments	Pre-trigger capability
V207–.B..	Choice of one or four Mbytes.	Four.	No pre-trigger; only post-trigger. Buffer is linear.
V207–.D..	Choice of four or sixteen Mbytes.	Eight.	Pre-trigger and post-trigger. Buffer is circular.

The D version requires two additional 32-bit registers (compared with the B version): the Countdown register at offsets 30h & 32h and the Trigger Address register at offsets 34h & 36h. One needs these additional registers because of the pre-trigger capability of the D version. The Countdown read/write register holds the desired number of *post-trigger* samples/channel. The Trigger Address read-only register contains the multi-buffer memory address at the start of post-trigger countdown. Specifically, the address points to head-of-the-scan-list sample data that immediately follows the capture trigger.

Bit definitions for the B and D versions are not identical for the Setup registers (at offset 06h) and the Buffer-Full Flag registers (at offset 28h). You can refer to the following two pages, as well as the programming example in Chapter 6, for details.

Setup Register (06h)

V207-.B..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	**Not Used**								Not Used				Global LV Enable	Transient capture Start	Tran-sient Enable	Run/ setup

Bit		Definition	
0	Run/Setup	0	Setup mode. Used to access Scan RAM and Limit RAM.
		1	Run mode. Data is available in Ping-Pong buffer and on Digi-bus.
1	Transient Mode Enable	0	Normal multi-buffer mode.
		1	Transient mode. Data automatically stops filling the multi-buffer when the multi-buffer is full.
2	Transient Capture Start	0	Stops storing of data in the multi-buffer.
		1	Starts storing of data in the multi-buffer.
3	Global Limit Enable	0	Disables limit checking.
		1	Enables limit checking. A limit violation triggers the start of transient capture.

V207-.D..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	**Not Used**								Transient trigger Enable	Not Used			Global LV Enable	Post-trigger Start	Multi-buffer Start	Run/ setup

Bit		Definition	
0	Run/Setup	0	Setup mode. Used to access Scan RAM and Limit RAM.
		1	Run mode. Data is available in Ping-Pong buffer and on Digi-bus.
1	Multi-buffer Start/Stop	0	Stops storing data in the circular multi-buffer.
		1	Starts storing data in the circular multi-buffer.

2	Post-trigger Capture Start	0	No action.
		1	Starts post-trigger countdown for transient data capture.
3	Global Limit Enable	0	Disables limit checking.
		1	Enables limit checking. A limit violation triggers the start of transient capture.
7	Transient Trigger Enable	0	Disables using a VXIbus trigger line to trigger transient data capture.
		1	Enables using a VXIbus trigger line to trigger transient data capture.

Buffer-Full Flag Register (28h)

V207-.B..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / selective clear	Not Used								Not Used			Overrun	Full4	Full3	Full2	Full1 / Transient Complete

V207-.D..

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read / selective clear	Transient Complete	Not Used						Overrun	Full8	Full7	Full6	Full5	Full4	Full3	Full2	Full1

The differences between the “B” and “D” versions of the Buffer-Full Flag register are as follows:

1. The “B” version has four multi-buffer segments and associated Full flags; the “D” version has eight.
2. The Overrun bit is bit 4 in the “B” version and bit 8 in the “D” version.
3. For transient data capture, the Transient Complete indicator is bit 0 in the “B” version and bit 15 in the “D” version.

Appendix B: Differences—Earlier versions

The last digit in the product number gives the version number. The current version number for the V207 is "2."

The differences between the V207-…2 and V207-…1 are as follows:

- The V207-.B.1 and V207-.D.1 provide only 16-bit (**D16**) transfers over the VXIbus while the V207-.B.2. and V207-.D.2 provide both 16- and 32-bit (**D32**) transfer capability.

Appendix C: Data Buffering Methods

Introduction

The objective in many data acquisition applications is to accurately measure the time history of one or more signals. The V200, V207, V208 and V213 provide internal programmable *sample clocks* that initiate a scan of all channels associated with the ADC on a precise time interval. Thus, sampled data for each channel is associated with the sample clock. These modules provide several methods to move sampled data, as shown in the following table:

Readout Method	Data Access	Availability
Ping-Pong memory	via VXIbus	standard on all of the above modules
Multi-buffer memory	via VXIbus	an option all of the above modules
Digi-bus	via the VXI Local bus	an option on the V207 / V208

This appendix discusses the various readout methods, including their features, limitations and guidelines for general application. The results in a particular system depend on a wide range of factors. Please contact the KineticSystems technical staff if you have questions regarding a particular application.

Ping-Pong Memory

The Ping-Pong memory provides a double-buffering method for acquiring data. You access this two-segment memory from the VXIbus. One segment contains data values for all active channels from the previous "tick" of the sample clock, while the other segment is receiving data values from the current "tick" of the sample clock. You must read all the data from one segment via the VXIbus before it begins filling again, and then do the same for the other…and so forth.

Multi-buffer Option

The optional Multi-buffer memory acts as a large circular buffer for all channel data. It has two operating modes—*continuous* and *transient*. When operated in continuous mode, it can buffer *n* scans of the sample clock. In many systems, this effectively overcomes the effects of system overhead by amortizing it over many data acquisition scans. When used in transient mode, the large circular buffer stores pre-trigger and post-trigger samples. One defines the size of the circular buffer during setup. In transient mode the data is read via the VXIbus after the recording is complete, placing fewer restrictions on the effects of system throughput. In -wByz versions of the V207 and V208 modules, the buffers are not circular but linear in transient modes and do not have pre-trigger capability.

Digi-bus Option

The Digi-bus option uses the VXI Local bus for movement of data. Modules that support the Digi-bus protocol to receive digital data must be located in slots immediately to the left of the one or more ADCs generating the main stream of data. Typical receiving devices are DSP and RAM modules. While limited to the use with data source and sink modules that support this protocol, the Digi-bus advantage is that data transfers are immune to VXIbus latency issues.

Acquiring Data from the Ping-Pong Memory

For applications with slower aggregate sampling rates (sample clock rates), the Ping-Pong memory serves as a simple method of acquiring data. In such applications an interrupt or trigger line transition serves as a signal to the processor to read the data from Ping-Pong memory. Since this memory only buffers a single data scan, it is incumbent on the application software to read the data from the memory between sample clock events. Thus, for a sample clock rate of 10

Hertz, the application software must read all of the data from the Ping-Pong memory during each 100 ms time interval defined by the sample clock.

Typical applications frequently use a computer with a multi-tasking kernel to read data from the Ping-Pong memory. When using this technique, the user must insure that the application task that reads the data can respond to the sample-clock event and read the data within one sample-clock period; otherwise data samples will be lost.

With today's multitasking operating systems, the data acquisition task is not always guaranteed to respond to an event and read the data within a fixed time interval, even if it is the only user-selected task running. This is because multiple tasks may still be competing for the system resources. In many cases these are system tasks or drivers responding to interrupts; both of these classes of tasks have higher priorities than a user task. In some cases, such as in systems operating under Windows 95, the user has little or no control over tasks. Even in operating systems where there is significant control (e.g. the VxWorks real-time kernel), it is difficult to prevent a low-priority task, for example, from issuing an I/O request that subsequently results in a very-high-priority driver I/O interrupt being handled. This is known as blocking.

The overriding issue is that a user data acquisition task will experience a variable latency in responding to an event (interrupt) and subsequently reading data. This delay is highly dependent on processor speed and configuration as well as on the other events that are pending or occurring during the period from the data acquisition event until the read request is satisfied.

In the case of data acquisition, loss of a single data scan can be quite undesirable. To prevent data loss it is necessary that the *worst case* process latency plus data transfer time be less than the sample-clock interval when using Ping-Pong memory. Worst case latency is a very difficult number to determine. As a general rule one can service the Ping-Pong memory reliably below 10 Hz. Somewhere between 10 Hz and 1 kHz, one will start to encounter excessive data loss. Above 10 kHz reliable data acquisition can only be achieved under highly controlled situations with dedicated processors using no operating systems or ones that are highly deterministic and exhibit low overhead.

Higher Ping-Pong sample rates can be achieved by the use of the KineticSystems Grand Interconnect (GI). If the V160 GI Slot-0 controller contains the Multi-buffer option, that buffer can be used to receive data from the ADC's Ping-Pong memory over VXIbus, with the data transfer being isolated from any issues related to the host computer's operating system. Transactions between the V160 and the host computer system are similar to those for an ADC with Multi-buffer.

Acquiring Data from the Multi-buffer Memory—Continuous Mode

In the previous section we have discussed the problems caused by computer latency in response to an interrupt. For example, assume a 20 millisecond worst-case computer latency in a 50-channel data acquisition system with a sample rate of 100 Hz. If we use the Ping-Pong buffer, a 50-word block of data is available every 10 milliseconds (a 100 Hz rate). Also assume that a block of data is transferred from the buffer to the host at a rate of 2 megabytes/s (0.001 millisecond per 16-bit word). It follows that data will be lost whenever the latency plus the time to transmit the data block exceed the 10 millisecond sample rate. For this example, if the worst-case latency occurs during data transfer, the resulting total latency + transfer time will be:

20 ms + (50 x 0.001 ms) = 20.05 ms

If we use the Multi-buffer to buffer multiple data scans, the effects of computer latency can be greatly reduced. If we take the previous example, but store 10-scan blocks of data (10 x 50 16-bit words = 500 words) in the Multi-buffer, we get the following results:

20 ms + (500 x 0.001 ms) = 20.5 ms

The total time (worst-case latency + data transfer time) has increased slightly (20.5 ms instead of 20.05 ms), *but we now have a 100 millisecond window (10 scans x 10 ms/scan) before the next block of data needs to be transferred.* In a particular system the maximum throughput that can be achieved is affected by the computer latency, the number of channels per scan, the sample rate, the number of scans in a Multi-buffer block, the block data transfer rate to the host system, and—if the data is to be transferred to disc storage in real time—the storage latency and transfer rate.

A consideration of buffering is that, if we wish to monitor the data in quasi-real time, the data available at the processor for monitoring may be delayed by as much as the buffering factor, and on average by the *buffering factor*/2. This can be partially overcome by subdividing the Multi-buffer into multiple segments. The standard V207/V208 Multi-buffer allows up to eight segments. The processor is interrupted when a segment is filled and the data is read. The segment size, however, can now be smaller since there are additional segments to fill before losing data, and the average latency is much less than the worst-case latency.

Acquiring Data from the Multi-buffer Memory—Transient Mode

For applications that involve the recording of data over a finite time period, transient mode can be the most desirable buffering method. For this mode the size of the Multi-buffer must be equal to—or greater than—the aggregate number of 2-byte samples to be recorded. For example, if 100 channels are to be recorded at 1000 samples/s per channel for 10 seconds, the total buffer size (in bytes) must be:

2 x 100 x 1000 x 10 = 2,000,000 bytes

Therefore a 4 Mbyte (4,194,304 byte) Multi-buffer memory would be sufficient for this application. Note that the Total Buffer Size register in the V207 or V208 sets the actual circular buffer size and represents the number of 32-bit words (4-byte groups) to be transferred - 1. For this example, the data to be loaded in the Total Buffer Size register is:

(2,000,000/4) - 1 = 499,999 (7A11Fh)

To initiate transient recording, data sampling is started, and the synchronous transfer of data to the Multi-buffer memory (circular buffer) within the module begins. When an event occurs, it causes the post-trigger countdown to begin (as the result of a trigger on the ADC front panel, from a VXI trigger line, from a preset limit being exceeded or from software). When the post-trigger countdown is complete, data is available to the host computer. The transfer of data to the host can now occur. Once the data is transferred, recording can be initiated and the cycle repeated.

Since all of the data associated with an event is stored in the Multi-buffer, the movement of data to the host does not limit the maximum aggregate sample rate—assuming that there is sufficient time to perform the data transfer to the host before sampling needs to be started for the next event. Sampling must be started early enough so that the actual recording buffer is filled with *new* data. Otherwise, some of the data at the beginning of the block will be old data from the previous event—or random data if this is the first test after power-on.

If, in the example above, we needed 60% post-trigger (40% pre-trigger) samples, we would load the number of 32-bit words in the V207 or V208 Countdown Register to select the post trigger samples:

0.6 x (2,000,000/4) = 300,000 (493E0h)

Since, in this example, the total recording time is 10 seconds, the time required for all of the pre-event samples to be recorded is 0.4 x 10, or 4 seconds.

Appendix D: About KineticSystems

KineticSystems Company, LLC designs, produces and markets high-performance data acquisition and control systems to a broad range of customers in aerospace, defense, automotive, scientific and other industrial markets.

The Leader in the Delivery of High-performance CAMAC-based Products

The company was founded in 1970 to develop and manufacture interface modules and associated products based on the international CAMAC standard. CAMAC, an acronym for Computer Automated Measurement and Control, is a set of specifications developed by a committee of the government-sponsored research laboratories (the NIM Committee). The committee's goal was to provide standardized modular building blocks for configuring a wide range of data acquisition and control systems. CAMAC, the first open-system real-time input/output (I/O) specification, later was accepted as a standard by the Institute of Electrical and Electronic Engineers (IEEE STD 583). We soon became the recognized leader in delivering high-performance CAMAC-based products. We have retained that leadership position.

Innovation with the CAMAC Serial Highway

As the need for large distributed data acquisition and control systems grew, the NIM Committee produced specifications for the CAMAC Serial Highway to allow communication between a host computer and CAMAC I/O chassis over some distance. In 1975, we delivered the first CAMAC Serial Highway computer interface. We soon were recognized as the leader in the delivery of Serial Highway system components. Serial Highway innovations included a block-mode protocol that increased data throughput by a factor of 10 and fiber-optic highway interfaces that allow the CAMAC chassis to be separated by up to 2 kilometers at full data rate. We continue to be an innovator in the field of high-performance data acquisition.

H•TMS, a Turn-key Testing Solution, Added to Our Product Range

In 1991, we purchased the H•TMS (High-performance Test Management System) product line. H•TMS is a modular set of microprocessor-based hardware and software components that delivers functions usually found in several instruments and a computer. H•TMS increases testing productivity and provides solutions to key problems encountered by test engineers and managers. The addition of this product line provides the answer for customers who need a turnkey testing solution.

A Major Player in VXIbus, a Rapidly Growing Interface Standard

VXIbus is a standard (now IEEE STD 1155) developed by the major instrumentation manufacturers to allow customers to move from chassis-type instruments to computer-interfaced modular building blocks. In 1992 we embraced the relatively new VXIbus standard. Using our extensive CAMAC experience, we soon produced 35 VXI modules for data acquisition and control. As the VXI market has grown to several hundred million dollars, we continue our innovations with additional high-performance products. This includes the development of a set of products called the Grand Interconnect™. These products allow VXI chassis to be distributed on a fiber-optic highway. CAMAC and mixed VXI/CAMAC chassis are also supported. We are an active member of the VXI *plug&play* Systems Alliance, an organization that promotes standards that make VXI easier to configure and to use.

KineticSystems Today

Our headquarters and factory facilities in Lockport, Illinois, have grown to 70,000 square feet. We also have an operation in Englewood, Colorado, and six domestic sales offices, including the Lockport facility. We have distributors in 17 other countries. VXI, CAMAC and H•TMS continue to be our major product lines.

Today, an increasing number of customers are demanding the delivery of data acquisition and control solutions, not just products. For many years we have provided software drivers for our products to make them easier to use. We have an active program to develop VXI *plug&play* instrument drivers for our range of VXI modules. We have developed a high-performance software application program called DAQ Director™. Distributed VXI and/or CAMAC systems can be configured using this software. By using UNIX workstations and powerful I/O control computers, a high I/O performance can be achieved that is not available with any other general-purpose software package. Additionally, to achieve more complete solutions, we provide integration services.

KineticSystems is ideally suited to meet customers' needs as we approach the twenty-first century. We have extensive experience in the design, manufacture and delivery of high-performance data acquisition products and systems. Even though the demand for standards-based products has only recently become a high priority, we have the experience with such products since 1970.

Ways to contact us:

KineticSystems Company, LLC
900 N. State Street
Lockport, IL 60441-2200

Phone: 1-800-DATA NOW (1-800-328-2669)
(815) 838-0005
Fax: (815) 838-4424

E-mail: mkt-info@kscorp.com
tech-serv@kscorp.com
sales@kscorp.com
Web: http://www.kscorp.com

Warranty

KineticSystems 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. KineticSystems 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 KineticSystems carry the original equipment manufacturer's warranty. KineticSystems 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 KineticSystems are prepaid by the purchaser, while charges for returning the repaired product to the purchaser, if located in the United States, are paid by KineticSystems. 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 KineticSystems, and KineticSystems assumes no liability relating to the services provided by the carrier.

The product warranty may vary outside the United States or Switzerland 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. KineticSystems 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 KineticSystems.

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

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

   KineticSystems Company, LLC
   Repair Service Center
   900 North State Street
   Lockport, IL 60441
   Telephone: (815) 838-0005
   Fax: (815) 838-4424

Index

Feedback

The purpose of this manual is to provide you with the information you need to make the V207 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 0095. Or mail a copy to KineticSystems Company, LLC, 900 N. State, 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