Model V208

16-bit, High-speed ADC Subsystem

User's Manual

April 27, 1998

Table of Contents

About this Manual

Organization

Chapter 1, ***Introduction***, gives you a brief overview of the Model V208, 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 V208 and correctly insert it into a C-size VXIbus mainframe.

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

Chapter 4, ***Understanding the V208***, describes the performance of the V208.

Chapter 5, ***Configuration and Operational Registers***, explains how to access and control the V208.

Chapter 6, ***Programming Information***, gives you example setup procedures for preparing the V208 to acquire data.

The ***Appendices*** provide additional information that may be helpful in the use of the Model V208, in learning more about KineticSystems and its products, and in quickly reaching us.

Terms and Conventions

Following are definitions 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 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 Addressing	The VXIbus addressing mode in which the address of a device is stored in a switch register. See also Dynamic Addressing.

Chapter 1: Introduction

About the V208

The V208 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, V243, V246, V252 and V253) to provide flexible system configurations without the need for inter-module cabling.

The V208 provides four instrumentation amplifier inputs to increase throughput with large numbers of input channels. When used with eleven V243 signal conditioning modules, up to 1056 channels can be converted with each channel having its own limit-checking value. If no **MUX-bus** signal conditioning modules are used, the V208 can digitize up to four preconditioned signals that connect to front-panel isolated BNC connectors.

The V208 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. 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 V208 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 V208 is available with sampling rate, **Digi-bus**, and Multi-buffer options. The part number is V208-*wxy*2, where:

w specifies the sampling rate:

Y = 100 ksamples/s
Z = 200 ksamples/s

x selects the digital output option:

A = no option
C = **Digi-bus** option
D = Multi-buffer option

y specifies the Multi-buffer size:

1 = no Multi-buffer
2 = 4 Mbyte Multi-buffer
3 = 16 Mbyte Multi-buffer

A V208 with sampling rate to 200 ksamples/s and 16 Mbyte Multi-buffer is a V208-ZD32, for example.

Getting Started

To set up and use your V208 VXIbus module, you will need most or all of the following:

- The V208, 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
  - V243 - 96-channel, low-level signal conditioner

  - 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 supportsw 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 V208

The V208 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

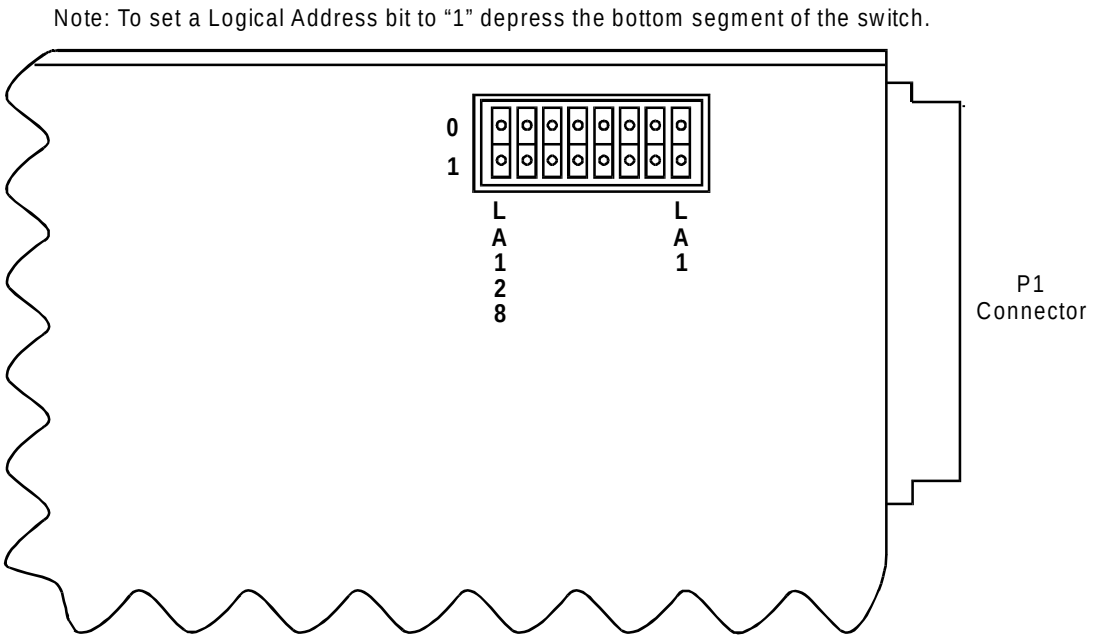
Setting the Logical Address Switches

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 writeable register. For reasons discussed in Chapter 5, all KineticSystems VXIbus devices support dynamic addressing.

Before installing the V208 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 a module is replaced, it is very important that the new module has 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).



V208 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, to assure that the V208 acknowledges interrupts properly, you must position certain jumpers correctly on the chassis backplane. Remove the Interrupt Acknowledge jumper from the slot selected for the V208 and install daisy-chain jumpers in any empty slots between the V208 and the Slot 0 Controller.

You can now insert the V208 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 V208.

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 V208 *configuration* registers. The second deals with setting parameters related to data acquisition and involves communication with V208 *operational* registers.

Bus-related configuration includes setting the logical address, specifying the amount of memory space required, specifying where in memory the V208 registers and memory blocks are located, and setting interrupt levels.

Data-acquisition related configuration for the V208 includes selecting a sample-clock source and clock rate, loading the scan list in Scan RAM, and perhaps selecting Multi-buffer or **Digi-bus** parameters.

VXIbus devices occupy system memory space. Each VXIbus device is allotted 64 bytes of memory space for its configuration registers in the upper 16 kbytes of the 64 kbyte **A16** memory space. The 8-bit Logical Address, whether set statically by switch register or dynamically by a writeable register, determines the base address of the 64-byte block of memory.

The bit pattern for configuration register addresses is:

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

The 64-byte block contains several registers that supply information about the module, such as the manufacturer, the module identifier (i.e., "208h"), 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. Depending on the option chosen, the V208 requires either 16 or 32 Mbytes of memory space. The V208 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. Please refer to Chapter 5 for details relating to the *configuration* and *operational* registers.

Chapter 3: Understanding the V207/V208 Family

Overview

The V208 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 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 V208 transmits its digitized signals via the VXIbus backplane or to one or more companion modules via the Local Bus using **Digi-bus** protocol. The flow of data is right to left: signal conditioning modules using **MUX-bus** protocol are positioned to the right of the ADC, and modules receiving digitized data using **Digi-bus** protocol are positioned to the left.

The V208 provides aggregate sampling rates to 100 ksamples/s or 200 ksamples/s per second, depending on the option chosen, and supports up to 2048 **MUX-bus** channels. The V207 provides sampling rates to 500 ksamples/s and supports up to 256 **MUX-bus** channels. By using a V208 with eleven 96-channel signal-conditioning modules, you can have up to 1056 analog-input channels in one mainframe.

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;
- **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 V208.

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**.

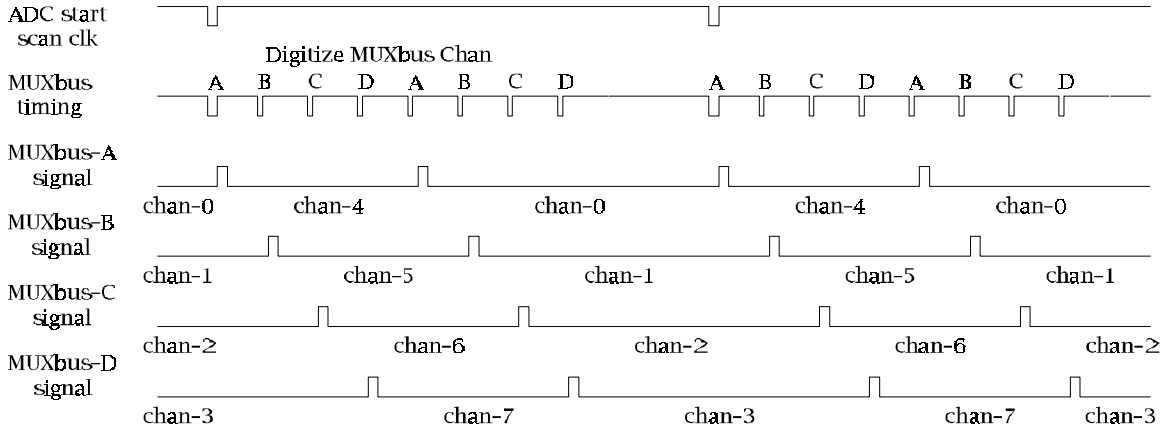
Data streams from the ADC to **Digi-bus** family modules over the VXI Local Bus using **Digi-bus** protocol and extends as far as adjacent modules in the VXIbus backplane propagate the bus. Brief descriptions of the **MUX-bus** and **Digi-bus** concepts and protocols follow.

The MUX-bus

The **MUX-bus** is a four-channel, differential, analog bus implemented on the VXI Local Bus. Under control of the V208, signal-conditioning modules multiplex up to 2048 analog signals onto the **MUX-bus**. The V208 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. The figure below illustrates the **MUX-bus** cycle for an 8-channel system.

The V208 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.



MUX-bus timing

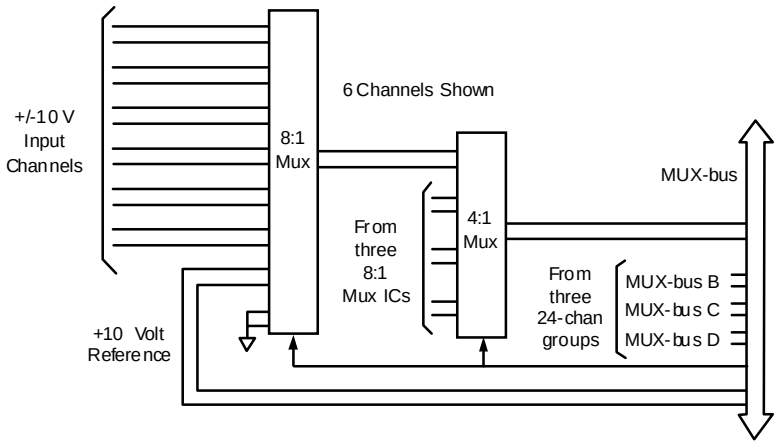
The V208 Family of Signal Conditioning and Multiplexing Modules

A number of multiplexing and signal-conditioning modules use the **MUX-bus** protocol to connect to the V208 ADC. The family consists of the V241 high-level scanning multiplexer, the V243 low-level signal conditioner, 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.

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. As many as eleven V241s can be used with the V208 in a single chassis for a maximum of 1056 active 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 V208.



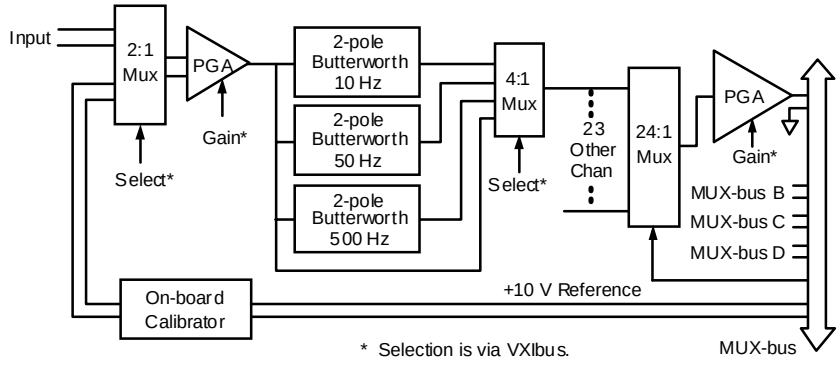
V241 Block Diagram

V243

The V243 acquires data from thermocouples and other low-bandwidth, low-level sensors. It provides up to 96 differential-input channels with programmable gain per channel and 2-pole, active, low-pass Butterworth filters on each channel. Filter cutoff frequencies of 10, 50, and 500 Hz (plus bypass) are software-selectable. Each channel has software-selectable pre-filter gains of 1, 10, and 100. Filter cutoff frequency and pre-filter gains are programmable in groups of eight channels. A common, multiplexed, post-filter gain of 1, 2, 5, 10, or 20 is selectable on a **MUX-bus** channel basis. Programmable gain per channel therefore ranges from 1 to 2000.

As many as eleven V243s can be placed in adjacent slots to the right of the V208 to provide 1056 active input channels.

V243s combined with the V208 provide a low-noise analog subsystem with built-in, per-channel calibration that is traceable to NIST standards. The V208 includes a factory-calibrated, precision reference source, and each V243 includes a precision calibrator. Each input channel connects to the calibrator output, ground, or the analog input under software control. In addition, for maximum accuracy, a small gain-correction factor for each calibrator output is stored in an EEPROM within each V243 as well as a small offset correction for different grounding paths. These correction factors can be applied during the calibration process to obtain maximum accuracy.



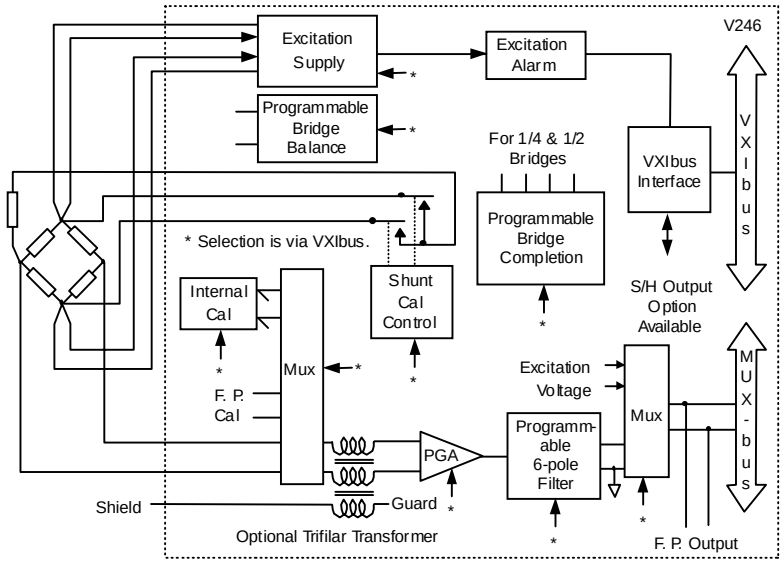
V243 Block Diagram

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.

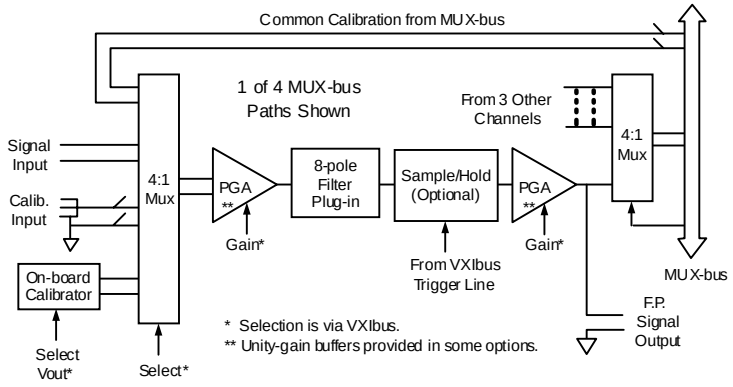


V246 Block Diagram

V252

The V252 provides eight or sixteen channels of fixed-frequency low-pass filtering. 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 100kHz.

Simultaneous sampling on all channels is available as an option. For calibration, the V208 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.

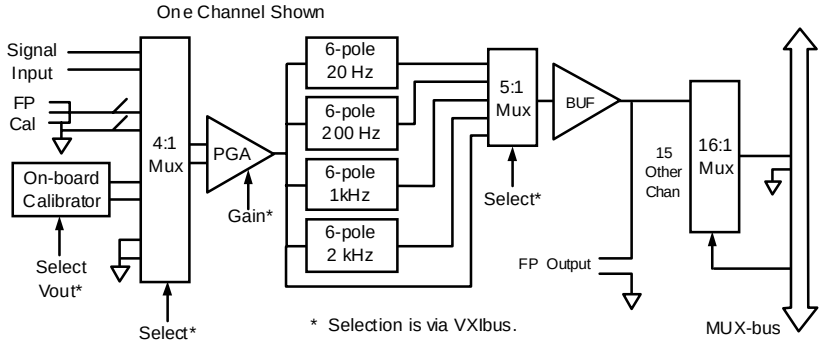


V252 Block Diagram

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.



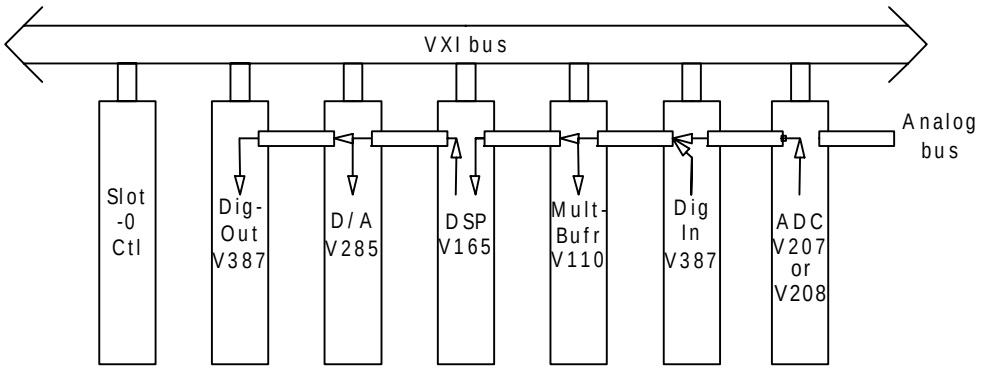
V253 Block Diagram

The Digi-bus

In multi-channel systems, the V208 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** allows the digital data from the V208 to flow to other modules without being subject to the timing uncertainties of the VXI backplane.

The following figure illustrates the **Digi-bus** concept. The V208 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.

Note that 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.



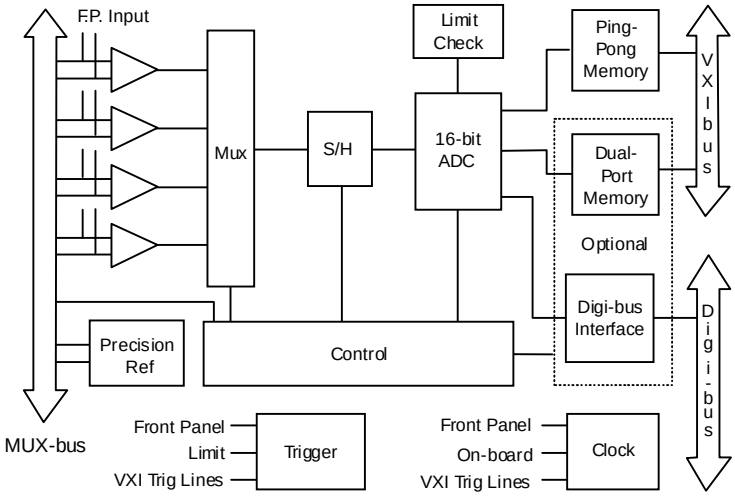
Digi-bus concept

Chapter 4: Understanding the V208

Overview

The V208 provides 16-bit analog-to-digital conversion at rates of 100 ksamples/s or 200 ksamples/s, depending on the option chosen. It is used with signal-conditioning multiplexing modules that use the analog **MUX-bus** protocol to pass analog signals to the V208.

The V208 provides four differential inputs that allow high data throughput with large numbers of input channels connected via the **MUX-bus**. The V208 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.



V208 block diagram

For continuous data access by the VXIbus, the V208 contains a "Ping-Pong" buffer, which is a pair of buffers each of which can hold 2048 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 V208. 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 V208 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 the V208 with Multi-buffer is *transient capture mode*—the ability of this module to store a block of data from an "event" in its local Multi-buffer, and then transmit that data over VXIbus. When this mode is activated, the V208 begins recording its data into the Multi-buffer in a circular-buffer fashion. The size of the buffer and the number of pre-trigger and post-trigger samples are programmable. The V208 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 the post-trigger countdown. After 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 V208, with the recording time only limited by the size of the Multi-buffer.

If the **Digi-bus** option is present, the V208 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 relates to the scan period.

Specifications

The following table presents the performance characteristics of the V208.

Item	Specifications
Input Channels	
Number	4, expandable to 2048 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	
Differential non-linearity	0.006% of FSR
Integral linearity	15 bits
No missing codes	Guaranteed
Dynamic Performance	
Maximum conversion rate	200 ksamples/s (V208-Zxyz), 100 ksamples/s (V208-Yxyz)
Cross-talk	-90 dB
Total harmonic distortion	-90 dB @ 100 kHz
Sample Clock	
Internal source	Derived from the VXIbus precision 10 MHz clock
Frequency choices	11 steps from 100Hz to 200 kHz (1, 2, 5 … progression)
External source	TTL signal to 200kHz
Duty cycle	50%
Backplane source	1 of 8 trigger lines on the VXIbus P2 connector
External Trigger	
Source	Negative-going TTL signal
Minimum value	50 ns
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	3.3 A without Multi-buffer, 4.4 A with Multi-buffer
-5.2 V	66 mA
+24 V	130 mA
-24 V	150 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)

Setup Memories

Scan RAM

You can multiplex up to 2048 analog-input channels into the V208. 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 2048-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 V208 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 V208 ADC. Only the end-of-list flag (bit 15) and the **MUX-bus** channel selection (bits 1 and 0) are used in the V208'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 V208'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 V208 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."

The following charts show the bit patterns to be written in the Scan RAM memories for this V208-V246 example:

Scan RAM - V208 ADC

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Data	Mux	Selection
000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
002h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
004h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
006h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
008h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
00Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
00Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
00Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0007h	D	
010h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
012h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
014h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
016h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
018h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
01Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
01Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
01Eh	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	8007h	D	End List
020h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
FFEh	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
000h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h	A	Channel 1
002h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001h	B	Channel 2
004h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002h	C	Channel 3
006h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4003h	D	Channel 4
008h	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4004h	A	Channel 5

00Ah	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	4005h	B	Channel 6
00Ch	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	4006h	C	Channel 7
00Eh	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	4007h	D	Channel 8
010h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0000h	A	
012h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
014h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
016h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
018h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0004h	A	
01Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
01Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
01Eh	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	8007h	D	End List
020h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
FFEh	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
000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	A	
002h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0001h	B	
004h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0002h	C	
006h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0003h	D	
008h	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0004h	A	
00Ah	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0005h	B	
00Ch	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0006h	C	
00Eh	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0007h	D	
010h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h	A	Channel 1
012h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	4001h	B	Channel 2
014h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	4002h	C	Channel 3
016h	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	4003h	D	Channel 4
018h	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	4004h	A	Channel 5
01Ah	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	4005h	B	Channel 6
01Ch	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0	4006h	C	Channel 7
01Eh	1	1	0	0	0	0	0	0	0	0	0	0	0	1	1	1	C007h	D	Channel 8 + End List
020h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h		
	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•			
FFEh	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 V208 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-Line 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 2048 in the V208. 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 V208 consists of two banks of 2048 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 V208. 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-Line Register #1 (offset 02h in A32 space). The processor can then monitor the selected trigger line and synchronize V208 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.

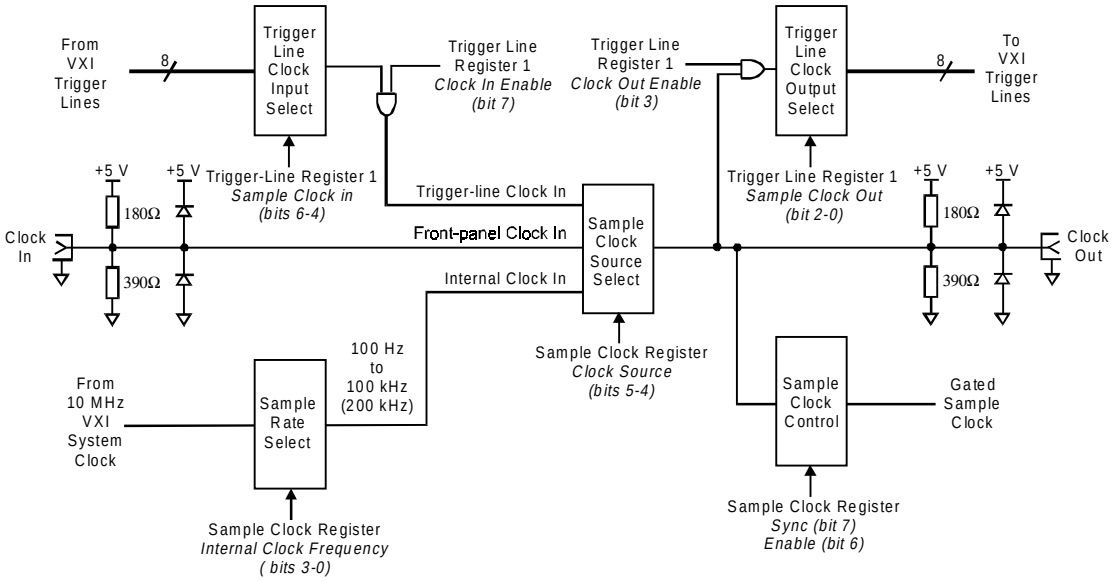
You can also configure the Multi-buffer for transient applications that need a large memory buffer. In transient mode, an "event" trigger starts post-trigger countdown. This trigger can be an event generated

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).

Timing Parameters: the sample clock, triggers and interrupts

Selecting the Sample Clock Source

As shown in the diagram, 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 Line Register #1 (at 02h offset in A32 space).



Selecting the Sample Clock Rate

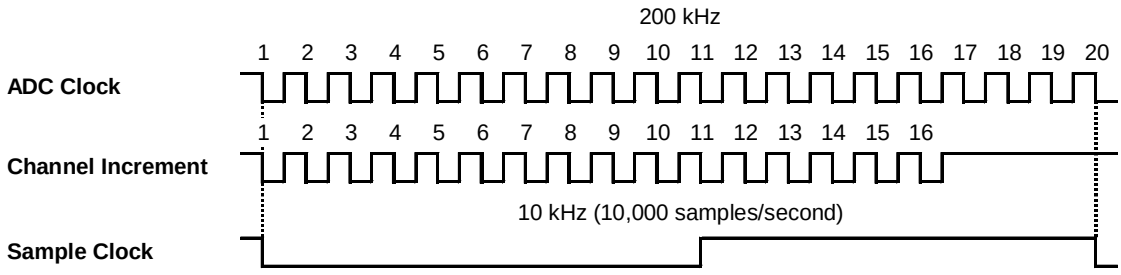
The frequency of the sample clock is the rate at which samples are digitized from each channel. The ADC digitizes at the maximum rate of 200 kHz (or 100 kHz for –Yxz2 options) regardless of the sample clock frequency. The *theoretical* maximum sample clock rate, therefore, is 200 kHz (or 100 kHz) divided 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 V208 ADC module can address up to 2048 channels of signal conditioning, with the modules coupled to the V208 via the backplane **MUX-bus**. Based on currently available modules, the maximum number of channels that can be supported in a 13-slot mainframe is 1056 (eleven 96-channel V241s and/or V243s).

The following definitions aid in understanding the V208 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, regardless of the sample clock selection. That rate is 100 kHz for the V208-Yxyz modules and 200 kHz for V208-Zxyz modules. This clock is derived from the 10 MHz CLK10 on the VXI backplane. Analog-to-digital conversion occurs at the ADC clock rate.

The following example shows a V208-Zxyz with its ADC clock rate of 200 kHz, 16 active signal conditioning channels and a sample clock rate of 10 kHz. This provides a per-channel sample rate of 10,000 samples per second and a data throughput rate of 160 ksamples/s.



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 200,000 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 type of V208 (100 or 200 kHz) and the number of active channels in the scan. The formulas are given here:

- V208-Yxyz Max Sample Clock Rate (Hz) = 100,000 Hz/Number of Active Channels
- V208-Zxyz Max Sample Clock Rate (Hz) = 200,000 Hz/Number of Active Channels

For example, a V208-Yxyz with 64 channels has a maximum sample clock rate of 1562.5 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. The following table gives 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.

Maximum Number of Channels with Internal Sample Clock

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

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.

Trigger Lines

The V208 makes use of the VXIbus trigger lines in a number of ways:

As Outputs

Using the Trigger-Line (TL) registers #1 and #2 in A32 space, you can drive selected VXIbus trigger lines (if enabled) with:

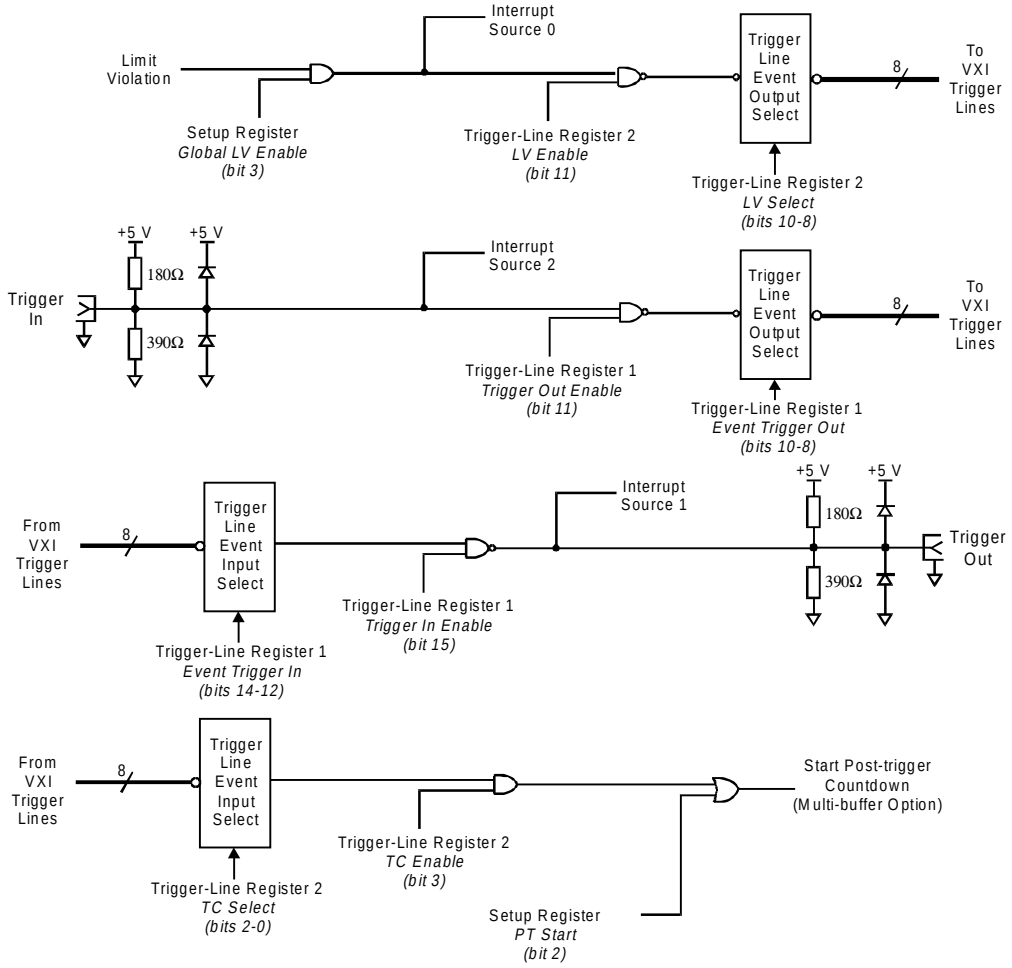
- the sample clock selected within the V208; (TL register #1, bits 3 – 0)
- the front-panel *Trigger In* signal; (TL register #1, bits 11 – 8)
- a Limit Violation event trigger occurring in the V208. (TL register #2, bits 11 – 8)

As inputs

Using the Trigger-Line registers #1 and #2 in A32 space, you can select (and enable) trigger lines to provide:

- the sample clock to be used by the V208; (TL register #1, bits 7 – 4)
- a front-panel *Trigger Out* signal; (TL register #1, bits 15 – 12)
- a trigger to start post-trigger countdown for Multi-buffer transient data capture. (TL register #2, bits 3 – 0)

The diagram illustrates the trigger line organization in the V208.



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 V208 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 V208

In the V208 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 V208s 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 V208s

The V208 includes various frequency dividers to derive the 200 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 V208s are in phase and, thus, no guarantee of the phase relationship between channels of different V208s.

The V208 provides for the synchronization of V208s in a single mainframe, such that the timing for the sampling of given channels among V208s 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-Line 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 V208 has eight interrupt sources. The table below lists these sources and their bit positions in the interrupt registers.

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 V208 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). There is an "interrupt pending" LED on the front panel.

	Interrupt Source #	Bit position in 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

Front Panel

LEDs

Add Rec

Illuminated when the module is being accessed.

Failed

Illuminated when the V208 has failed its self-test.

Int Src

Illuminated as long as the V208 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 V208 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 theV208, 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 **MU**

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 4 Gbytes.

Slave devices such as VXIbus data acquisition modules exist for a variety of purposes and can be very simple or very complex. Communication between host and slave can require access to a handful of registers in one device or access to many Mbytes of memory in another. ("Devices" and "modules" are terms often used interchangeably. A 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**…with 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 with 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. The following table shows the relationship of 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.

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	“

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 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, and the value held in the switch register is the Logical Address.

When the Logical Address is set to 255, a device responds to accesses at that address 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.

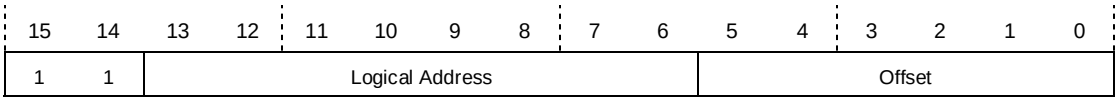
Every device has at least three configuration registers: ID / Logical Address, Device Type, and Status / Control. Modules using **A24** or **A32** addressing must also have an Offset register. The rest of the 64-byte block can contain registers appropriate for the operation of the specific device.

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 V208 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*

0 0 Memory device

0 1 Extended device

1 0 Message-based device

1 1 Register-based device

The V208 is an *Extended device.*

Addressing Mode *Bits 13 and 12*

0 0 A24

0 1 A32

1 0 Reserved

1 1 A16

The V208 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 V208. It returns 6208h or 7208h, 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	0	1	1	x	0	0	1	0	0	0	0	0	1	0	0	0
	Required Memory								**Model Code = 208h**							

Required Memory ***Bits 15 - 12***

0 1 1 0 A value of **6** indicates that the V208 uses 32 MB of A32 memory space.

0 1 1 1 A value of **7** indicates that the V208 uses 16 MB of A32 memory space.

Model Code ***Bits 11 - 0***

The model code for the V208 is 208h.

Status Register *04h*

This read-only register provides binary information about the status of the V208.

	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 V208 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 V208 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 V208 dynamically.

Ready ***Bit 3***

"**1**" in this field indicates that the registers have been successfully initialized. The V208 is ready for access.

Passed ***Bit 2***

"**1**" in this field indicates that the device self-test has passed. A "**0**" indicates that the V208 has failed—or is currently executing—its self-test.

Sysfail Inhibit ***Bit 1***

"**1**" in this field indicates that the V208 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 V208 is in a reset state. While in this state, the V208 will allow access only to its Configuration registers.

Control Register *04h*

This write-only register causes execution of specific actions by the V208.

	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 V208.

Sysfail Inhibit *Bit 1*

Setting this bit to "**1**" disables the V208 from driving the SYSFAIL* line.

Soft Reset *Bit 0*

Setting this bit to "**1**" forces the V208 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 V208 requires either 16 or 32 MB of the 4 GB **A32** memory space. The Required Memory parameter, *m*, in the Device Type register is therefore 7 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., 8 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 V208'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 V208 is capable of generating interrupts.

*Interrupt Handler Control** *Bit 1*

"**1**" indicates that the V208 is *not* capable of Interrupt Handler Control.

*Interrupt Status Reporting** *Bit 0*

"**0**" indicates that the V208 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 V208'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
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

Reserved / Logical Address *Bits 7 - 0*

During a read, these bits return all “1’s.” During an interrupt acknowledge cycle, these bits return the V208’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.

	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.

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

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 V208 is a VXIbus-defined Extended Device.

Bits 14 through 0 indicate that the V208 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 V208'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	X	x	0	1	0	0	0	x	x	x
	ASCII code for Y or Z = 59h or 5Ah (1st character)								**ASCII code for A, C or D = 41h, 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	1	0

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

The suffix options for the V208 are:

1st character: "Y" indicates the maximum digitizing rate is 100 ksamples / second.
"Z" indicates the maximum digitizing rate is 200 ksamples / second.

2nd character: "A" indicates that no option is present.
"C" indicates that the **Digi-bus** option is present.
"D" indicates that the Multi-buffer option is present.

3rd character: "1" gives no additional information.
"2" indicates that the Multi-buffer capacity is 4 Mbytes.
"3" indicates that the Multi-buffer capacity is 16 Mbytes.

4th character: "2" gives no additional information.

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 V208 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 mode to synchronize sample clocks across multiple V208s. Setting this bit to **"1"** enables *Sync* mode. In *Sync* mode, if the sample clock is not enabled (Enable = **"0"**) a trigger will clear the clock generator's internal counters. If the sample clock is enabled (Enable = **"1"**) a trigger will synchronously start the internal counters. Note that this mode is only valid in systems where the trigger event is synchronized to the 10 MHz clock.

Enable *Bit 6*

When set, the sample clock is enabled. Note that if *Sync* mode is set (i.e., Sync = **"1"**), the sample clock will not start until the next trigger (as selected by Trigger-Line Register #1 at 02h).

Clock Source *Bits 5, 4*

0 0	Internal Clock source
0 1	Clock source from Trigger Lines
1 0	External Clock source (Front Panel)

Internal Sample Clock *Bits 3 - 0*

0 0 0 0	not valid	0 1 1 0	5 kHz
0 0 0 1	200 kHz (not valid for V208-Yxy2)	0 1 1 1	2 kHz
0 0 1 0	100 kHz	1 0 0 0	1 kHz
0 0 1 1	50 kHz	1 0 0 1	500 Hz
0 1 0 0	20 kHz	1 0 1 0	200 Hz
0 1 0 1	10 kHz	1 0 1 1	100 Hz

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-Line Register #1 determine which trigger line is selected. The Front-panel Clock Input is TTL-compatible. Setting the enable bit to "**1**" enables the selected clock.

Trigger-Line Register #1 *02h*

This read/write register selects VXIbus trigger lines that connect logically to the front panel Trigger Out and Trigger In connectors and bring the sample clock into or out from this module.

	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*

This field selects 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:

Field	VXIbus Trigger Line
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

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 V208 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 V208 is thereby made available to other system elements.

Trigger-Line 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

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

Not used

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 reset 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*

	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

Setting this bit to **"1"** enables the start of post-trigger countdown for capture of transient data. When so enabled, the countdown can start when an event trigger occurs on a pre-selected VXIbus trigger line. The trigger-line selection is made in Trigger-Line Register #2, bits 2 – 0. Bit 3 in Trigger-Line Register #2 (TC Enable) must be set *after* bit 7 in the Setup register (Transient trigger Enable) is set.

Global LV Enable *Bit 3*

Setting this bit to **"1"** enables checking.

Post-trigger Start *Bit 2*

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

Multi-buffer Start *Bit 1*

Setting this bit to **"1"** begins filling the circular buffer.

Run/setup *Bit 0*

Setting this bit to **"1"** places the V208 in Run mode to begin acquiring data. Set this bit to **"0"** to place the V208 in Setup mode to configure or examine the contents of the Scan RAM or Limit Memory.

0 Setup mode

1 Run mode

Channel Limit-Violation Register *08h*

This read-only register indicates which channel (0 to 2047) 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	Limit Violation Channel Number										

Channel Number ***Bits 10 - 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. This includes all samples from *every* ***Digi-bus*** *source module* in this set. Setting the value to zero specifies one sample/frame. 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	Total Number of Samples per Frame from all modules															

Samples per Frame ***Bits 15 - 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					Strobe Enable	Number of Samples per Frame from this module										

Strobe Enable ***Bit 11***

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

Samples per Frame ***Bits 10 - 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	Starting Time-Slot															

Starting Time-Slot ***Bits 15 - 0***

Digi-bus Example 1 - One source module

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

V208 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

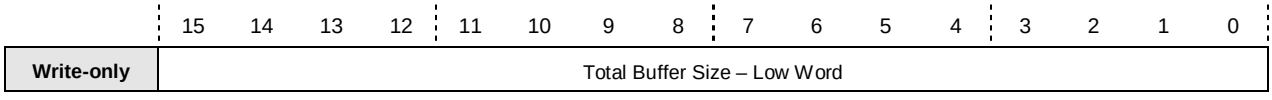
Total Buffer-Size Register (Multi-buffer option) *20h, 22h*

The Total Buffer-Size registers (high word and low word) must be written with the 32-bit value representing the Multi-buffer memory required for the application. The following equation gives the value:

[(individual buffer size) x (number of buffers)] - 1

The individual buffer size is the 32-bit value written to the Individual Buffer Size registers (24h and 26h in A32 space).

20h



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 registers (high word and low word) must be written with a 32-bit value representing the number of 32-bit words (i.e., 2 data samples) desired in each segment of the Multi-buffer memory.

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 in that writing a **"1"** to a bit position resets that bit to **"0."**

	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*

The Transient Complete bit is set to a **"1"** when the transient acquisition is complete. It signifies that the post-trigger countdown has decremented to zero. This bit is reset to **"0"** when the V208 is placed into "Run" mode and the "MB Start" bit is set.

Overrun *Bit 8*

The Overrun bit indicates a buffer has been over-written before its contents were read. This bit is cleared by writing a **"1"** to bit 8, which results in a selective clear of that bit.

Buffer Full Flags *Bits 7 – 0*

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 must be written with a 32-bit value corresponding to desired number of ″Post Trigger″ Samples/Channel. These registers need only be programmed 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 1000h - 1FFEh

The Scan RAM contains the scan table which must be set up properly on all **MUX-bus** modules. The Scan RAM is 2048 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 used in the V208. 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 10 - 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 V208 in "Setup" mode. The V208 (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 0-10 specify the channel number (0-2047). Bits 11-14 are not used in the V208 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 2048 ADC time slots per frame. 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 00000000000 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 10-0)	Comment
1000h	xxxxxxxxx00	These channels are assigned to MUX-bus path A.
1002h	xxxxxxxxx01	These channels are assigned to MUX-bus path B.
1004h	xxxxxxxxx10	These channels are assigned to MUX-bus path C.
1006h	xxxxxxxxx11	These channels are assigned to MUX-bus path D.
1008h	xxxxxxxxx00	Repeat back to MUXbus path A, etc.

Limit Memory *2000h - 2FFEh*

The Limit Memory contains the data for limit checking (using level and slope) in the V208. 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.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read/write	Not Used						Slope	Enable	Limit Values							

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 *3000h - 3FFEh*

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 3000h) is associated with the channel number specified in address x of the Scan RAM (relative to the base address of 1000h).

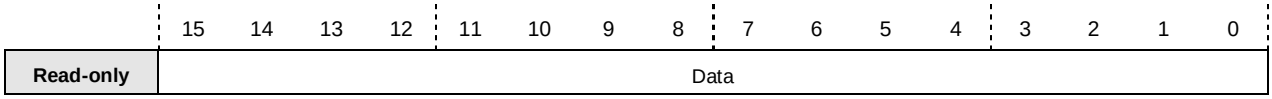
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read-only	Data															

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

Multi-buffer Memory (optional)

4 MB option ***400000h - 7FFFFEh***

16 MB option ***1000000h - 1FFFFFEh***



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.

Chapter 6: Programming Information

Example Setup Procedure

4-Channel Transient Capture with Pre-Trigger

Following is an example setup procedure to acquire four channels of data via the V208 front-panel connectors in Transient mode.

- Read register 06h in A16 space to get the A32 offset.
- Logical shift A32 offset 16 places to the left to get the V208 base address.

All registers from here on are in A32 space, and the corresponding addresses should be added to the A32 base address.

- Write register 00h with data of 44h (20 kHz). *This specifies a 20 kHz Sample Clock rate, generated from the Internal Clock Source. It also enables the Sample Clock.*
- Write register 06h with data of 00h. *This puts the module in Setup mode.*
- Write register 1000h with data of 0000h. *This enables channel 1 in Scan RAM.*
- Write register 1002h with data of 0001h. *This enables channel 2 in Scan RAM.*
- Write register 1004h with data of 0002h. *This enables channel 3 in Scan RAM.*
- Write register 1006h with data of 8003h. *This enables channel 4 in Scan RAM and specifies end of list.*
- Write register at 20h with data of FFFh (4095). *Sets Multi-buffer total size to 4k 32-bit words, or 8 ksamples (2 ksamples per channel).*
- Write register at 22h with data of 00h. *Upper address bits = 0 for 4k buffer.*
- Write register at 24h with data of 1000h (4096). *Set to 1 more than the number written in #7 and #8. This sets the Individual Buffer Size in accordance with one desired buffer for this transient example.*
- Write register at 26h with data of 00h. *Upper bits of Individual Buffer Size High register. This register provides the upper 16 bits for the Individual Buffer Size and should be used when the value is greater than 16 bits.*
- Write register at 30h with data of 200h. *Sets Countdown register Lo to 512 post-trigger samples per channel.*
- Write register at 32h with data of 00h. *Upper bits of Countdown register.*
- Write register 06h with data of 3h. *This puts the V208 in Run mode and begins storing data to the Multi-buffer.*

When you are ready to trigger data collection:

- Write register 06h with data of 7h. *This triggers the transient acquisition and leaves the module in run mode and writing to Multi-buffer.*
- Now, poll on bit 0 (Buffer #1 Full Flag) of register 28h. *Continue when that bit is set.*
- Read data starting from 400000h *(assumes a 4 Mbyte Multi-buffer). Data will be in consecutive locations with the channels interleaved.*

Channel 1, Sample 1	400000h
Channel 2, Sample 1	400002h
Channel 3, Sample 1	400004h
Channel 4, Sample 1	400006h
Channel 1, Sample 2	400008h
Channel 2, Sample 2	40000Ah

Appendices

Buffering

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 {\em 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.

A number of mechanisms are provided to move sampled data

Mechanism	**Access**	**Availability**
Ping-Pong Memory	VXIbus	standard all modules
Multi-buffer	VXIbus	option all modules
Digi-bus	VXI Local bus	option V207, V208
Com Port	Separate Cable	option V200

Ping-Pong Memory

This memory is accessable from the VXIbus and holds data values from the previous sample clock. This memory is updated at each sample clock edge (V200 Ping-Pong flip trigger lines) by the ADC.

Multi-buffer Option

This optional memory provides buffering for all channels. The memory from the ADC side functions as a large circular buffer with data strobed into it as it is available from the ADC. The size of the circular buffer is defined during setup up to the full available memory size.

Digi-bus Option

This option is a private bus to the left of the ADC using the VXI Local bus. Data from the previous data can (sample clock) is strobed onto the Digi-bus at the sample clock edge.

Com Port Option (V200)

This option is a private bus from individual V200 modules to a TI TMS320C40 Digital Signal Processor. Data is streamed across this bus synchronous with the V200 Ping-Pong flip trigger line.

Acquiring Data from Ping-Pong Memory

For applications with limited channels and slow sampling rates (sample clock rates), 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 the Ping-Pong memory only buffers a single data scan, it is incumbent on the application software to read the data from the Ping-Pong memory between sample clock events. Thus for a sample clock rate of 10 Hertz the application software must read all the data from the Ping-Pong memory in the 100ms time interval defined by the sample clock for each occurrence of 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 the data acquisition task is the only task running. This is because multiple tasks are competing for the system resources. In many cases these are system tasks or drivers responding to interrupts both having higher priorities than a user task. In some cases the user has little or no control over tasks. Windows 95 is a good example. Even in operating systems where there is significant control (e.g. the VxWorks kernel), it is hard to prevent a low priority task from issuing, for example an I/O request, that then results in very high priority driver I/O interrupt handling. This is known as blocking.

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

In the case of data acquisition, loss of a 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 hard number to determine. As a general rule one can service Ping-Pong reliably below about 10 Hz., and 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 and no or minimal operating systems.

Multi-Buffer

By buffering data from multiple data scans (sample clocks) at the ADC it is possible to minimize the effects of latency situations discussed in the previous section. Data buffering reduces the frequency that the process latency must be accounted for. For example, by buffering data from 100 sample clocks the variable latency in responding to the event is reduced by a factor of 100. There is only one event to respond to rather than one hundred. Larger data blocks (for example 2000 samples per transfer rather than 20) allow for much more efficient DMA data transfers with greatly reduced CPU utilization.

While all the considerations about multitasking still apply, now these effects only come into play at rate of 1/ *buffering-factor* or as in our examples above at 1% of the unbuffered frequency. Thus for a buffering-factor of 100:1, our maximum "safe" sample rate goes from 10 Hz to 1 KHz., 1000:1 gets the maximum sampling rate to 10 Khz, etc.

The down side of buffering is that if we wish to monitor the data in quasi-realtime, the data available at the processor to monitor will be delayed by as much as the buffering factor, and on average *buffering-factor*/2. This can be partially overcome by multi-buffering. That is by subdividing the buffer into some number of segments. The standard V207/V208 multibuffer 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 than the size imposed by the worst case latency, since there are in general 7 additional segments to fill before losing data and in general the average latency is much less than the worst case latency.

About KineticSystems

KineticSystems Corporation 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 turn-key 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 been providing 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 Reality®. 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 are providing 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.

How to contact us

KineticSystems Corporation

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

techserv@kscorp.com

sales@kscorp.com

Web: http://www.kscorp.com

Warranty

KineticSystems Corporation 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. Software products manufactured by KineticSystems are warranted to conform to the Software Product Description (SPD) applicable at the time of purchase for a period of ninety days from the date of shipment to the original end user. Any 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 shall be prepaid by the purchaser, while charges for returning the repaired product to the purchaser, if located in the United States, shall be paid by KineticSystems. Return shipment will be made by UPS, where available, unless the purchaser requests a premium method of shipment at his expense. The selected carrier should not be construed to be the agent of KineticSystems, nor will KineticSystems assume any 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.

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. Product failures created by unauthorized modifications, product misuse or improper installation are not covered by this warranty.

The warranties provided herein are the purchaser's sole and exclusive remedies on any claim of any kind for any loss or damage arising out of, connected with, or resulting from the use, performance or breach thereof, or from the design, manufacture, sale, delivery, resale, or repair or use of any products covered or furnished by KineticSystems including but not limited to any claim of negligence or other tortious breach, shall be the repair or replacement, FOB factory, as KineticSystems may elect, of the product or part thereof giving rise to such claim, except that KineticSystems' liability for such repair or replacement shall in no event exceed the contract price allocable to the products or part thereof which gives rise to the claim. In no event shall KineticSystems be liable for direct, indirect, special, incidental, or consequential damages, including loss of profits.

Products will not be accepted for credit or exchange without the prior written approval of KineticSystems. 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 prior to shipping the product to KineticSystems. The following steps should be taken before returning any product:

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 a description of the problem and the name of your technical contact person

	with the product.
5	Ship the product prepaid with the RA Number marked on the outside of the package to:

USA, Canada, and Mexico

KineticSystems Corporation

Repair Service Center

900 North State Street

Lockport, IL 60441

Telephone: (815) 838-0005

Facsimile: (815) 838-4424

Index

S

T

U

V

Feedback

The purpose of this manual is to provide you with the information you need to make the V208 as easy as possible to understand and to 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 encourage you to fill out this form and Fax it back to us at (815) 838 4424. Your input is extremely valuable to us.

Please rate each of the following.

The information in this manual is:

	Agree									Disagree
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:

Name Phone

Company