VXI Data Acquisition Handbook

J. W. Tippie, Ph.D.

KineticSystems Corporation

February 16, 1998
Revision 1.3

KineticSystems Corporation

900 N. State St., Lockport, Illinois 60441 • (815) 838-0005 • FAX (815) 838-4424

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

The purpose of this guide is to provide the reader with an overview of important considerations in the design and implementation of high performance data acquisition systems *and* how KineticSystems products can be configured to meet some of the more demanding applications. The product designers at KSC have gone to considerable effort to design a *family* of VXI modules which provide a wide range of flexibility in the design of high performance systems. It is important to understand how the features of the various system components interact and can be brought together to provide unique system capabilities.

This document goes beyond a functional description of the individual modules and illustrates some of the unique features that have been designed into the system components. KineticSystems manufactures a wide variety of modules that are appropriate over a wide range of applications. This guide focuses on a subset of modules that play a critical role in demanding high performance applications. Not all applications require high performance, and many high performance applications have components that are low or medium performance that can be more cost effectively solved with other components. For this reason the data acquisition systems designer should be acquainted with the full product line and not just the limited products discussed here.

Finally there is the question of just what is "high performance?" There are many answers to this question and one needs to keep in mind that while one thermocouple channel with a 10 Hz sample rate may not be "high performance," 1000 channels at 10 Hz with 3 microvolt accuracy may be something else.

1.2 VXI

The choice of a modular backplane standard for I/O can be an important consideration. VXI is an industry standard with multiple manufacturers. It is an extension of the VME bus standard for Instrumentation. For purposes of this document we will focus on the C-size VXI. Some of the features it brings to the user over VME are as follows:

- The VXI standard provides a number of chassis-wide timing features that can be particularly valuable in synchronizing data acquisition activity among multiple modules. For C-size VXI these include:

  - Common precision 10 MHz clock that can be used as a common clock to synchronize activity within modules.
  - Eight TTL and two ECL "trigger lines" for inter-module synchronization such as sample clocks and event triggering.

  These features can be particularly valuable in applications where it is important to synchronize data acquisition activity across multiple modules.

- The VXI standard provides a 12-bit "local" bus between adjacent modules that can be used to pass analog or digital information between cooperating modules. This bus frequently eliminates the need for "messy" wiring between modules making for a cleaner looking configuration with significantly improved system reliability.

- The VXI standard provides a *geographic addressing* capability. This can be important since wiring is more easily associated with a physical location (slot) than by module type and it's associated options and address strapping—especially when the address strapping is not visible from the front panel. Be sure *your* VXI vendor's modules support this feature.

- The shielded module provides good EMI shielding for sensitive analog circuitry which is necessary for good accuracy.

- The deeper card allows the designer to place analog circuitry further from the high speed digital backplane further reducing noise sensitivity.

- Chassis cooling specifications of the VXI standard insures a good uniform cooling environment which minimizes thermal drift and gradients. High-gain analog circuitry is generally sensitive to the thermal environment which affects stability and channel-to-channel measurement accuracy particularly with thermocouples and other low-level analog signals.

- The VXI standard provides *Module ID* registers that allow software to verify that it is addressing the proper module. Modules are identified by a unique manufacturers ID, model number and hardware/firmware revision level. This can avoid costly mistakes resulting from taking data from an inappropriate module. It also enables software to verify system configuration—especially when combined with geographic addressing.

- The VXI standard provides for each module, as appropriate, to perform any self-diagnostics during power-up and provide pass/fail status bits. This feature reduces the chances of attempting to take data from a failed module.

Some features unique to the KineticSystems *second generation* VXI products include:

- Automated channel-by-channel calibration and system checkout is an important tool in data acquisition. KSC modules implement the capability to inject either an internal or external precision reference into each channel for calibration and end-to-end system checkout.

- Many VXI modules have multiple options available. KSC modules implement a *module option ID* in addition to the standard manufacturers ID and model number for more complete module identification.

- In many cases users have a need to track certain critical information regarding particular modules. KSC modules implement a user-writable EEPROM for this purpose.

- Signal conditioning is an important consideration in many data acquisition applications. To provide maximum flexibility in signal conditioning KSC has developed a powerful new architecture that makes extensive use of the VXI Local Bus to interconnect the ADC to a family of signal conditioning modules on a mix and match basis. The architecture provides a clean analog environment at the signal conditioner with minimum digital circuitry to introduce noise in low-level front-end circuits and sufficient board real estate to provide a wide range of signal conditioning capabilities *with no inter-module wiring mess.*

- KSC modules make full use of VXI trigger lines for inter-module synchronization and clocking. The KSC *standard* is to provide full flexibility to the user in trigger line assignment and usage and also provides external clocking and triggering capability.

Chapter 2

VXI Architecture

The VXI standard was initiated in 1987 when representatives from Colorado Data Systems, Hewlett-Packard, Racal Dana, Tektronix, and Wavetex formed an ad-hoc committee to engineer standards for a new open architecture instrumentation bus based on VMEbus, the Eurocard standard and other standards such as IEEE 488.

The objective was to formulate a modular standard for instrumentation that supersedes the "rack-and-stack" instrumentation based on IEEE 488 and provide benefits of speed and modularity of VME.

VXIbus is the acronym for VMEbus eXtensions for Instrumentation. The goal of the VXIbus standard was to define a technically sound instrumentation specification based on VMEbus that is open to all manufactures and is compatible with present industry standards.

The basic relationship between VME and VXI is shown here:

- VXI uses the VME bus protocol for data transfer between modules.
- VXI uses the identical backplane connector pin out as VME for the P1 (top) connector and for the center row of the P2 connector. The two outer rows of the P2 connector are undefined in VME.
- VXI adds many specifications—including the signal definition of the outer rows of P2 and all of P3, if used—to those provided by VME.

VXI provides numerous enhancements to VME when applied to instrumentation. The major improvements are:

- VXI provides larger card options than VME to allow room for sophisticated analog modules and to isolate critical signals from the digital backplane.
- VXI includes mandatory analog power supply voltages.
- VXI provides for shields on C- and D-size modules.
- VXI specifies clock and trigger lines, geographic addressing, an analog summing bus, and a local bus on connector P2—and P3, if used.
- VXI specifies important characteristics, such as shielding and cooling, of the I/O chassis. These chassis are called mainframes.

2.1 VXI Standard

The VXI standard is based on the VMEbus and provides a number of highly desirable extensions that are valuable in configuring data acquisition, control, and automatic test systems. The standard employs a powered chassis or “mainframe” with a controller in the first (left-most) slot and 12 additional slots for instrumentation modules.

Extensions to VMEbus include:

Larger Cards: The card size (C- and D-size) has increased depth over VME to provide additional area to implement instrumentation functions. The module spacing is also increased to provide room for two boards internal to the module and additional room for connections to the front panel.

Shielding: The VXI specification requires shielding on both sides of C- and D-size modules to minimize inter-board noise and minimize EMI.

Chassis-wide Clocks and Triggers: The VXI specification provides for a number of chassis-wide clocks and trigger lines for inter-module synchronization.

Local Bus: A *local bus* is defined by the standard. It is propagated from slot to slot forming a *private* bus for communications between adjacent modules. Front panels are keyed so that modules that use the local bus with significantly different signal levels cannot be inserted in the backplane adjacent to each other. For example a module that uses local bus for TTL signals and one that uses local bus for analog signals.

EMC and Cooling Specifications The standard establishes EMC and cooling specifications for chassis.

Massaging Protocol The standard establishes a messaging protocol for modules that use message-based communications similar to IEEE 488 devices. The protocol provides a level of interoperability between modules form different vendors.

Module Identification Each VXI module includes registers that can be read from software which identify the module manufacturer as well as a module identifier (part number) and version number. Each manufacturer is assigned a unique identifier.

Geographic Addressing A mechanism is provided so that modules can be referenced based on their physical location in the chassis. This can be extremely valuable when multiple modules of the same or similar type are located in the chassis with wiring from a module at a specific location to the device(s) under test.

VME Module Coexistence The standard provides the ability to use VME modules within a VXI chassis provided they do not use *uncommitted lines* which conflict with VXI.

Improved Standard Power Specifications The VXI standard provides for increased number of standard power voltages as well as tighter specifications on allowed power noise and ripple over VME. These standards provide the voltages that are common in analog circuitry as well as improved noise immunity.

2.1.1 VXI Module Sizes

There are three VXI module sizes, defined as B, C and D. The B-size module has the same dimensions as a B-size VME module, with all three rows of contacts on the P2 (bottom) connector fully defined for the VXI module. The relative sizes of the three module types are shown in Figure 2.1. For most instrumentation, the clear choice is the C-size module. This configuration provides sufficient board space for sophisticated instrumentation and is deep enough to allow for physical isolation of low-level analog signals from the fast-risetime signals associated with the digital backplane. D-size modules also can be used for very sophisticated circuitry. However, the extremely large board area of this module makes an extremely high-cost building block.

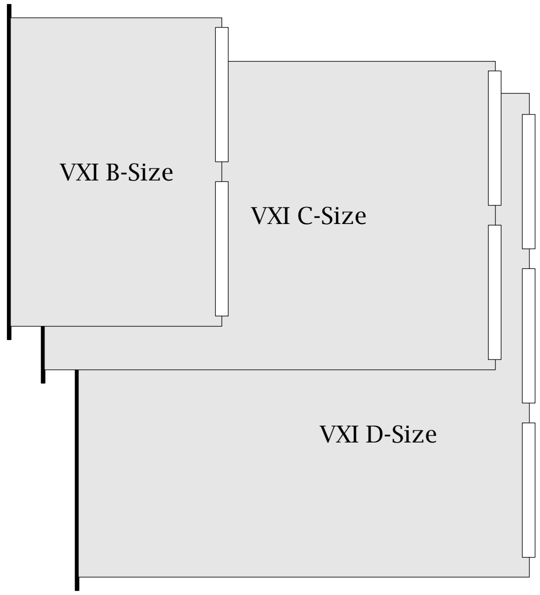


Figure 2.1: Relative sizes of VXI module options

VME and VXI modules can be mixed in the same chassis. However, care must be taken to prevent severe damage to the modules. The outer rows of the P2 connectors in a VXI mainframe include +5 volt, -2 volt, -5.2 volt and ±24 volt power pins. Many VME modules include manufacturer-defined intermodule buses on these connector rows. Therefore, inserting such a VME module directly into the P2 connector will destroy any TTL logic connected to these power pins. These ICs may then short circuit to other P2 contacts and damage logic in the VXI modules contained in the mainframe. Adapters are available from several manufacturers to allow VME B-size modules to be inserted in VXI C-size mainframes. These units "split" rows A and C on the P2 connector to prevent the connection of incompatible signals. The adapters also logically buffer the bus lines so that unacceptably long unterminated signal "stubs" are eliminated.

2.1.2 The VXI Mainframe

The powered chassis that houses VXI modules is called a mainframe. Since B-size modules do not include shields, a full-size mainframe for these modules contains slots for 20 modules, while C- and D-size mainframes can accommodate 13 modules. On these chassis the module positions are designated, from left to right, slots 0 through 12. The VXI specification covers important mainframe considerations, such as cooling, power supply noise, electromagnetic compatibility (EMC) and backplane construction. VXI can be used for high frequency applications into the gigahertz range. The specification allows mainframes to include optional shields between the modules for these applications.. More consistent cooling is provided by mainframes that include fans that "push" the air through the modules and that include air baffles at each slot to provide sufficient air flow through the modules in a partially filled mainframe. Careful choice of a VXI mainframe is extremely important for successful system operation.

2.1.3 The Slot-0 Controller

Since VME—and, by reference, VXI—is a multiprocessor bus, any slot in a VXI mainframe can act as bus master. However, Slot 0, the leftmost slot in the chassis, includes some unique features and backplane wiring. The VXI mainframe "host" is usually an embedded computer that is part of the Slot-0 Controller or an external computer connected by a bus connector on the front of the controller module. Popular embedded controllers include 486 PC-compatible and 68030-, 40-, and 60-based processors. Connection to an external computer can be made by use of the MXI parallel bus specified by VXI, Ethernet (with an intelligent processor), IEEE-488 or other interconnection bus.

A Slot-0 Controller provides two primary resources via the P2 connector to the VXI modules, a 10 megahertz common clock and module ID lines. These functions are implemented for the B, C and D form factor. D-size controllers also provide a 100 megahertz common clock on the P3 connecter.

2.1.4 Using the Module ID Lines

Twelve module ID lines (called MODID) emanate from Slot 0 to the other twelve slots in a 13-slot VXI mainframe. These lines are driven by the Slot-0 Controller and are used for geographic addressing in the following manner:

- A logical address switch is provided on each VXI module. This switch is used to set the logical base address for the module—the location of a block in memory space for the module's I/O registers.

- If the module address switch is set to “all ones,” the module uses its slot position (geographical address) for autoconfiguration of its logical address.

- The resource manager software contains information regarding the physical location of each module and its logical base address.

- During autoconfiguration, the system software interrogates each module via its MODID line to ensure that the proper module is located in each slot. The software can also determine if a module is not present.

- Again, by geographic addressing, the software transfers data to a register that sets the logical base address for that module.

- This process is continued for all modules that support geographic addressing.

While all Slot-0 Controllers must support geographic addressing (MODID), this is an optional feature for all other module types. Particularly for data acquisition and control applications, it is extremely important that the modules support geographic addressing. The primary reasons are given here:

- Incorrect setting of the address registers on modules can lead to severe problems and even equipment damage—if a module controls the wrong I/O points.

- On systems that contain more than one I/O module of a given type, it is fairly easy to switch the positions of two modules after removing them from the mainframe. When the modules are re-cabled, they will be connected to the wrong I/O points if module ID setup is not used.

- Autoconfiguration greatly simplifies system setup, and the software can check that the modules are located in the desired slots.

2.2 Register-Based vs. Message-Based Modules

VXI specifies that modules can communicate over the backplane by register-based or message-based protocol. 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. Also, if a consistent lexicon is used, such as the Standard Commands for Programmable Instrumentation (SCPI) specified by the consortium, an instrument from another manufacturer, ideally, could be substituted with little or no change in the software.

High-performance data acquisition and control modules are usually register-based for the following reasons:

- The I/O throughput of a register-based module may exceed 1,000 times that of a message-based module, even if they both are performing the same functions, due to the extensive time needed to parse the word-serial messages in the message-based module.
- Most high-performance data acquisition and control modules contain a number of unique functions, obviating the ability to use the same messages for other brands of hardware without software changes.
- The tendency is to consider a software driver to be "par" of a VXI module. A properly designed software driver for a register-based module can make the interface to an application program easier, while maintaining high I/O throughput.

2.3 VXI C-Size

The VXI C-size modules are by far the most popular. They provide sufficient board real estate to implement most functions required in data acquisition while at the same time providing an adequate level of modularity. For these reasons KineticSystems Corporation has chosen to focus development in this area.

Also the VXI standard provides for both *Register-based* and *Message-based* modules. Again KineticSystems has chosen to focus on register-based devices because of the significantly higher performance that these modules offer.

This document will focus on C-size Register-based VXI.

2.3.1 Clocks

The VXI Slot-0 controller is required to provide a 10 MHz buffered differential ECL system clock. This clock is required to have a minimum accuracy of ±100ppm (0.01%) over its specified operating temperature and time. Further the absolute delay of CLK10 from Slot-0 to any module *shall not* exceed 8ns and each slot's CLK10 is driven by a unique backplane buffer output over separate traces to the slot.

This clock provides a common precise time base that can insure synchronous operation of all modules in the chassis which use this clock. Further the VXI specification recommends that the Slot-0 controller provide for an external 10 MHz clock as an alternative to the internal clock so multiple chassis can be synchronized from a single common clock or driven from a frequency reference such as a Rubidium Standard.

2.3.2 Trigger Lines

The VXI standard provides 8 TTL and 2 ECL trigger lines on C-size VXI. These lines provide chassis wide synchronization.

TTL Trigger Lines

These lines are open collector and can be driven from any slot in the crate. Three signal protocols are defined by the standard, SYNC, ASYNC, and Start/Stop (STST).

SYNC The synchronous protocol is the most common. Any module may assert the trigger line and one or more modules may monitor the line.

ASYNC The asynchronous protocol involves two trigger lines with a single source and single acceptor. The source initiates an operation by asserting the lower numbered line and the acceptor acknowledges by asserting the higher numbered line.

Start/Stop Start/Stop (STST) protocol provides a method for starting and stopping module clusters synchronously. One trigger line is driven from the Slot-0 controller and its state signifies **START** or **STOP**. All participating modules respond to this line synchronously at the next CLK10 rising edge.

ECL Trigger Lines

The two ECL trigger lines on C-size VXI are intended to be an intermodule timing resource. Any module including the Slot-0 controller may drive these lines or receive information from these lines. The ECL trigger lines support a similar protocol to the TTL trigger lines. Refer to the standard for details.

2.3.3 SUMBUS

The SUMBUS is a 50Ω analog summing node that is bused the length of the VXIbus backplane. Any module can receive information from this bus through a high impedance receiver. Modules driving the bus use an analog current source driver.

2.3.4 Local Bus

Local Bus is a daisy chained bus that is propagated from slot to slot. It is defined by adjacently installed modules. The backplane connects the LBUSC pins of Slot N to LBUSA pins of Slot N+1. For C-size VXI the Local Bus is a 12-line bus. Several *signal classes* are allowed. These are summarized in table 2.1. A front panel key prevents inserting VXI modules with different Local Bus *signal classes* adjacent to each other to prevent damage. Note that for proper operation the user *must* insure that adjacent modules that use Local Bus are designed to work with each other.

Beyond this *class standard* for Local Bus, VXIbus designers are free to assign signals on the local bus in any manner consistent with the broad guidelines established by the standard. When configuring systems with modules that have Local Bus options, the user should exercise care to be sure that adjacent modules use Local Bus in a consistent manner. For example just because two adjacent modules use Local Bus for example TTL-level signals *does not insure that both modules use the Local Bus TTL signals in the same way.*

2.3.5 VMEbus Addressing Modes

The VMEbus specification forms a part of the VXI standard. The VMEbus has a 100 ns cycle and has a 32-bit wide data path. It supports three addressing modes A16, A24, and A32 as well as three data transfer modes D8, D16, and D32.

2.3.6 VMEbus Throughput Considerations

The theoretical limiting throughput of the VMEbus is 40 Mbytes/second, however this throughput is rarely achieved in practice. A number of considerations

Table 2.1: Local Bus Signal Classes

Number	Class	-Limit	+ Limit	Drive Limit
1	TTL	-0.5 V	+5.5 V	200 mA
2	ECL	-5.46 V	+0.0 V	50 mA
3	Analog Low	-5.5 V	+5.5 V	50 Ω
4	Analog Med	-16.0 V	+16.0 V	500 mA
5	Analog High	-42.0 V	+42.0 V	500 mA
6	*Reserved*			

become important in determining the actual throughput that can be realistically achieved.

The VMEbus is an arbitrated bus. Thus for a module to become bus master to initiate a transfer, the module must arbitrate for bus mastership. Other bus transactions can delay acquiring bus mastership as well as the arbitration overhead.

The VMEbus supports 8, 16, and 32-bit transfers. Each transfer takes a minimum of one bus cycle of 100 ns or more. Thus the limiting throughput for 8-bit transfers is 10 Mbyte/sec, 16-bit transfers 20 Mbyte/sec, and 32-bit transfers 40 Mbyte/sec. Modules supporting D32 provide the best throughput.

The VMEbus transfers are handshaked. Thus the time to perform a single transfer depends on how fast the addressed module responds. This alone can severely limit throughput as it depends critically on both the addressed modules interface to the VMEbus and the access time of the addressed register or memory.

Transfers on the VMEbus can be one of two types; *single transfers* and *burst mode*. For *single transfers* the bus master arbitrates for the bus, asserts the bus address, transfers the data, and finally releases the bus. A DMA transfer using this mode must go through this sequence for each data word transfer.

For *burst mode* transfers the transfer is identical to the *single transfer* except the DMA controller does not release the bus and only asserts the address once at the start of the transfer. Burst mode transfers are limited to blocks of 256 or less transfers before the controller must release the bus and re-arbitrate. This is to insure other devices have a chance to acquire bus mastership. Burst mode provides the highest throughput, but may introduce unacceptable latencies for other system components.

Chapter 3

Analog Architecture

The KineticSystems *second generation* analog modules consist of three families of ADCs and signal conditioning covering the low, medium, and high performance range.

High Performance	V200
Medium Performance	V207/V208
Low Performance	V213

3.1 ADC Performance Comparison

A quick comparison of ADC conversion rates is illustrated in Table 3.1. It should be noted that conversion rates (sampling rates) do not give the complete picture.

Other considerations include signal bandwidth, antialiasing requirements, noise environment, accuracy requirements, and the cost of oversampling *(through-*

Table 3.1: ADC Conversion Rates.

ADC	Sample Rate
V200	195 Ksample/sec.
V207	500 Ksample/sec.
V208	100 Ksample/sec.
V213	50 Ksample/sec.

put, data storage, data access time).

Some *"rules—of—the—road"* for analog include:

1. Signal filtering (antialiasing and noise limiting)...
must be done prior to sampling i.e. **before multiplexing**.

2. For optimum signal to noise and accuracy, gain (if needed) must be as close physically and electrically to the signal source as possible i.e. gain stages should be first.

3. Multiplexing of analog signals reduces cost per channel by time-sharing expensive components (amplifiers, sample and holds, ADCs, and digital logic associated with ADC).

3.1.1 Input Network Considerations

The input networks play a critical role. The function of the input network includes keeping high frequency *noise* out of the front-end, limiting signal bandwidth to a value consistent with the sampling rate and accuracy requirements, and providing any needed gain. Some *typical* input networks are illustrated in Figure 3.1.

The lower drawing in Figure 3.1 provides the optimum performance and accuracy. It includes an instrumentation amplifier for each channel with only a passive RC filter to limit high frequency noise reaching the instrumentation amplifier. The noise limiting network attenuates the frequency range above which the instrumentation amplifier begins to loose its common mode rejection. The optional trifilar transformer can be used to further limit high frequency noise for applications measuring very low-level signals in noisy environments with high accuracy.

The active lowpass filter following the instrumentation amplifier (and gain stage) limits the signal bandwidth and minimizes the errors introduced by signal aliasing. To minimize noise, the active filter is placed after the instrumentation amplifier and first gain stage. Since it is an active filter a sharp cutoff can be used for signals with significant frequency components above the Nyquist frequency (half the sampling rate). This can be particularly important when sampling high bandwidth signals since oversampling drives up the system cost because of the need to acquire data at considerably higher throughputs (faster buses, more critical timing, faster storage media, etc.).

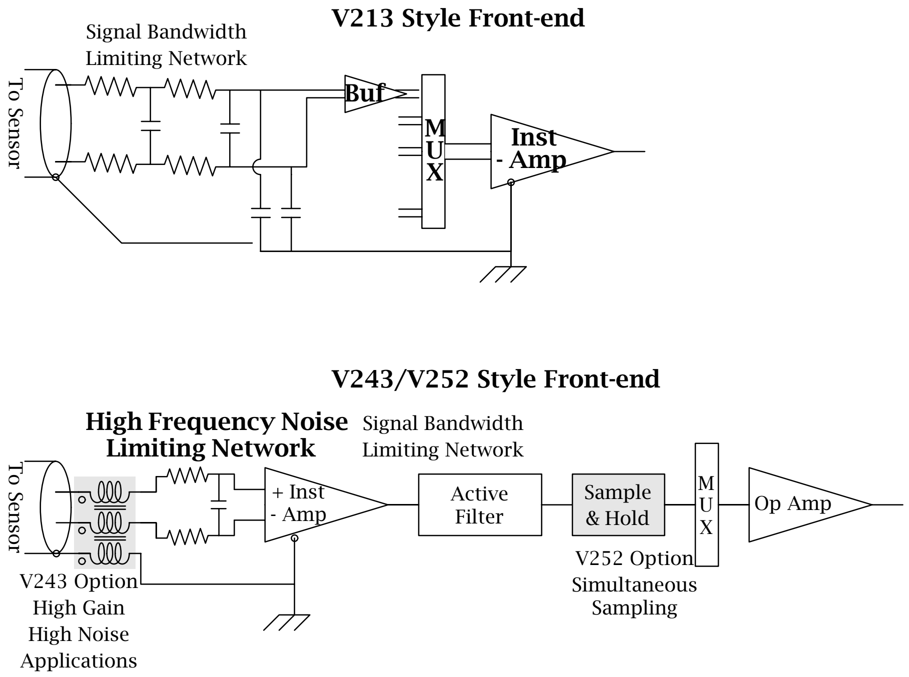


Figure 3.1: ADC Input Networks

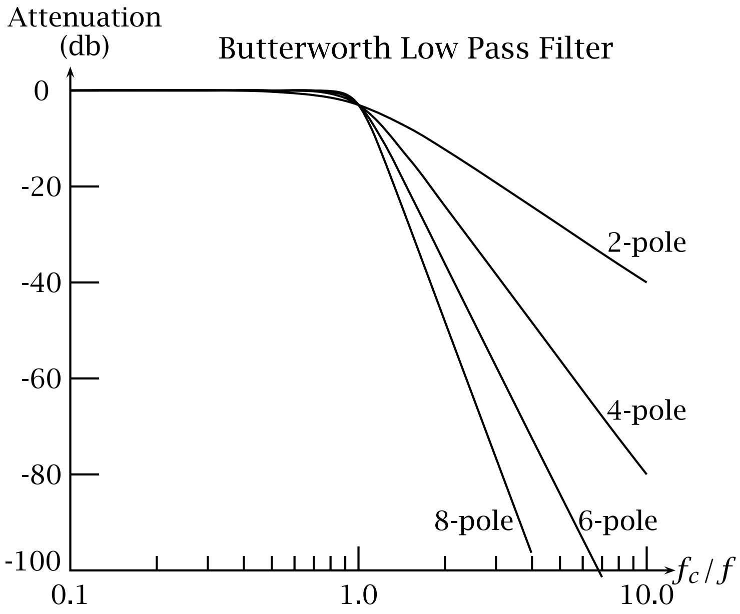


Figure 3.2: Filter Attenuation

The downside to the lower drawing in Figure 3.1 is cost. Instrumentation amplifiers and active filters per channel drive up the cost per channel as well as the cost of the trifilar transformer.

An alternative approach is illustrated in the upper drawing of Figure 3.1. Here as many of the expensive components are moved behind the signal multiplexor (the instrumentation amplifier and associated gain). This way the instrumentation amplifier expense incurred once per group of channels rather than on a per channel basis. Since bandwidth limiting filtering *must be done prior to sampling —which is in effect at the multiplexor*, a simple passive 2-pole RC filter is used to minimize cost. This choice limits the available filter rolloff rate, and results in the need to sample the signal at higher rates to avoid aliasing errors.

The effects of different filter rolloff rates are illustrated in Figure 3.2 for 2, 4, 6, and 8-pole filters. For example, to get a 100:1 attenuation of signals above the Nyquist frequency, one must sample at 20 times the filter cutoff frequency (pass band) with a 2-pole Butterworth filter, and 4 times the filter cutoff frequency with a 8-pole Butterworth filter. If the goal is 0.1% measure-

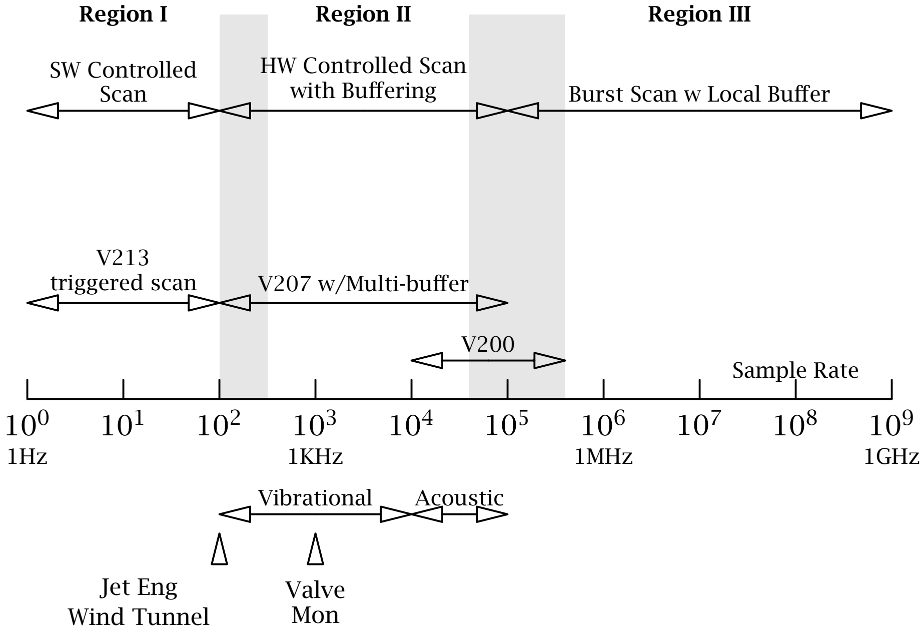


Figure 3.3: Application "Regions"

ment accuracy, then the 100:1 attenuation above the Nyquist frequency is adequate for applications where the frequency components above the Nyquist frequency are at least 20 dB below those in the pass band.

3.2 ADC Family

For purposes of meeting a wide range of applications, KineticSystems has designed a number of *families* of analog I/O modules to meet different kinds of application. Figure 3.3 illustrates the various "regions" and the product family that is designed to provide the optimum price and performance. The *regions* are basically divided by sampling rate or in effect signal bandwidth. Depending on the specifics of the application there can be considerable *gray area* between different regions.

Figure 3.4 summarizes the various ADC families and their capabilities including sample rates and signal bandwidths for various numbers of channels and filter selections.

	ADC &	ADC	No Ch	Sample	Nyquist	Filter	Effective Bandwidth	
	Signal Cond	rate	/ADC	Rate	(f_n)		-20dB @ f_n	-40dB @ f_n
R3	V200	12.5M	1	12.5M	6.25 MHz	3R	—	—
				(195K)	97.5KHz	FIR	—	90.6 KHz
R2	V207/V252	500K	8	50K	25 KHz	8B	18 KHz	12.5 KHz
			16	20K	10 KHz		7 KHz	5.0 KHz
	V207/V246	500K	8	50K	25 KHz	6C	21 KHz	14.0 KHz
			16	20K	10 KHz		8 KHz	5.8 KHz
R1	V208/V243	100K	32	2K	1 KHz	2B	312 Hz	100 Hz
			96	1K	500 Hz		156 Hz	50 Hz
	V213 (50K)	50K	16	2K	1 KHz	2R	250 Hz	71 Hz
	(gain 1-20)		32	1K	500 Hz		125 Hz	36 Hz
	V213 (20K)	20K	16	1K	500 Hz	2R	125 Hz	36 Hz
	(gain 1-2000)		32	500	250 Hz		62 Hz	18 Hz
	V213 (2K)	2K	16	100	50 Hz	2R	12 Hz	3.5 Hz
	(low noise)		32	50	25 Hz		6.2 Hz	1.7 Hz

8B	8-pole	Butterworth (1.35, 2.0)	2B	2-pole	Butterworth (3.2, 10)
6C	6-pole	Chebyshev (1.2, 1.7)	2R	2-pole	RC network (4.0, 14)
FIR	Digital	Finite Impulse Response	3R	3-pole	RC Network

Figure 3.4: ADC Performance Comparison

3.3 The V207/V208 Family and MUXbus

The V207/V208 and associated signal conditioning modules are based on the concept of a multiplexed analog bus over the VXI Local Bus or “MUXbus”. The MUXbus provides a means to connect a variety of signal conditioning modules to the ADC without inter-module wiring on the front panel.

The MUXbus analog architecture is based on a four channel wide differential analog bus implemented over the VXI Local Bus. The ADC module receives each channel with a differential instrumentation amplifier and samples each of the four channels in sequence. This approach is used to provide settling time on each bus for analog signals prior to being multiplexed to the ADC for digitization. Signal conditioning and multiplexing is handled by separate signal conditioning modules.

The family consists of the V207 and V208 ADCs, the V241 High Level Scanning MUX, the V243 Low Level MUX, the V246 Bridge Signal Conditioner, the V252 and V253 Gain/Filter modules.

3.3.1 MUXbus Architecture

The MUXbus consists of a four channel differential bus implemented over the VXI local bus. Signal conditioning modules multiplex analog signals from input channels onto the bus under control of the ADC. The architecture provides timing and control for up to 256 (2048 for V208/V243) analog channels. The bus provides timing signals to step to next channel address, and a signal designating the start of a “frame.” Signal conditioning modules include an 8-bit (11-bit) “address” counter which is incremented by the MUXbus timing strobe and reset at the start of a scan by the first timing strobe which is twice as wide as the stepping strobes.

The MUXbus cycle is illustrated in Figure 3.5. Basically the MUXbus operates as a four phase analog bus. The use of four phases provides extra settling time for each of the channels before it is sampled and digitized at the ADC. Each of the four analog buses are designated by the letters A—D.

Signal conditioning modules and ADC contain a *scan table.* The tables determine the order the channels are scanned and when the analog signal is enabled on the appropriate MUXbus channel. On a given signal conditioning module channels 0, 4, 8, 12,. . . are associated with MUXbus channel A. Channels 1, 5, 9, 13,. . . with channel B, etc. The order of the channels in the scan list *must be ordered such that analog signals are enabled on the MUXbus channels in the order (A, B, C, D, A, B,. . .).* To put it another way the lower order two bits

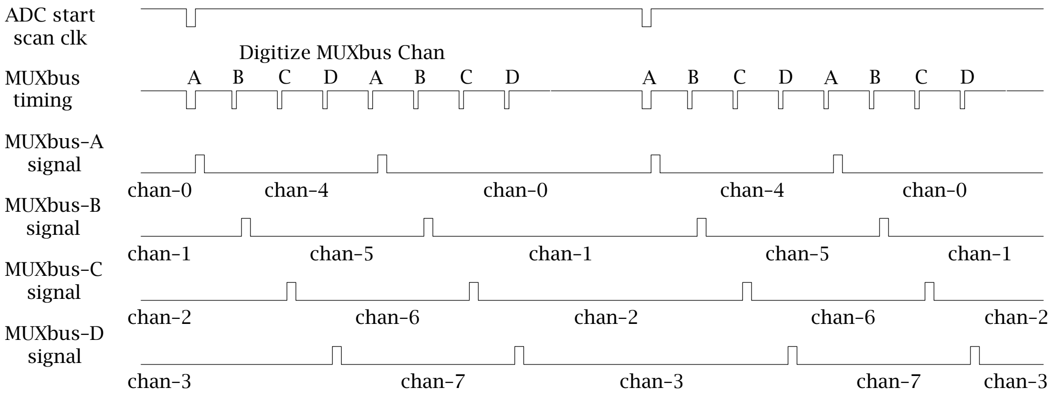

ADC start scan clk
Digitize MUXbus Chan
MUXbus timing
A B C D A B C D A B C D A B C D
MUXbus-A signal
chan-0 chan-4 chan-0 chan-4 chan-0
MUXbus-B signal
chan-1 chan-5 chan-1 chan-5 chan-1
MUXbus-C signal
chan-2 chan-6 chan-2 chan-6 chan-2
MUXbus-D signal
chan-3 chan-7 chan-3 chan-7 chan-3

Figure 3.5: MUXbus Timing with 8 scanned channels 0-7

of the channel number in the scan list must follow the sequence 00, 01, 10, 11, 00,.... Other than this restriction channels may be scanned in any order across a group of signal conditioning module on a common MUXbus.

Basically the scan list is a list of channel numbers in the order to be scanned. The best strategy in building scan lists for multiple modules is to generate a prototype scan list that lists the channel numbers for all the signal conditioning modules on the common MUXbus *in the order to be scanned.* Then just prior to loading a copy of the prototype scan list turn on the appropriate "enable" bits for the signal conditioning module to be loaded. In applications with multiple signal conditioning module on a common MUXbus, each signal conditioning module goes through all the switching steps for each channel in the scan list except it only enables the signal on the MUXbus if the enable bit is set in the list.

3.3.2 V207/V208 ADC

A general block diagram of the V207/V208 ADCs is illustrated in figure 3.6. Each of the four analog bus lines are received by a high quality differential instrumentation amplifier. This approach minimizes noise pickup on the backplane. Also only high level (±10 V) analog signals are transmitted over MUXbus which provides further noise immunity. All signal gain is provided in the signal conditioning modules prior to multiplexing the signal onto the MUXbus.

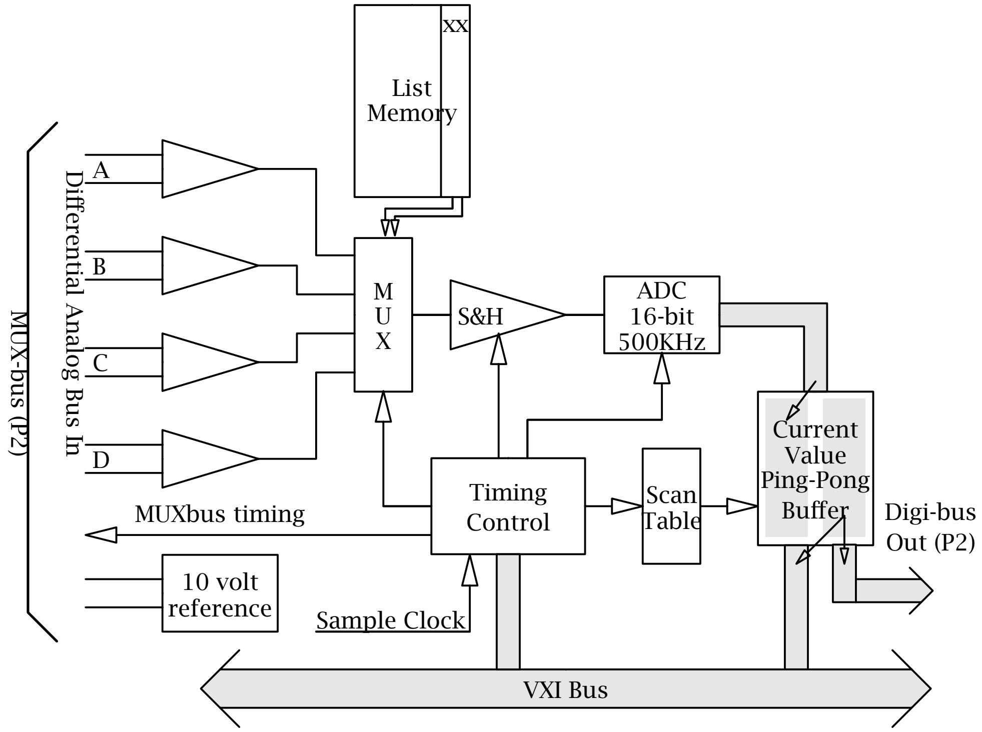


Figure 3.6: Block diagram of wideband ADC

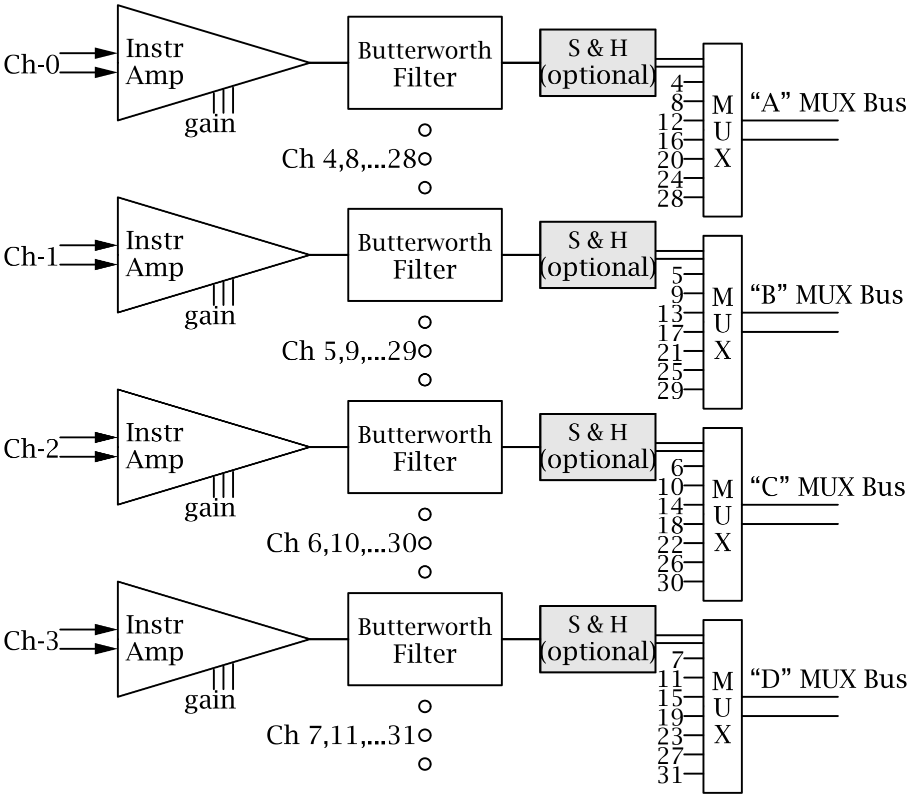


Figure 3.7: Block diagram of typical signal conditioning module

3.3.3 Signal Conditioning

A *typical* signal conditioning module is illustrated in Figure 3.7.

3.4 Calibration

The typical analog input circuit is illustrated in Figure 3.8. A precision 10 volt reference is provided in both the ADC and the Signal Conditioning module. The reference source is calibrated against a NIST traceable DVM at the factory. For applications requiring NIST traceable measurements it is recommended that the precision reference be periodically calibrated with a NIST traceable

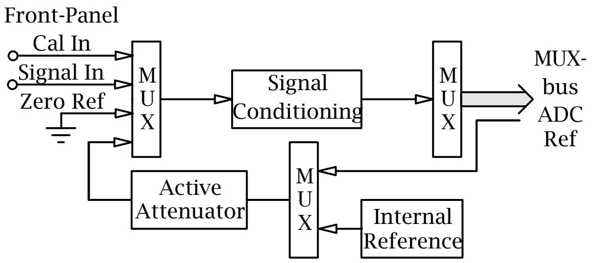


Figure 3.8: Typical calibration configuration

DVM.

The reference signal is divided down by a programmable precision divider. With this combination a precision reference voltage from ±5mV to ±10V can be injected into the analog input circuitry under program control.

The recommended calibration procedure is to make a calibration measurement just prior to taking data, but only after the equipment has been running for about 1/2 hour and has had time to reach thermal equilibrium with the environment.

The calibration procedure is to set up the gains and filters that are to be used in the measurement. Then select +full scale reference and make approximately 20 to 50 measurements. Repeat with −full scale reference and with shorted input. Average each of the groups of measurements. The shorted input gives the offset, and the slope is determined from the full scale + and − readings. A detailed discussion is included in Appendix A.

Chapter 4

Sigma-Delta ADC Architecture

The purpose of this Chapter is to acquaint the user with sigma-delta ADC technology, its advantages, and how it is implemented in the V200.

The sigma-delta ADC can be thought of as a very high sampling rate ADC with very poor (1-bit) resolution. The bitstream from the ADC is then *averaged and down-sampled* to achieve improved resolution at a lower *effective* sampling rate. The *averaging* is accomplished with a Finite Impulse Response (FIR) digital filter.

Recent advances in high speed digital electronics has made the sigma-delta ADC architecture competitive with more traditional ADC architectures.

Radical differences in sigma-delta over conventional ADC technology requires re-thinking a number of technical considerations. These include:

Gain vs Input Span The term *gain* has meaning when it is understood that the input to the ADC is ±10 Volts. With sigma-delta ADCs the input to the ADC is typically not ±10 Volts so the relevant parameter is *input span* instead of *gain*.

Oversampling and Antialias Filtering In conventional ADCs it is frequently necessary to sample at the lowest feasible rate to bound system throughput. When signal aliasing is of concern very sharp low-pass analog filters are typically required. The closer the Nyquist frequency (sampling-rate/2) is to the upper end of the passband the sharper the filter requirement. The high (64:1) oversampling rate of the sigma-delta greatly reduces the need for sharp analog filters. Digital filtering and decimation internal to the sigma-delta minimize system throughput requirements for a given system bandwidth requirement.

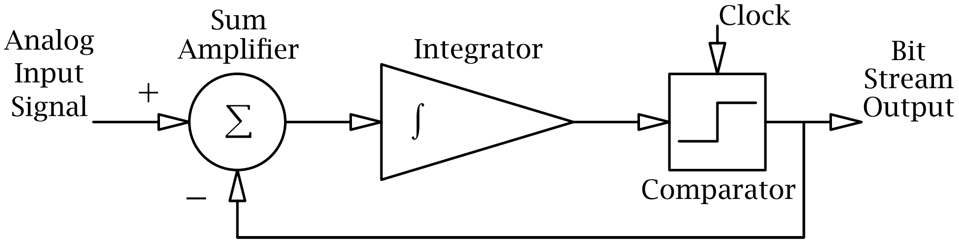


Figure 4.1: Typical Sigma-Delta Architecture

"Simultaneous" Sampling In conventional multiplexed or ADC per channel configurations *simultaneous sampling* (sample and hold per channel) is required when relative phase or samples between different channels is to be compared. The oversampling and "digital averaging" of the sigma-delta presents unique differences in the concept of *simultaneous sampling.*

4.1 Sigma-Delta Operation

The typical sigma-delta ADC is illustrated in Figure 4.1. The analog input signal and a bitstream whose bit density of 1's bits is a representation of the magnitude of the analog signal are fed into a summing amplifier. This is then integrated and enters a comparator which outputs a 0 or 1 depending whether the output of the integrator is below or above the comparators threshold.

The simplest way to understand the operation is to assume a slight variation from a steady state condition and trace what happens. For example, assuming a positive analog input, if the *average* 1's bitstream density at the negative input of the summing amplifier is high compared to the analog input, then the output of the summing amplifier will be negative. This over time results in a corresponding drop in integrator output and the comparator output generating 0's. The 0's in turn cause increased positive output of the summing amplifier which over time results in higher integrator output, and 1's at the comparator output.

It should be clear from the above discussion that the *average* density of the bitstream must closely track the analog input signal. Any deviation will quickly result in the average bitstream density from the comparator being adjusted to follow the analog input signal. The comparator is in effect a very high gain amplifier whose output is driven to “1” or “0” very quickly depending on the difference between the input and the reference.

This action forms a strong, high gain, negative feedback loop which gives the sigma-delta ADC excellent linearity and no missing codes. It also minimizes the effects of component drift with time making the sigma-delta very stable compared with its “open loop cousins.”

4.2 Oversampling, Averaging, and Resolution

The sigma-delta modulator discussed above is essentially an ADC with one bit resolution. Resolution can be increased by averaging. Thus by averaging the output of the sigma-delta modulator the resolution of the analog signal level can be improved. A low-pass digital filter is a form of averaging. In fact an N-point average is no more than an N-tap Finite Impulse Response (FIR) digital filter with coefficients of 1/N.

In the Analog Devices AD-7722, which is the sigma-delta ADC that is used in the V200, there are two FIR filters. The first is a 384 tap filter that samples the output of the sigma-delta modulator at the clock frequency f_{clkin} followed by a second 151 tap filter that samples the output of the first filter at $f_{clkin}/32$ and decimates the output by a factor of two. In effect the second filter averages over $151 \times 32 = 4832$ analog samples to achieve 16-bits of resolution and the *effective* sample rate is $f_{clkin}/64$.

4.2.1 Antialias Filtering

The V200 includes a 2-pole RC antialias filter and a two stage digital FIR decimating filter. The high 64:1 oversampling ratio of the sigma-delta ADC pushes the first band of frequencies aliased into the 0 — 92 KHz passband out to the 12.8 MHz region when the V200 is operated at a maximum effective sample rate of 200 Ksamples/second ($f_{clkin} = 12.8$ MHz). Refer to Figure 4.2.

The 2-pole RC antialias filter ahead of the sigma-delta provides >54dB of attenuation in the 12.8 MHz region. This is adequate for most applications since it rare for the signal source to have significant frequency components at 64 times the upper end of the frequency region of interest (passband). The digital filters provide high attenuation (>90 dB) for signals in the stopband above 106.34 KHz. This includes aliased signals which fall within the stopband. Furthermore the FIR digital filters provide excellent channel to channel matching and linear phase response.

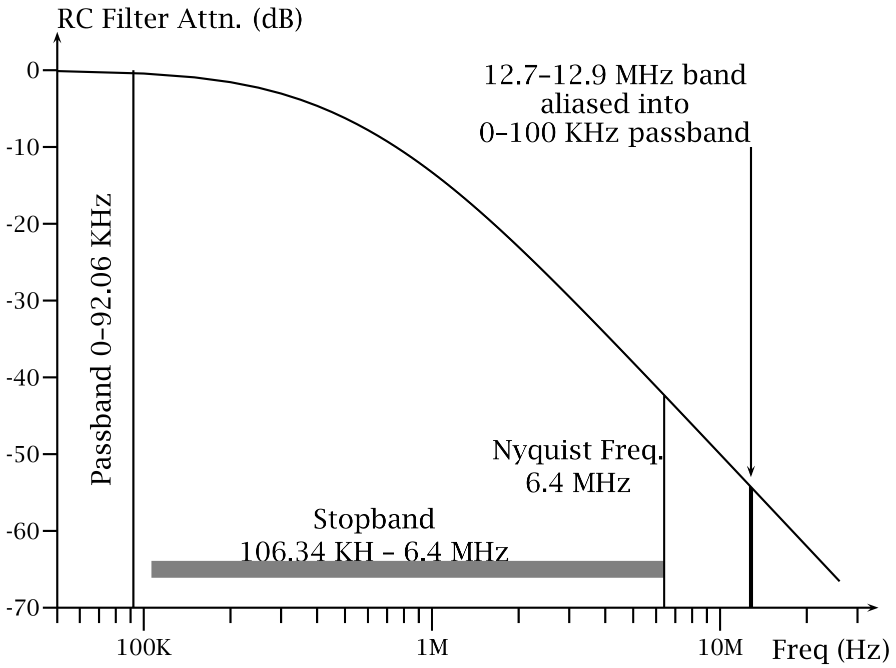


Figure 4.2: V200 2-Pole RC Antialias Filter Characteristics

This contrasts with a conventional ADC where sampling rate and throughput limitations force minimum sampling rates and thus the Nyquist frequency to be as close to the upper passband edge as possible. Since some transducers can have significant signal power above the region of interest, lowpass filters with much higher attenuation at the lower aliased frequencies are frequently required.

The decimating FIR digital filter in the sigma-delta ADC provides a very sharp cutoff above 92 KHz passband, and >90dB attenuation in the stopband above 106.34 KHz when operated at an effective sample rate of 200 samples/sec.

As a result of the high oversampling rate of the sigma-delta, a simple two pole RC filter provides excellent antialias protection in most cases. Several factors contribute. The high oversampling rate pushes the band of frequencies aliased into the passband out to high frequencies where the RC filter has good attenuation (refer to Figure 4.2. Also at this higher frequency the original signal is likely to have significantly smaller frequency components. Finally the digital filter which is part of the sigma-delta highly attenuates signals above the stopband edge.

4.3 "Simultaneous Sampling" with Sigma-Delta ADCs

Due to the architecture of the sigma-delta ADC which uses digital filtering and decimation techniques, the concept of *simultaneous sampling* loses its literal meaning since any given output sample is a weighted average over some number of input samples. Thus there is no well defined *time* when the analog signal was "sampled." This, however **does not** preclude the output of two different channels being time or phase related.

Running sigma-delta ADCs from a common clock is sufficient to guarantee a fixed phase relationship between the channels. By also resetting the ADCs from a common reset signal the absolute phase difference between channels can be eliminated.

In most applications the *real requirement* is either a stable or zero phase relationship between different channels **not simultaneous sampling.** Simultaneous sampling is only *one* means to an end that works for conventional ADCs. Obviously sigma-delta ADCs *can meet* the *real* requirement.

In a few applications there is a need to establish a "true" zero time relative to some event. For example the firing of a laser or other event. With conventional ADCs one approach is to correlate the "sample" signal with the external event to establish t=0. With sigma-delta ADCs this is not usually feasible. An alternative approach which works with sigma-delta as well as conventional ADCs is to dedicate one analog channel to sampling the trigger signal. This is probably the most accurate approach for either type of ADC.

Chapter 5

Digital Architecture

Up to this point we have defined an analog front-end architecture. The ADC may be thought of as generating a “digital stream” of data synchronous with the “tick” of the “sample” clock. A variety of applications require further buffering and/or processing of the data. Also in some applications it is necessary to introduce discrete digital data into the same data stream as the analog data that is “sampled” synchronously with the analog channels.

If the data is processed in realtime (e. g. a DSP), then a “processed” digital data stream may result. For some applications it may be necessary to drive digital or analog outputs with the processed data stream for example control applications. Also it may be desirable to stream either the raw or processed data stream to the Slot-0 controller for transfer to a workstation or archival to permanent storage.

For streaming data from, for example, an ADC, to other system components KineticSystems has developed a very simple “synchronous” bus... *the DigiBus*. The advantages of such a bus is that it is not subject to timing uncertainties of a fully arbitrated bus with handshaking such as VXI backplane bus. The KineticSystems implementation of *DigiBus* is a dedicated synchronous bus with a defined maximum throughput of 10 Mbytes/second. It is implemented over the VXI “local” bus, and extends as far as adjacent modules in the VXI backplane propagate the bus. Implicit in this structure is that transfers on a given bus segment are synchronized by a common clock.

Figure 5.1 illustrates the concept. The ADC may be thought of as the primary source of the data stream with the *Digital In* providing additional discrete data to the stream (one or more modules). The Multi-Buffer module provides multi-buffered access to the data via the VXI bus, and the DSP provides local

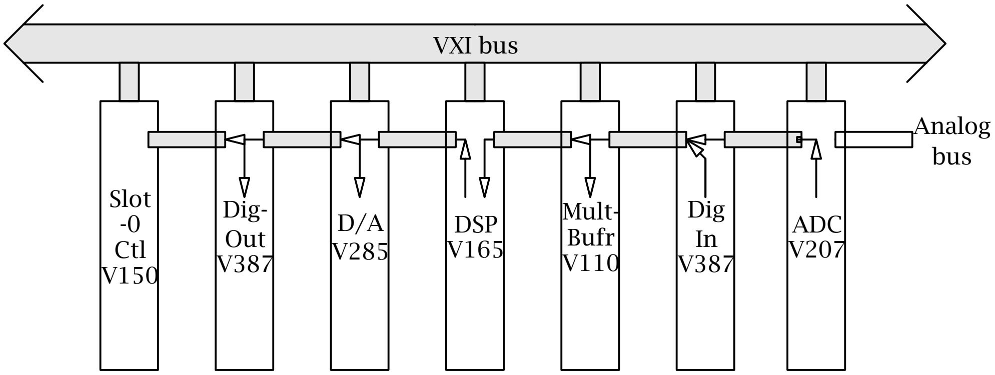


Figure 5.1: Functional Digital Wideband Architecture

processing of the data stream in realtime. The *input* data stream terminates at the DSP with the DSP software determining what is optionally passed on to the left. The D/A and Digital Output modules may select items from the data stream for output, and the Slot-0 Controller may buffer the data stream directly for transfer to a host (for example the Grand Interconnect) or write the data to disk (V150 Slot-0 Controller).

5.1 Digi-bus Applications

It is anticipated that most applications will implement only a subset of this architecture. For example, many applications require just the ADC and Multi-buffer. In this type application data from the ADC is stored in a large segmented circular RAM buffer that is accessible from the VXI bus. Each time the sample clock “ticks” a block of data representing the digitized data from each “active” channel is transferred to the multi-buffer. The multi-buffer *selects* some or all the data and stores it sequentially in the buffer. When a buffer segment fills an interrupt is generated signaling the Slot-0 Controller that the buffer segment is full and ready to be read. While the processor is servicing the interrupt and moving the data, the Multi-buffer is buffering additional blocks of data into other buffer segments. This technique insures that no data is lost as long as the host computer can (on the average) accept data at least as fast as the ADCs produce data.

Since the Multi-buffer can be provided in relatively large sizes of 1 Msample or more, the computer latency for responding to an interrupt is not critical. The multi-buffer approach permits the use of moderate sized buffer segments that minimize average latency for data access. The ability to select items from the data stream enables the user to select out only those data channels that are of current interest without having to reconfigure the input signals. Also multiple multi-buffer modules can be used to separate out different data streams that are needed for different purposes.

Other applications may require an ADC and DSP. In this configuration the DSP can perform a variety of functions including selecting data based on some criteria (e.g. a valve activation in valve monitoring), monitor the data stream based on some limit threshold criteria, perform engineering units conversion, perform FFTs on the data stream, perform digital filtering operations and decimation on the data stream, do signal averaging operations, and many other functions. The DSP provides a large shared RAM memory with the VXI bus for data buffering and inter-processor communications.

Other applications may require a D/A added to the configuration above to provide synchronous data output. An example might be the need for an analog signal representing an input channel after it had been digitally filtered by the DSP, or a very high performance PID loop control.

The architecture provides a mix-and-match capability to meet a wide variety of applications. For example the Valve Monitoring application might consist of the ADC, Digital-In, and DSP (with buffer option).

5.2 Common Architectural Features

Several features are common across the architecture and will be discussed in this section. These include the use of trigger lines, synchronization across multiple modules, selection of data from the data stream.

5.2.1 VXI Trigger Line Usage

Frequently in data acquisition applications there is a need to synchronize or provide a common clock between groups of modules. All KSC *second generation* modules provide a means to select a clock source under software control from **one** of the following sources:

- any of the 8 VXI trigger lines
- an external clock from the front panel connector

- or an internal clock

When an internal or external clock is selected the module has the capability to drive a selected trigger line or a front panel connector for purposes of synchronizing other modules.

In transient capture applications a "trigger" signal is required to signal modules to capture an event. KSC *second generation* modules can select one of the 8 VXI trigger lines or a front panel connector as the source of the event trigger. When the front panel trigger option is selected the module provides the capability of optionally driving a selected trigger line. The V207/V208 multibuffer option as well as the V110 multibuffer memory can function in such a *transient capture mode*.

5.2.2 Digi-Bus Protocol

The "Digi-Bus" is implemented on the VXI Local Bus and is propagated from slot to slot by appropriate modules. Transfers of "frames" of data over the Digi-Bus is synchronous with the clock line used to drive the ADC and/or Digital-In modules. The right most "source" module is designated as the master. The *master* provides the time base for a frame of data as well as control and termination for the bus. One fixed length frame of data is generated for each tick of the designated clock. Frame length is determined at configuration time by software consistent with the source module capabilities and configuration. The clock "tick" serves as the beginning of a new frame. In a multi-source configuration each source module is configured by software to generate a selected number of data items per clock tick at a selected location in the frame. In effect each module is assigned a "time slot" for transferring its data. The Digi-Bus supports multiple data sources as well as multiple data sinks. The left most sink also terminates the bus.

The unit of data on the Digi-Bus is a 16-bit "word". A 16-bit word accommodates the resolution of most ADCs. Data sources with more than 16-bit resolution are handled by a double word. When transferring data in the double word format the first word is the least significant (low order) 16-bits followed by the most significant (high order) 16-bits (little enden).

The Digi-Bus supports transfer rates up to 10 Mbytes/second (10 MHz) with bus timing controlled by the designated bus master. The right most (source) module serves as bus master and provides all necessary timing and control as well as termination. The left most (sink) module also provides bus termination. Data flow is generally from right to left.

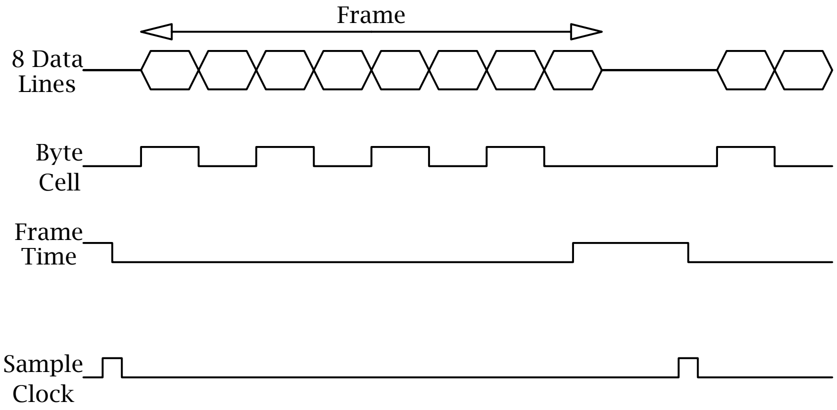


Figure 5.2: Digi-Bus Timing Diagram

In general, modules can be classed as data sources (ADCs, discrete inputs) and data sinks (D/As, discrete outputs, memories and buffers). Some modules like the V165 DSP are both data sources and sinks since they take in data, operate the data, and then pass the transformed data on to the left. In this case the Digi-Bus input is terminated at the DSP and the DSP becomes the *master source* for modules to its left.

ADCs which implement both the MUX-bus and Digi-Bus always function as the Digi-bus master and provide termination. Other data source modules such as the V387 pass the Digi-bus through.

Figure 5.2 Illustrates the timing and use of the local bus lines for the Digi-Bus. Eleven of the 12 VXI Local Bus lines are defined (refer to Table 5.1 for VXI Local Bus pin assignments). Digi-Bus timing is summarized in Table 5.2.

Digi-Bus source modules implement configuration registers which define the time slice "address" used by each active data channel. Each module derives the address by counting the trailing edge of the *byte strobe* and resets the address by the sample clock.

In general ADC modules function as the *master* source module and provide the necessary Digi-Bus timing. ADC modules that implement the **Mux-Bus** to the right, are always Digi-Bus Masters and implement the Digi-Bus to the left. ADCs that do not implement the **Mux-Bus** optionally pass the Digi-Bus through, and have the option of being Digi-Bus master.

Source modules that do not implement the Mux-Bus such as Discrete I/O modules optionally propagate the Digi-Bus through. Sink modules such as memories, FIFOs, etc. optionally terminate the Digi-Bus. Processors such as

Table 5.1: Digi-Bus pin assignments.

Usage	Left Side	Right Side
DB7	A15	C15
DB6	A14	C14
DB5	A12	C12
DB4	A11	C11
DB3	A9	C9
DB2	A8	C8
DB1	A6	C6
DB0	A5	C5
Byte Cell	A17	C17
Frame Time	A18	C18
Sample Clock	A21	C21
spare	A20	C20

8 Data Lines: Eight lines are allocated for data. Data transfers are byte serial with two bytes per "channel" or data word. Devices requiring 32-bit words would allocate two logical channels. Transfers are in the order *lower* byte followed by *upper* byte (little enden). Data lines are high true and are driven by the addressed source module.

Byte Cell Strobe: The leading edge of this line designates the start of the "byte cell time." The "source" module must assert the low-order byte on the data lines on the rising edge and the high-order byte on the falling edge of the byte cell strobe. In both cases data must be stable on the data lines between 20 and 80ns of the 100ns byte cell time. Consumer modules should strobe data during the 60ns period while data is stable. This line is driven by the **master** source module.

Frame Time Strobe: The Frame Time Strobe is generated by the **master** source module and the negative edge designates the start of a frame. The trailing (positive) edge designates the end of a frame. Frame Time must be asserted a minimum of 50ns before assertion of Byte Cell and cannot be negated until 50ns following the trailing edge of the last byte cell of the frame. Furthermore Frame Time cannot be re-asserted for a minimum of 100ns. This line is driven by the **master** source module.

Sample Clock The Sample Clock is generated by the **master** source module and may be used by any slave source module as the clock to sample the input signal(s). This line is driven by the **master** source module.

Table 5.2: Digi-Bus Timing Summary.

Sources of Frame Time & Byte Cell:

1. Frame Time must be asserted a minimum of 50ns before initial assertion of Byte Cell.
2. Frame Time cannot be negated until $CellTime + 50ns$ minimum after last trailing edge of Byte Cell for the last cell in the current frame.
3. Frame Time cannot be asserted until it has remained negated for 100ns.

Data Sources:

Devices sourcing data onto the local bus must provide stable data a maximum of 20ns into the cell time and must maintain valid data until a maximum of 20ns before the cell ends.

Data Sinks:

Devices receiving data must delay the Byte Cell signal $50ns \pm 20ns$ to guarantee $10ns$ setup and $10ns$ hold time on the incoming device.

Logic Levels:

Data Lines	HIGH-TRUE
Byte Strobe	Both edges are active when Frame Time is asserted
Frame Time	LOW-TRUE

Termination:

Pluggable termination of 180Ω to $+5v$ and 390Ω to ground *must* be provided on any module using the local digital bus.

DSPs that can function both as sources and sinks must terminate the bus to the right, and provide master functionality to the left.

5.2.3 Data Sinks

Data sinks provide the capability of selecting a subset of data from the Digi-Bus on a channel by channel (word) basis as well as selecting every N-th frame. Frame selection is determined on a frame count basis ranging from selecting every frame to every 256th frame. Channels within a frame are selected on a pass/no-pass basis based on a bit pattern where each word in the frame corresponds to a pass/no-pass bit in the bit pattern. The bit pattern is 256 bits long (2048 bits long for modules supporting the extended MUX-bus addressing like the V208). One bit per word up to the maximum frame size of 256 words (2048 words for extended addressing).

The subsetting of data as it enters the data sink modules is an important feature. Frequently it is critical to place selective criteria on the data stream to limit bandwidth and buffering requirements to manageable proportions. For example suppose it is desired to buffer and stream ALL the data to disk, but to be able to monitor several channels in real time. To use the processor to select out the data for monitoring could place an excessive load on the processor as well as on the bandwidth of the interconnect. By selecting a "sample" of the data (maybe every 10th sample) on a few channels from possibly hundreds of channels significantly reduces the bandwidth requirements of the sink module.

5.2.4 Source Module Registers

Digi-Bus "source" modules implement two registers, one to define the offset address and increment in the frame where data is to be stored (Data Source Register) and one to define the frame size and strobe enable (Frame Size Register).

Data Source Register

The *Data Source Register* is a 16-bit register that defines the offset(frame address) and increment (delta) in the frame where the module is to insert data.

15															0
Δ 7	–	–	–	–	–	–	Δ 0	SA 7	–	–	–	–	–	–	SA 0

SA7–SA0 Specifies the start address within the frame for data insertion. $Data = 0$ indicates the first location within the frame.

Δ7–Δ0 Specifies the number of time slots in the frame that are to be skipped before the next data insertion. $\Delta = 0$ indicates data is to be inserted in every time slot, $\Delta = 2$ is every 3rd time slot.

Frame Size Register

The Frame Size Register is a 16-bit register containing the number samples per frame and a strobe enable control bit. This register must be properly loaded when the module is functioning as the *master*.

15															0
0	0	0	0	0	0	0	STB Ena	SF 7	–	–	–	–	–	–	SF 0

SF7–SF0 Specifies the number of samples per frame. This is the number of 16-bit samples (2 bytes) on the digital on the digital local bus. Data =0 indicates 1 sample/frame. Data=255 indicates 256 samples/frame.

STB ENA When 1, strobes are enabled onto the local bus. When 0, strobe lines are tri-stated on the local bus.

5.2.5 Data Sink Registers

Data sinks provide a set of registers to select data. Two mechanisms are used, one to select every N-th frame, and one to select data within a frame.

Frame Data Selection Register

Data sinks provide a block of 16 contiguous 16-bit data words for specifying selection of incoming local bus data. Each bit in a word enables or disables reception of its corresponding sample. Extended MUX-bus modules provide a block of 128 words.

15															0
SMP 15	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SMP 0
SMP 31	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SMP 16

⋮

SMP 255	–	–	–	–	–	–	–	–	–	–	–	–	–	–	SMP 240

Frame Selection Count Register

A 16-bit W/R register is provided for specifying the frame selection count and an enable bit.

15															0
0	0	0	0	0	0	0	CAP Ena	FC 7	–	–	–	–	–	–	FC 0

FC7–FC0 Specify the frame count interval. If data=0, sink will store every frame, data=1 will store every other frame, etc.

CAP ENA Set to 1 to enable data capture, set to 0 disable data capture.

5.3 ADC as Digi-bus Master

The V207 and V208 ADCs function as a data source and are the “master” for the Digi-Bus. As master it determines the Digi-Bus clocking time corresponding to the per channel analog-to-digital conversion plus settling time. Synchronous with the clock, the ADC initiates a new data frame on the Digi-Bus, and gates its data into the frame at the designated part of the frame. Besides the Digi-Bus the ADC provides a “ping-pong” buffer for VXI bus access. With each “scan” clock the ADC swaps buffers so that the data presented to the VXI bus is associated with the previous clock tick.

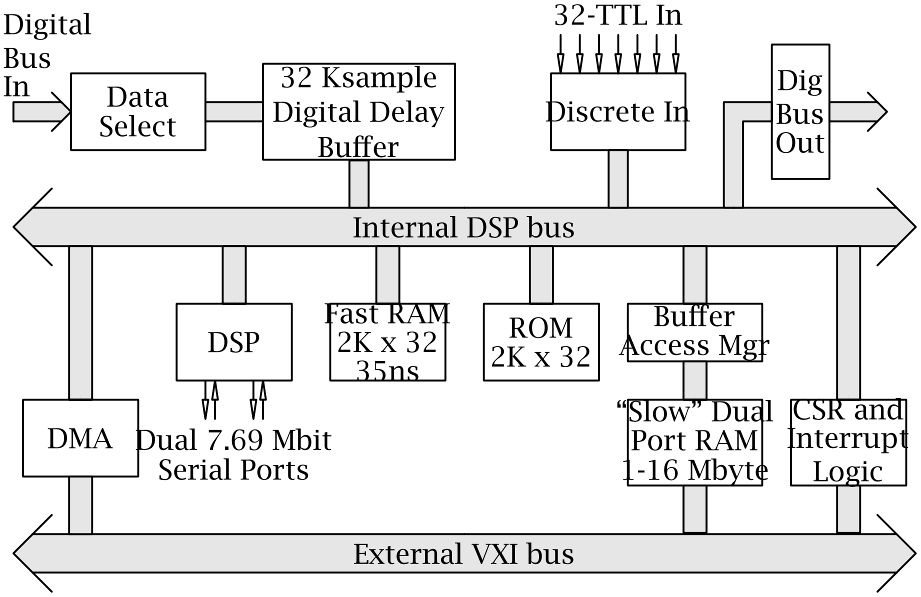


Figure 5.3: DSP Block Diagram

5.4 DSP

The DSP is a key element on the digital side. Some applications need the processing power at the front-end to compute time-critical quantities and minimize the bandwidth requirements of the rest of the system by reducing the data or in some way extracting the pertinent information from the data stream (e.g. signal averaging, digital filtering, extracting data of interest and discarding the rest, etc.); minimizing the "computational bandwidth" of the rest of the system by preprocessing the data (e.g. FFT, cross or auto correlations, conversion to engineering units, limit checking, etc.); or providing high performance realtime response that can only be achieved with a dedicated processor (e.g. safety protection –keeping a machine within safe operating margins, PID or other types of closed loop control, etc.). The logical block diagram of the DSP is illustrated in Figure 5.3 .

The DSP module is architected to provide a high performance "engine" to process a digital data stream either through the Digi-Bus or through the VXI bus. Two important features are provided on the Digi-Bus. First the "Data Select" provides a mechanism to select a subset of data from ever N-th frame. This feature can be used to limit data throughput to the DSP. Second, the 32Ksample Digital Delay Buffer is a circular buffer where data is stored from the Digi-Bus. This buffer can be used as a FIFO or digital delay when used with the TMS320C30's Block-Size Register (BK) and circular addressing. A "current frame pointer" is also provided that holds the address of the most recent frame of data. This mechanism permits the DSP to monitor one or more channels for a signature indicating data of interest while maintaining some "pre-trigger" history.

The DMA controller provides access to other I/O modules on the VXI bus. While this access does not provide the performance that can be achieved by synchronously sending data over the Digi-Bus, it does provide access to the full line of I/O modules.

The "slow" Dual Port RAM is a large (1 Mbyte or 4 Mbyte) memory that is addressable from both the DSP and the VXI bus. Slow is a relative term here as a 200ns memory access time to the DSP is slow when it must stall 7 clock cycles to get a word. The purpose of this memory is to hold data buffers and control tables that both the DSP and VXI bus need to access. The Buffer Access Manager does provide some access enhancements for certain types of access from the DSP. Since it is anticipated that the DSP would be used for collecting, monitoring and buffering data a larger but slower shared memory is anticipated to be a good cost tradeoff. It is anticipated that in most applications, the average throughput will not exceed something on the order of 1 Msample/second. At this rate the DSP only needs to access the shared memory at a $1\mu sec$ interval, thus, a slower lower cost memory that is large is a good compromise.

Finally the architecture includes Control Status Registers (CSR) and interrupt logic so the DSP can be controlled from the VXI bus as well as allowing the DSP to post asynchronous events to other processors on the VXI bus.

5.4.1 DSP, High Speed RAM and ROM

The DSP module is based on a 33MHz Texas Instruments TMS320C30 Digital Signal Processor Chip. Included on the board is a fast 2K x 32 35ns access SRAM. This SRAM is intended for program storage and high high speed data scratch pad. An optional 8K x 32 SRAM is available.

High speed SRAM is an expensive component. It is anticipated that some applications will require additional relatively high speed memory –particularly FFT applications with a requirement for larger sample lengths. FFT algorithms must make multiple accesses to each data point making it desirable to place the data in fast RAM.

Also included is a fast 2K x 32 ROM memory. This memory is intended for bootstrap code and dedicated applications code. Optional 8K x 32 ROM is supported. The DSP and the associated fast SRAM and ROM are designed to allow the DSP to operate at full speed with no wait states.

5.4.2 Dual Port RAM

The 1M x 32-bit Dual Port RAM serves as a "mailbox" between the DSP and other processors on the VXI bus. Since one of the functions of the DSP is to buffer data, a relatively large 1M x 32 or optionally 4M x 32 memory is provided. To minimize cost a relatively standard DRAM is used —something with a access time on the order of 200ns or less. Since the DSP is only accessing this memory for storing data and shared control structures, the need for rapid access can be minimized in most cases through proper software design. DSP access to the memory takes precedence over VXI bus access.

The Buffer Access Manager (BAM) is provided to minimize the impact of the slow memory access. For DSP writes, the BAM latches the address and data and allows the DSP to proceed with out any wait states. If the DSP attempts to store additional data within the memory cycle time the BAM causes the DSP to wait until it can accept the write operation.

Read operations are not as easy. Two read access methods are provided. The first simply causes DSP wait states until the read data is available —this is approximately 7 clock cycles assuming 200ns memory access and 33MHz clock. Fortunately most applications will largely involve write operations.

The second read access method is through an address and data register. When the DSP loads the address register the BAM will initiate a memory read cycle to move the data to the data register. Once the data register has data, the DSP can read the data register. Whenever the data register is read by the DSP the BAM will increment the address and fetch the next item of data. If the DSP attempts to read the data register faster than data can be retrieved from memory, an appropriate number of wait states will be generated to stall the DSP.

This approach helps to minimize slowing the DSP execution because of access time of a slower memory. Since the most frequent operation to the dual port memory is likely to be writes operations interspersed by computation, the buffering of write operations minimizes the need to stall the DSP. The most intense read activity is expected to be applications such as signal averaging. In such cases read access should be sequential where the BAM overlaps DSP computation and fetch of the next word.

5.4.3 CSR and Interrupt Logic

The Control Status Register (CSR) and interrupt logic provide the means for a host computer to initialize and coordinate activity of the DSP. The CSR provides the necessary control and status bits to start, stop, and initialize the DSP. Interrupt logic is provided so that the DSP software can signal via interrupt on the VXI bus.

5.4.4 DMA (Master)

Some applications require the DSP to access to other modules on the VXI bus. A DMA controller is provided that can become bus master and initiate transfers of data to and from other VXI modules.

5.4.5 Data Select

The Data Select feature between the Digi-Bus and the Digital Delay provides a means of selecting a subset of the data stream. This feature can be important in high throughput applications where the DSP does not need to process the entire data stream. This block appears on all "consumer" functions on the Digi-Bus.

Two selection criteria are applied to the data stream. First is a counter controlled gate that passes every N-th data frame where $1 \leq N \leq 256$. The second criteria is a binary pass/no-pass selection on each word within the frame which is controlled by a 256-bit mask. Thus, for example, the Data Select is software configurable to pass the 1st, 7th, and 25th word of every 10th frame as well as pass all or nothing.

5.4.6 Digital *Delay* Buffer

The purpose of the digital delay is to provide a FIFO-like buffer for either holding a fixed time data history (pre-trigger samples when the DSP is functioning as a Transient Recorder), or to buffer data into larger blocks for operations that are block orientated such as the Fast Fourier Transform (FFT).

The Digital Delay Buffer is a 32K x 16-bit RAM which is used to buffer input data from the Digi-Bus. This memory is treated as a circular buffer. The Digi-Bus interface maintains two registers, the "Current Frame Pointer" and "Frame Count" registers. The Current Frame Pointer holds the address of the most recent (complete) Frame. The Local Bus interface updates this register whenever a complete frame has been received. This register can be used to access the most recent data frame. The Frame Count register is incremented by the Local Bus interface logic each time a frame is received and can be decremented or read by the DSP. The function of the Frame Count register is to keep a count of the number of unprocessed frames in the Digital Delay Buffer. When used as a delay or FIFO, the DSP must maintain its own pointer to the next item to be processed, and must check the Frame Count register to insure that data in the circular buffer is not overwritten before it is processed. In the “digital delay” mode the Frame Count register is monitored by the DSP and frames are processed whenever the frame count is greater than the desired delay.

The DSP has read access to the frame counter to determine the frame depth at any time. The counter must provide an accurate frame count over time as frames are added and removed. The frame count read may be one frame short if read occurs just as the counter is incremented as a result of the last word of a frame being placed in the buffer. However a second read yields the correct count.

To maintain frame synchronization the last word of a frame includes a flag bit that can be tested by the DSP.

5.4.7 Digi-Bus Out

Some applications require the DSP to generate a data stream out similar to the ADC. The DSP functions as Digi-Bus “master”. The Digi-Bus Out provides the DSP a mechanism to output a “frame” of data. The output frame is any sequence of data as determined by the DSP code. Unlike the data stream generated by the ADC the DSP data stream timing is determined by when the DSP loads the output register and not a nice equally spaced interval like the conversion time of the ADC.

5.4.8 Digital Input Option

Some applications must sample a digital input as well as analog channels. The architecture defines a separate digital in module with all the bells and whistles. A separate 32-bit TTL input register is provided on the DSP board for applications with simple digital input requirements.

5.5 Ping-Pong Buffer

The ADCs and other input modules in the second generation family provide a "Ping-Pong" buffer for data. At any time one buffer of the pair is being filled by, for example the ADC, and the other is addressable from the VXIbus and is available for reading. When the ADC initiates a scan the Ping-Pong buffer is swapped so that the data from the previous scan is available for reading. This technique provide *a full sample interval* for the processor to read the data.

5.6 Multi-buffer

The Multi-buffer is an important consideration in high sample rate applications. It provides buffering of the data stream. Buffering is typically required at sample rates above 100 Hz depending on details of the operating system and software. The reason is that with out buffering the computer system *must guarantee response* within one sample time or risk loosing data. Most operating system environments have *worst case latencies* well over 1 milli-second.

The multi-buffer is available as an optional daughter card for the V207 and V208 as well as a separate V110 Digi-bus module. The buffer depth is configurable and is typically divided into a minimum of four segments.

Each segment is of equal size where the size can be any value up to the full memory capacity. Once the number of segments and segment size is chosen memory is filled in a circular fashion. Flags for each segment are provided and are set as each memory segment is filled. An attempt to write to a segment with a full flag set (overrun condition) sets the overrun flag. In an overrun condition the data source (ADC) continues writing data into the circular buffer.

Multi-Buffer is a bit like a FIFO buffer except it uses conventional memory and can be subdivided into segments. The advantage is that it is easy to implement much larger buffers than are typically available as FIFO chips *plus* the ability to divide the buffer into more than two segments.

From the ADC or data source perspective, the multi-buffer is simply a large segment of memory with a size that is set under software control from one word (data sample) to the full memory size. The data source treats the memory segment as a circular buffer. That is, when it reaches the end it simply resets its address to the start of the buffer.

From the users prospective the buffer can be segmented into one to typically four "segments" (maximum number of segments is implementation dependent). Each segment is of equal size. When the data source fills a segment

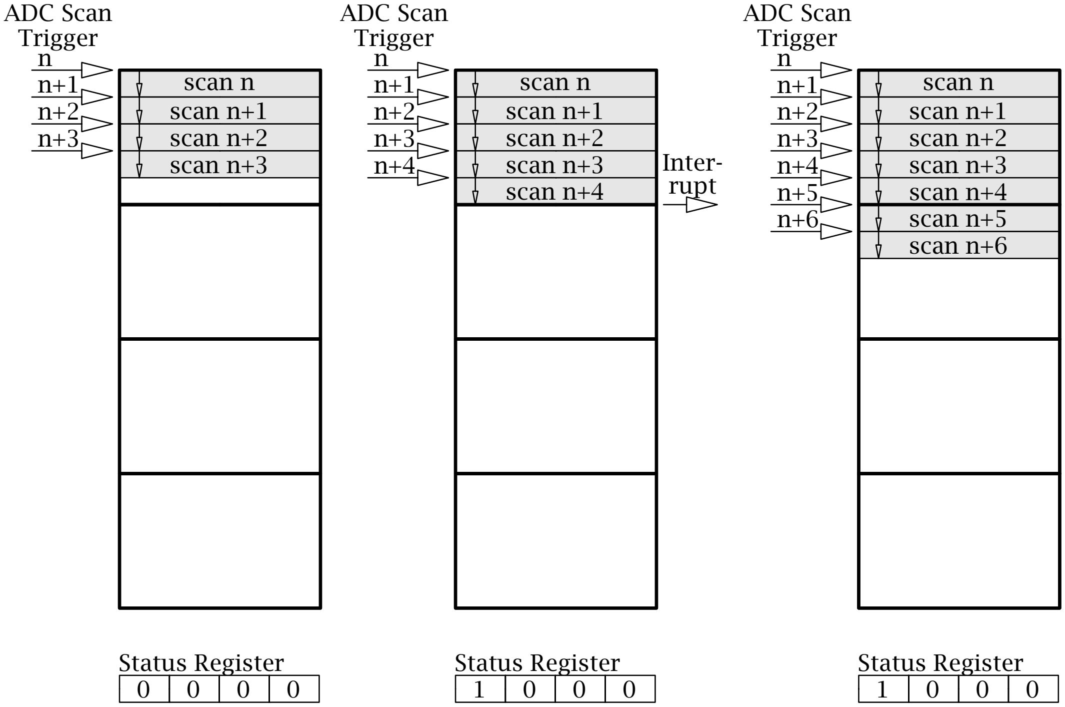


Figure 5.4: Data filling Multi-Buffer, t=3, 4, 6

the multi-buffer signals the user application with an interrupt and sets a *segment full* flag. It is the users responsibility to read out the segment *and reset the segment full flag* before the data source wraps around and starts overwriting the segment with new data. The multi-buffer will set an *overrun flag* if it starts to overwrite a segment whose *segment full flag* has not been cleared.

Figures 5.4 and 5.5 illustrate how multi-buffer operates. The segment size has been chosen to hold exactly 5 scans of data and the multi-buffer is segmented into 4 segments. The first (left most) diagram in Figure 5.4 shows the state at time t=3 with 4 scans of data. The next diagram shows the state at t=4 with 5 scans of data and the segment–1–full interrupt is posted along with the segment–1–full flag set. The right diagram shows the state at t=6 assuming that the computer has not completed reading segment–1.

The left diagram in figure 5.5 illustrates the state at t=9 with the second

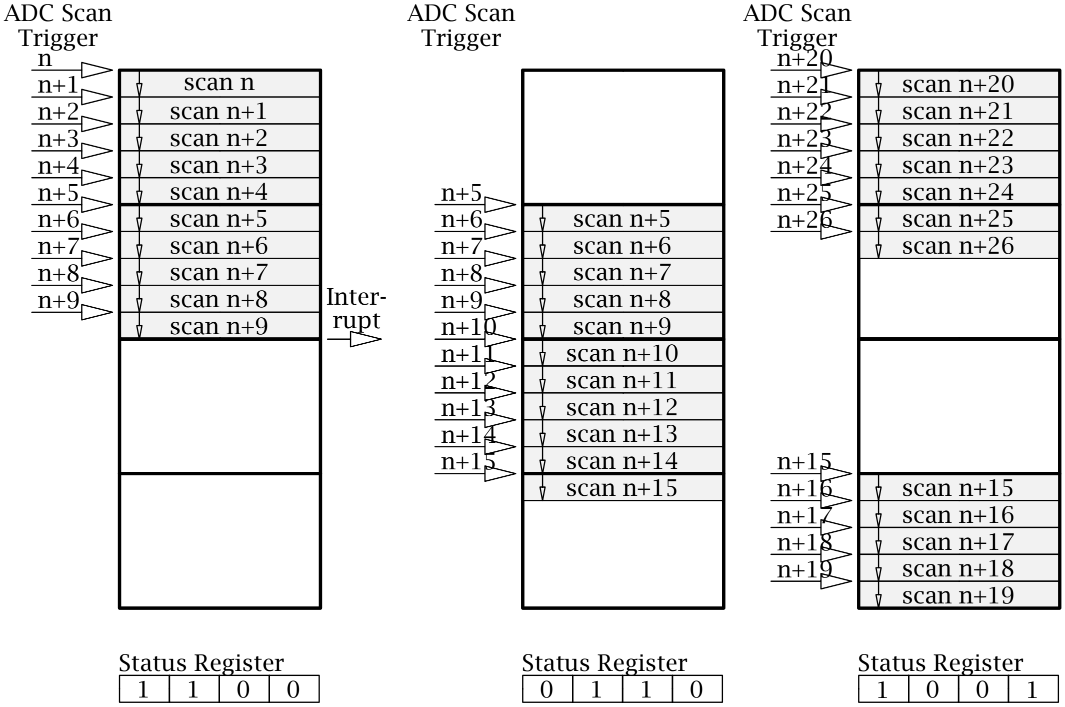


Figure 5.5: Data filling Multi-Buffer, t=9, 15, 26

segment full, a segment full interrupt posted, and neither segment 1 or 2 having been read out i.e. both segment full flags set. The middle diagram illustrates the state at t=15. At this point the computer has read out segment–1 and cleared the segment-1-full flag and the data source has started to fill segment–4. The right diagram illustrates the state at t=26. By this point the computer has read segments 2 and 3 and cleared the respective flags. The data source has started filling segment–2.

5.6.1 Transient Capture Mode

The multi-buffer is also capable of operation in the “transient capture” mode. In this mode the buffer is filled in the same way as before . . . as a circular buffer. When an event signals the presence of a signal of interest, the multi-buffer acquires a preselected number of additional samples and stops. Data is then read starting at the location following the last acquired point i.e. the oldest point first, and then wraps around to read the most recent point last.

5.7 Discrete I/O

In most applications the V387 Digital Input module(s) function as “slave” modules on the Digi-Bus, although it is possible to configure a digital input as master in the event that the application does not include analog data. The digital input module “statusizes” the digital input with the leading edge of the sample clock.

Chapter 6

Computer Interfacing

Several techniques have been used to interface VXI mainframes. The simplest and most direct is an Intelligent Slot-0 Controller where the computer is part of the controller. The others employ some bus or fiber optic link between an interface board in the computer and a “controller” module that interfaces the bus or fiber to the VXI mainframe.

In the latter category is the MXI interface from National Instruments, various GPIB controllers, and the fiber optic Grand Interconnect from KineticSystems.

6.1 Intelligent Slot-0 Controllers

A range of *intelligent* Slot-0 controllers are available from the major VXI manufacturers including KineticSystems, National Instruments, RACAL, and Hewlet Packard. Processors include 68030, 68040, 68060, 486, and HP PA RISC. Supported operating systems include VxWorks, Windows95, Windows NT, Lynx OS. These controllers typically include such options as Ethernet, and SCSI ports and implement the Resource Manager functionality.

6.1.1 V150 Intelligent Slot-0

The V150 is an intelligent slot-0 controller from KineticSystems. It is based on the Heurikon 66 MHz 68040 or 60 MHz 68060 VME boards. The module can be ordered with 8 to 64 Mbytes of DRAM. It includes a 10 MHz clock and a programmable clock derived from the 10 MHz clock with a frequency range from 0.00231 Hz to 500 Khz and periods from 2μs to 429 seconds with a resolution of 100 ns.

6.2 Bused Slot-0 Controllers

6.3 Grand Interconnect

6.3.1 Introduction

The *Grand Interconnect* is a high-throughput, low latency, highly deterministic I/O subsystem designed for high performance distributed data acquisition applications. The interconnect is specifically designed to facilitate synchronous data acquisition over a distributed area while carefully maintaining the time coherency of the data. It operates over a wide range of sampling rates and throughputs as well as collect data from a number of modular standard I/O chassis including VXI, CAMAC (IEEE 583) and VME (IEEE 1014).

This implementation of the interconnect is based on 125 MHz bit-serial technology and a protocol that provides data throughputs to 10 Mbytes/second. The interconnect uses fiber optic technology which provides excellent noise and ground loop isolation between nodes. Fiber optic links up to 2 km between nodes are available using 50 or 62.5 micron cable. The interconnect supports up to 126 nodes or I/O chassis.

Each node controller includes a local "scan list" capability that can be triggered by a local internal crystal-controlled programmable clock, an externally supplied TTL trigger, a local I/O chassis interrupt, or by a host generated trigger message. Each node also provides optional local data buffering.

The interconnect is designed as a single master (host) with up to 126 addressable *slave* nodes. In general, all communication is initiated by the host except that any node may inject an asynchronous event for the host between messages on the highway. This architecture eliminates much of the overhead associated with conventional peer-to-peer networks. This includes contention for network access as well as providing a method for arbitrating network mastership. This architecture results in a greatly simplified protocol and permits implementation of the link management in hardware logic which results in higher performance and lower latencies than competing alternatives. The ability for a node to inject asynchronous event messages into the serial stream minimizes latencies associated with reporting of asynchronous events at the node level.

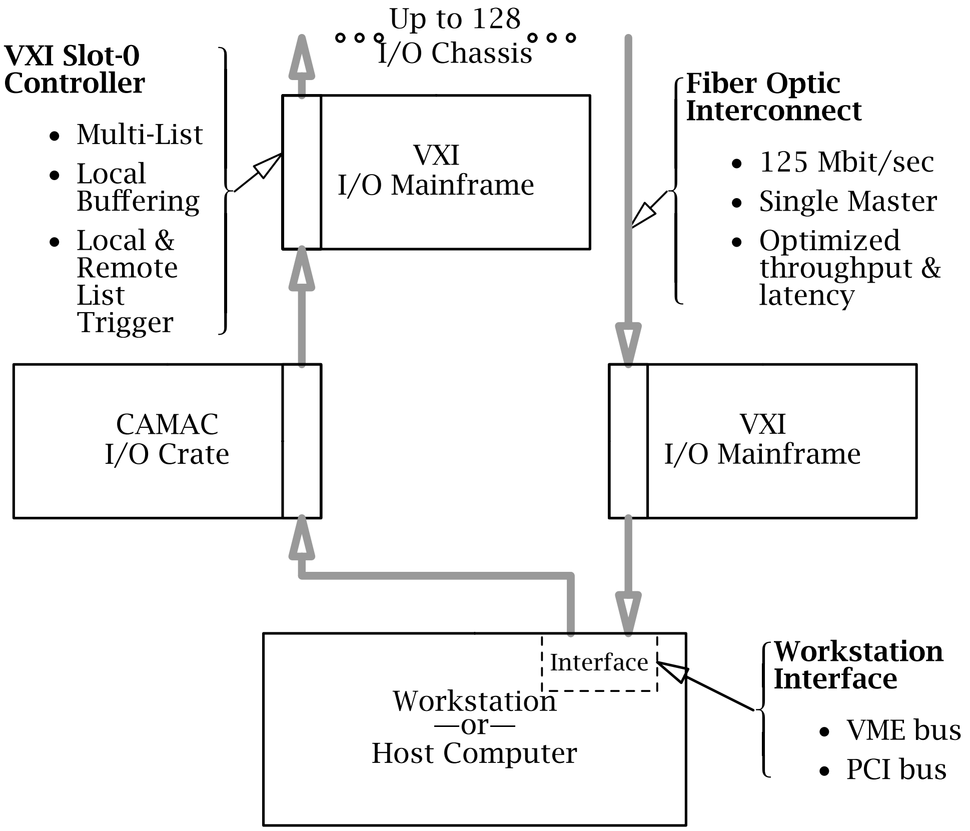


Figure 6.1: *Grand Interconnect* Ring Architecture

One of the features of this architecture is that slave controllers can implement the interconnect protocol in hardware logic due to its simplicity. This feature minimizes node costs since a high performance processor is not needed to support a networking protocol.

6.3.2 Architectural Overview

The *Grand Interconnect* is implemented as a “ring” architecture with a single host which functions as the controlling node. Refer to Figure 6.1. The single *master node* eliminates contention and arbitration issues associated with conventional peer-to-peer networks. The implementation is based on 125 MHz bit-serial technology. For this architecture, a relatively simple protocol can be used which provides a maximum effective *data throughput* of 10 Mbytes/second for large block transfers. Local buffering at the I/O chassis level facilitates transfer of large data blocks, and the simplified protocol eliminates the need for complex network protocol firmware.

To facilitate the data acquisition process, each node includes a firmware-based "list processor" that can be triggered to execute a local "scan list" and acquire data either into a local buffer or respond to a physical I/O request on the serial interconnect in response to a host I/O request. This feature provides a built-in low-latency scatter-gather capability that can respond to I/O chassis events in well under 1 microsecond with a very high degree of determinism. The current implementation of the list processor is based on the Texas Instruments TMS320C25 40 MHz DSP. The choice of using a DSP for the list processing element is governed by the low cost, high speed, and flexibility that is achieved through firmware updates. Through use of custom firmware and extensions to the scan list structure it will be possible to meet many applications with special requirements.

6.3.3 Data Acquisition System Considerations

The *Grand Interconnect* is specifically designed to provide the features needed in high performance data acquisition as well as to provide this capability for physically distributed applications. The critical features for such applications are summarized in the following sections.

Capturing Raw Data

It is generally good practice to capture the raw data and record it for later detailed analysis. There are several good reasons for recording raw data as opposed to processed data. These include:

- Raw data is frequently more compact requiring less storage space and lower system throughput. A typical 16-bit ADC generates 2 bytes of raw data for each conversion. When converted to engineering units it takes 4 bytes (32-bit floating point) to 8 bytes (64-bit floating point) to represent the converted data sample.

- Once data is converted (e.g. to engineering units based on calibration information available at acquisition time) it is difficult to apply new calibration information should the original calibration be found inaccurate.

Exceptions to the above occur when it is possible to significantly compress the data by some amount of preprocessing such as digital filtering and decimation, using n-point averages, signal averaging, etc.

Time Coherency and Synchronization

It is generally important to acquire data in a time-coherent manner especially where one is expecting to extract time-dependent information from the data—which is typically the case. In such applications it is frequently important to be able to relate data from different sensors in time as well as having a uniform sample interval for each sensor. Depending on the degree of synchronization required this can be problematical with some configurations, especially when the system is physically distributed.

The *Grand Interconnect* provides a number of features that insure that data is captured synchronous with a precision internal crystal clock or an external clock. Furthermore, data samples in the data stream are organized in a predetermined fixed relationship with time, i.e. the sample time can be reliably correlated with the relative position of the sample in the data stream.

When the I/O chassis are co-located, it is possible to share a common clock between chassis. The VXI Slot-0 *Grand Interconnect* Controller V160 provides the capability of sharing a common 10 MHz system clock as well as one of the 8 VXI TTL trigger lines across multiple chassis using a coax cable.

The interconnect also supports a “broadcast” *trigger message* that can be used to synchronize clocks in different I/O chassis. Each *slave* node includes a presettable delay counter that “delays” a synchronization pulse generated from a broadcast trigger message. By presetting the delay counters in each slave node by an appropriate delay factor, it is possible to compensate for the fixed delays encountered by the trigger message to each node.

6.3.4 Alternative Architectures

Before examining the architecture of the *Grand Interconnect* in detail, it is worth examining some of the commercial alternatives including FDDI and Ethernet. Both of these architectures are generally available and widely used. Both work very well for general purpose peer-to-peer processor links. They can be successfully used in data acquisition applications with suitably low performance requirements, and modest time coherency requirements between nodes. In general, they have good throughput, poor latency, and always require a relatively high performance processor to handle network protocol. Latency, the time from when one node attempts to initiate a transfer of information until the target accepts the transfer of the first byte, is a direct consequence of the peer-to-peer architecture of these networks. Furthermore, while "average latencies" of such networks may be quite acceptable for some cases, *worst case latencies can be substantial*—even in relatively controlled environments.

In many high performance data acquisition applications, these latencies may not be acceptable and may result in some level of difficulty in correlating data from different nodes. The *Grand Interconnect* was developed to overcome these limitations as well as provide a lower cost solution for high performance data acquisition.

6.3.5 Grand Interconnect Architecture

The *Grand Interconnect* architecture is illustrated in Figure 6.1. It includes a single host computer interface which initiates *all* transactions except for Asynchronous Event Notification transactions which can be generated by any slave node and inserted in the data stream between messages. The host interface can initiate one of four general messages:

Read Request This message is generated by the host to read one or more items from the addressed slave I/O chassis.

Write Request This message is generated by the host to write one or more items to the addressed slave I/O chassis.

Control This message is generated by the host to effect a control operation at the addressed slave I/O chassis

Trigger This message is generated by the host to generate an asynchronous event (trigger) at the addressed I/O chassis. If the message is addressed to chassis "0", the trigger is accepted by *all* I/O chassis.

Each message type includes a 7-bit "chassis address" which identifies the chassis to which the message is being sent. Data transfer messages include additional fields including *VXI address modifier*, *32-bit VXI address*, and for block operations a *32-bit transfer count.* Write requests include the write data. Read requests cause the addressed chassis to generate a **Read Reply** with the data that was read. Trigger requests include a data word that identify which trigger is to be generated at the addressed I/O chassis.

Other message types include:

Asynchronous Event This message is generated by a *slave* node to inform the host of an asynchronous event, e.g. selected interrupt or trigger line event.

Read Reply This message is generated by a *slave* node in response to a read request and identifies the start of data. Following the data is read status.

Write Reply This message is generated by a *slave* node in response to a write request. Its primary function is to pass write status back to the host.

Control Reply This message is generated by a *slave* node in response to a control request and includes control status information.

Trigger Reply This message is generated by a *slave* node in response to a trigger message (except in response to a broadcast trigger) and includes status.

Suspend Transfer This message is generated by a host node when a slave is sending data faster than it can process it, or by a slave node when the host is sending data faster than it can process it.

Continue Transfer This message is generated by the node which issued the *suspend* to resume data transmission.

In addition, each node includes FIFO buffering of the input serial stream. The FIFO provides sufficient buffering of the incoming data stream to permit the throttling mechanism (*Suspend Transfer*) time to suspend the incoming stream and allow the receiving node time to catch up before a *Continue Transfer* is issued. The throttling mechanism uses the FIFO half-full flag transitions to control throttling. This method insures that the device that is limiting the transfer rate always has a queue of work to perform.

Error Detection

Messages on the interconnect include a transverse parity bit on each data byte as well as a longitudinal parity byte. In general, fiber optic links provide very low error rates, and experience over many years using similar techniques has indicated that error recovery is in general not a major issue.

Synchronization Features

Synchronous deterministic capture of data is an important consideration in a data acquisition system. In any application where it is important to extract the time history of one or more signals requires that the relative sampling time of the signal(s) be known. It generally is not required to time-stamp at high resolution exactly when each sample is taken, but rather to sample at one or more known fixed frequencies and to use the relative position of the data in the data stream to infer the sample time. In such a scenario the data acquisition process must be driven by a stable, known frequency source with minimum jitter—typically a crystal clock.

The *Grand Interconnect* provides several mechanisms to support synchronous data acquisition. First, the V160 Slot-0 controller includes a precision 10 MHz clock to drive the VXI 10 MHz clock line as well as logic to derive sub-multiples of this clock to drive the list processor and trigger lines. By driving the ADCs and other I/O elements in the slave I/O chassis from these clocks and synchronously triggering the list processor it is possible to acquire and buffer data at multiple rates within the slave I/O chassis in a highly deterministic manner with very low jitter in the sampling interval. For co-located chassis the V160 Slot-0 controller provides a mechanism to share a common 10 MHz clock and one trigger line across multiple chassis to insure synchronous data acquisition across multiple chassis.

For highly distributed applications with less stringent jitter requirements, the *Grand Interconnect* provides a list processor at the host interface that can be triggered either from an internal programmable clock or an external clock. When triggered the host list processor can generate a sequence of interconnect commands that initiate data capture in various slave I/O chassis and initiate transfer to the host.

An associated feature that can be used to better synchronize I/O chassis activity in a distributed configuration is the *broadcast trigger message*. Each V160 includes a programmable delay that is associated with a broadcast trigger. This delay can be set up to compensate for fixed routing and propagation delays around the interconnect loop. Using this technique, it is possible to achieve somewhat better synchronization of data acquisition activities than otherwise.

For applications that require some form of time stamping of the data, e.g.

when data acquisition is triggered from an asynchronous external event, the Slot-0 provides an internal clock. This internal clock can be read as part of the data scan.

Chapter 7

VXI System Configurations and Applications

7.1 High Accuracy Thermocouple Application

7.1.1 Application Requirements

- Sampling rate 10 to 100 Hz per Channel.
- 240 channels of Thermocouple.
- Cold junction reference provided by customer.
- Continuous sampling.
- Measurement accuracy 3μVolt NIST traceable.
- Throughput 2.4 to 24 Ksamples/second.
- Periodic calibration against standards —must be automatic.

7.1.2 Design Considerations

- Use V208/V243 for high accuracy low-level analog signals.
- Ping-Pong buffer adequate for most applications.
- External NIST traceable DVM (typical 1μV accuracy).

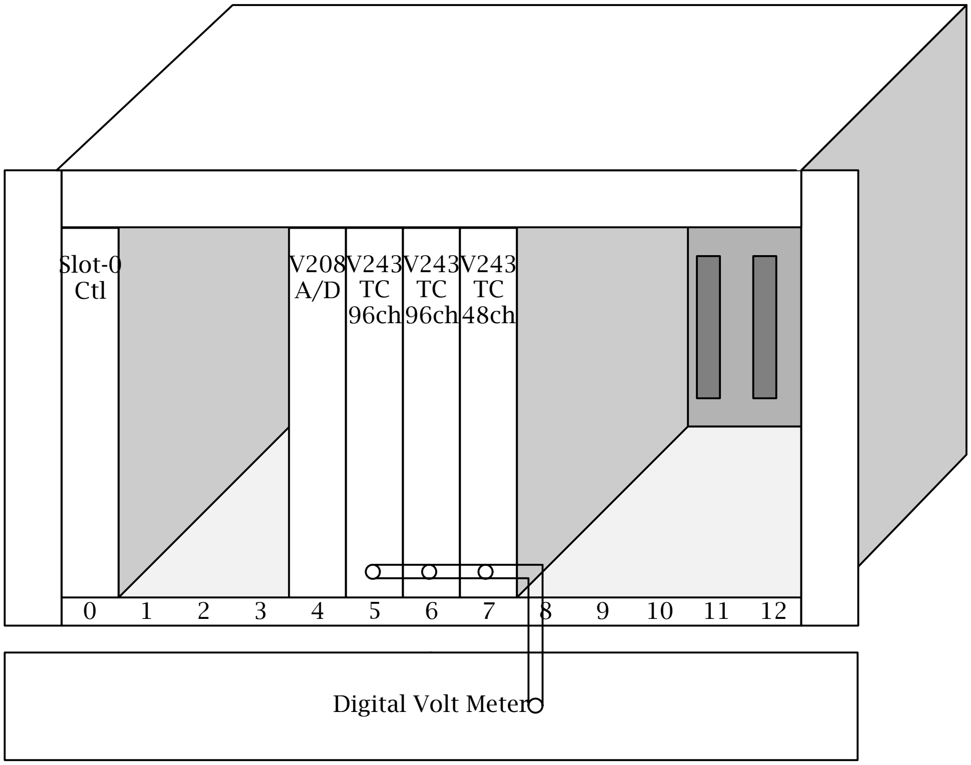


Figure 7.1: High Accuracy Thermocouple Application

- Front-panel bus cable between V243s and DVM for automated calibration.
- Use High Precision Calibration to achieve accuracy.
  - Offset and Gain *error correction terms* stored in EEPROM of each V243 and used to correct gain (m_j) and offset (b_j) terms.
  - Periodic calibration with NIST traceable DVM used to re-calculate *Gain error correction* and re-store in V243 EEPROM.

7.2 Signature Transient Capture

7.2.1 Application Requirements

- Monitor 32 analog channels.
- Sample rate 10 Ksamples/second/channel.
- Throughput 320 Ksamples/second.
- Capture criteria may involve complex relationship between different input channels over last 0.05 seconds (500 samples).
- When capture criteria is met:
  - Store 1 Ksamples pre-trigger (last 1/10 th second).
  - Store 10 Ksamples post-trigger (last 1 second).
  - Post Interrupt to Slot-0 to signal data ready.
- System must be capable of capturing back-to-back events.

7.2.2 Operation

- Data from each V207 data scan is deposited directly by hardware into 32Ksample circular buffer in V165 via DigiBus.
- DSP monitors data in circular buffer for “event” signature.
- When event signature is detected, 11 Ksamples/channel of data is moved to Dual Port RAM including 1K pre-event samples.
- When all 11 Ksamples have been moved to Dual Port RAM DSP asserts Interrupt Request to Slot-0.

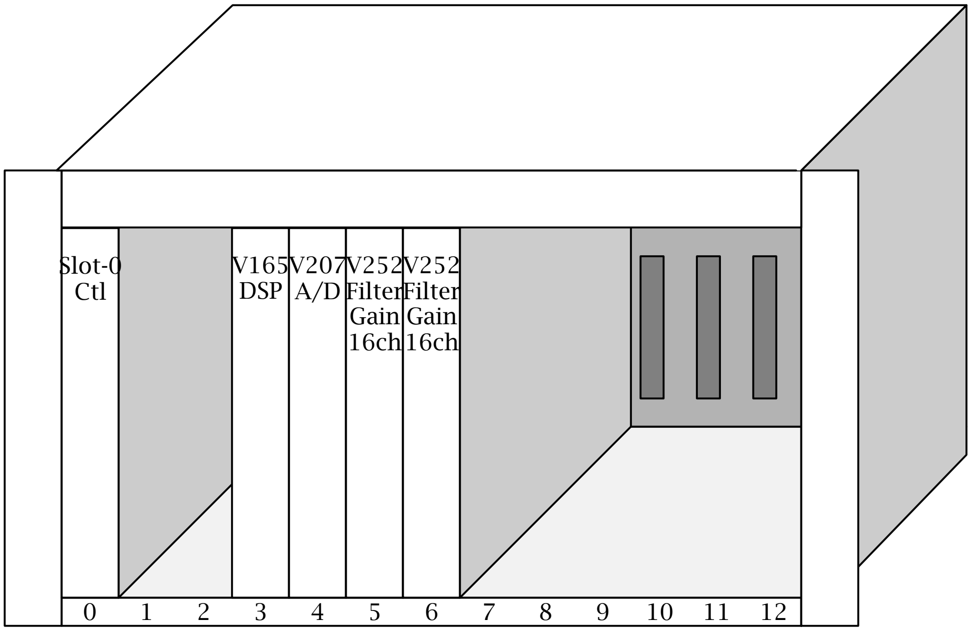


Figure 7.2: Transient Capture Application

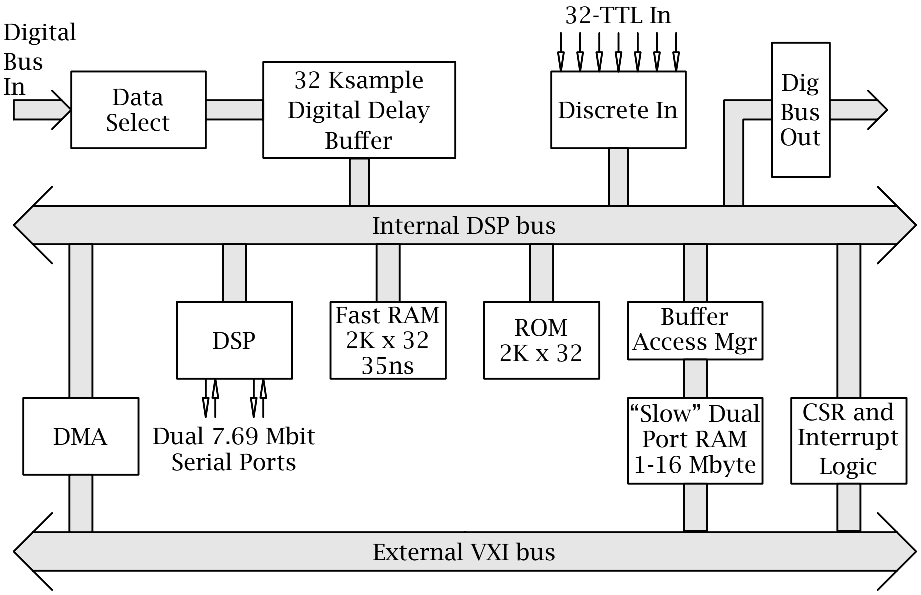


Figure 7.3: V165 DSP Block Diagram

7.2.3 "Typical" Applications

- Lightning Strokes on High Voltage Power Transmission Lines.
- Loose Parts Monitoring in Power Plants.
- Valve monitoring in e.g. Nuclear Power Plants.

7.3 High Performance Closed Loop Control

7.3.1 Application Requirements

- Monitor 32 analog channels data to be accessible by host computer for monitoring and display.
- Sample rate 10 Ksamples/second/channel.
- Throughput 320 Ksamples/second.

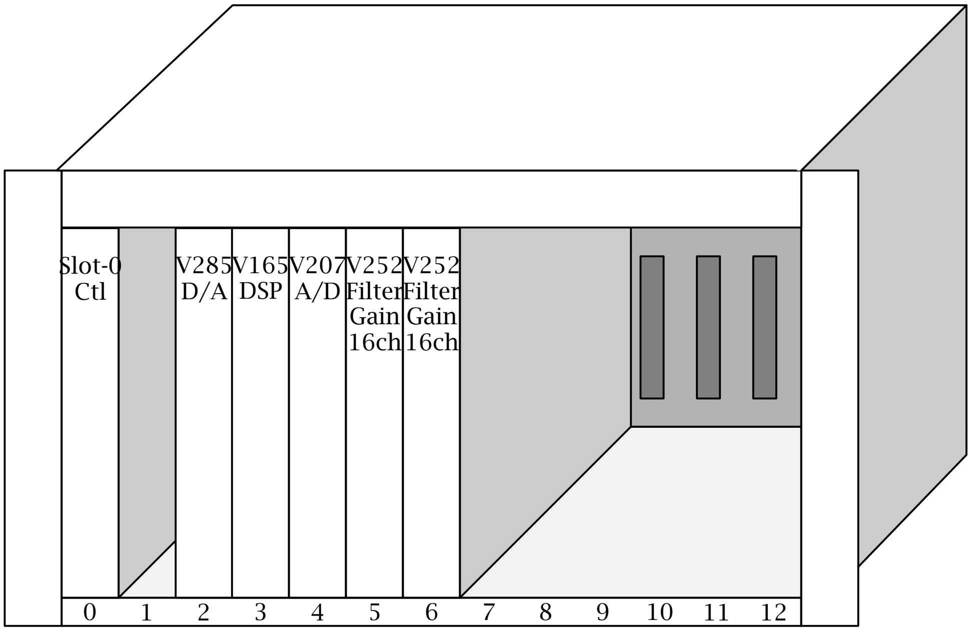


Figure 7.4: High Performance Closed Loop System Configuration

- Perform 4 control loops based on 8 of the input analog channels and control parameters updated from host.
- Generate 4 independent analog output channels for control.
- Inputs and outputs to be synchronous with 10 KHz clock.
- Analog outputs must be “glitch-free”.
- Control signals to be ramped to a “safe state” when any of 4 digital inputs are “true” or when an unsafe state is detected based on input signals.

7.3.2 Operation

- Eight channels of selected data from each V207 data scan is deposited directly by hardware into 32Ksample circular buffer in V165 via DigiBus.

- DSP picks analog inputs from circular buffer and current control parameters from dual-access RAM, computes control algorithm and sends updated analog output values to V285 D/A.

- DSP also checks operating conditions for safe state as well as digital inputs. If unsafe state is detected the DSP generates a sequence of analog output values to bring unit under test to a "safe state."

- Host computer can access any of the analog input channel data direct from the Ping Pong memory on the V207.

Appendix A

Front-end Analog Calibration

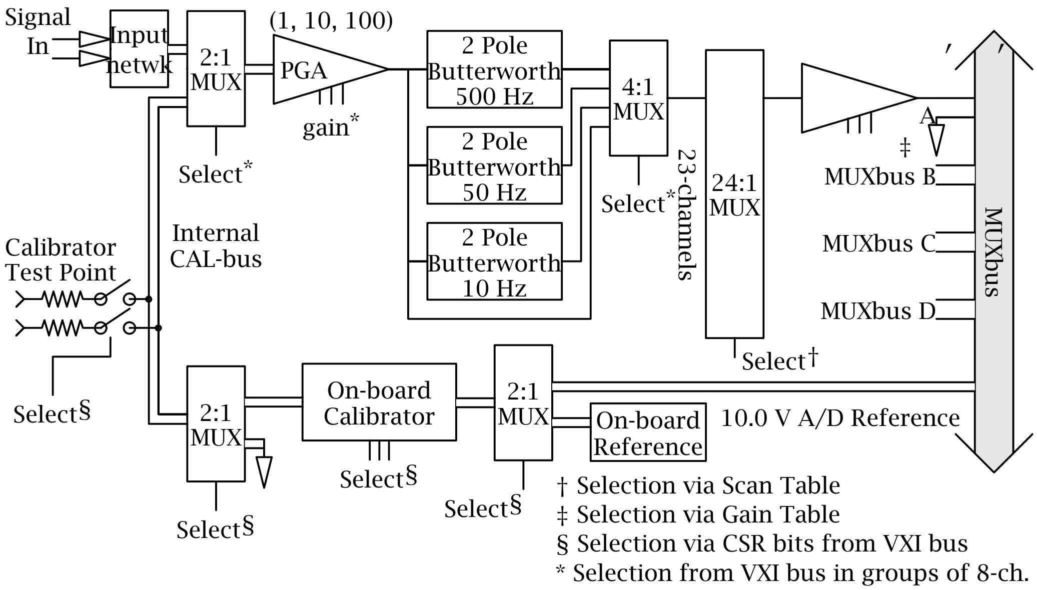
The KineticSystems *second generation* family of modules are designed for high performance, high accuracy applications. This family share a number of common *family* design philosophies including calibration.

In the design of analog hardware, two design approaches are possible. One is to select precision analog components and trim such parameters as gain and offset to acceptable tolerances during manufacture. This technique results in a module that requires considerable effort to calibrate and necessitates component tradeoffs between precision and long term stability.

The alternative approach, which is used in the KSC Silver Bullet line, is to choose components based foremost on stability and performance and to achieve precision by performing a calibration prior to data acquisition. Some of the advantages of this approach include better price/performance due to the need for fewer precision components and the end-to-end checkout of analog and digital paths prior to collecting data.

A.1 Calibration Technique

A standard calibration philosophy is used across the KSC VXI-family of analog input modules. Each signal conditioning module is capable of switching the front-end analog circuitry between the input signal that is to be measured, ground and the output of a hybrid calibrator. The reference for the hybrid calibrator can be either an on-board reference or a reference voltage provided by the V207 or V208 ADC. This approach provides an end-to-end calibration of the analog sections and at the same time provides verification that the entire data acquisition system is functional. Refer to Figure A.1.



† Selection via Scan Table
‡ Selection via Gain Table
§ Selection via CSR bits from VXI bus
* Selection from VXI bus in groups of 8-ch.

Figure A.1: Typical analog front-end illustrating calibration paths

A.1.1 Precision Reference and NIST Traceability

The precision reference for the signal conditioning modules and ADCs is adjusted at the factory to +10.0 Volts using an NIST traceable Digital Voltmeter (DVM). The precision reference is a Burr-Brown REF102CM with a temperature coefficient of 2.5ppm/°C, and 5ppm/1000hr drift.

For applications where accuracy and NIST traceability are important KineticSystems recommends that a *periodic system calibration* be performed at approximately 6 month intervals using a NIST traceable DVM to calibrate the precision reference. In general users should choose a periodic calibration interval that is appropriate for their situation. Test points are provided on the front panel of modules which contain precision references. The user can adjust the precision reference output to 10.0 Volts using a recessed screwdriver adjustment and the NIST traceable DVM.

A.1.2 On-board Calibrator

A custom active hybrid calibrator is used at each signal conditioning module to provide a precision programmable voltage divider for the reference voltage.

The output of the calibrator is capable of generating output voltages from ±2mV to ±10V in a 1, 2, 5, 10 progression. The calibrator divides the voltage in two steps: a *decade divider* with ranges of 1, 0.1, 0.01, and 0.001; and a *vernier divider* of 1.0, 0.5, and 0.2. Output accuracy is summarized for each stage in Table A.1.

Table A.1: Hybrid calibrator maximum error and temperature stability table.

Stage	Gain	Max Error (%)	Max ppm/°C
Polarity	±1.0	±0.01%	±6 ppm/°C Max
Decade	1.0	0%	0
	0.1	±0.007%	0
	0.01	±0.03%	±10 ppm/°C Max
	0.001	±0.1%	±50 ppm/°C Max
Vernier	1.0	±0%	0 ppm/°C Max
	0.5	±0.005%	±2 ppm/°C Max
	0.2	±0.007%	±2 ppm/°C Max

For example the **maximum** error for a 5mV calibration output from a 10.0 volt reference is the sum of the polarity, decade and vernier errors

$$0.01 + 0.1 + 0.005 = 0.115\%$$

or $\pm 5.75\mu$Volts.

Temperature stability of the hybrid calibrator is also given in Table A.1. For example the worst case temperature stability of the hybrid calibrator on the 5mV scale is

$$6 + 2 + 50 = \pm 57\text{ppm}/^{\circ}\text{C Max}$$

Long term drift of the hybrid calibrator is ±50 ppm/1000 hours.

A.2 Standard Calibration Procedure

For optimum accuracy it is recommended that the user perform a calibration of the channels to be used prior to acquiring data. The calibration procedure should be performed following a minimum equipment warm-up of approximately 1/2 hour. Each channel or group of channels which share common active input circuitry should be calibrated following warm-up and prior to acquiring data.

It is recommended that the following calibration procedure be followed. The calibration should be performed with no more than 8 independent channels per hybrid calibrator to minimize loading of the calibrator and that no more than 8 hybrid calibrators be switched to a single reference source at one time (a consideration when the ADC reference source is used). The calibration measurement should be made with the calibrator range (k) set per note 3 below for each channel (j).

1. Acquire N-samples with grounded inputs selected $\ldots X_i^j(0)_{i=1,N}$.

2. Acquire N-samples with +CAL inputs selected $\ldots X_i^{jk}(\text{+CAL})_{i=1,N}$.

3. Acquire N-samples with -CAL inputs selected $\ldots X_i^{jk}(\text{-CAL})_{i=1,N}$.

Notes:

1. Be sure to allow adequate settling time between switching input levels *especially if lowpass filters are present* in the analog path. The time for a simple RC filter to settle to one lsb of a 16-bit ADC is 10.38 time constants or $1.66/f_c$ where f_c is the filter cutoff frequency. Filters with sharper roll-off typically require proportionally longer settling times.

2. It is recommended that at least N=20 samples be acquired for each input level.

3. The calibrator voltage should be set to give an output voltage that is near full scale on the ADC. For example with a gain of 100 the input range of the ADC is ±102.4mV so the calibrator should be set to the ±100mV range.

From this calibration data a slope (m_j) and an offset (b_j) are computed for each channel (j). These quantities are then used to convert the measured ADC counts for channel j to volts.

A.2.1 Determining the offset b_j and slope m_j

The offset b_j for channel j is computed from the N *calibration* measurements $X_i^j(0)$ with the input switched to ground as follows:

$$b_j = \frac{1}{N}\sum_{i=1}^{N} X_i^j(0)$$

The slope m_j^k for channel j is computed from the N measurements X_i^{jk}(+CAL) with calibrator range k (+CAL) selected and the N measurements X_i^{jk}(-CAL) with calibrator range k (-CAL) selected as follows:

$$m_j^k = \frac{\frac{1}{N}[\sum_{i=1}^{N} X_i^{jk}(\text{+CAL}) - \sum_{i=1}^{N} X_i^{jk}(\text{-CAL})]}{E_0^k(\text{+CAL}) - E_0^k(\text{-CAL})}$$

Where:

X_i^{jk} is the ith measured ADC counts for channel j using calibrator range k.

E_0^k is the published calibrator voltage value for range k.

A.2.2 Applying the calibration to data

The input voltage to be measured V_j is computed from the acquired ADC count X_j by the following relation:

$$\boxed{V_j = \frac{(X_j - b_j)}{m_j^k}.}$$

Where:

V_j is the *derived* analog input voltage for channel j in Volts.

X_j is the *measured* analog input for channel j in ADC counts

b_j is the offset for channel j as determined from the calibration run prior to taking data as outlined in Section A.2.1.

m_j^k is the gain for channel j as determined from the calibration run prior to taking data as outlined in Section A.2.1.

A.2.3 Periodic System Calibration

As discussed earlier KineticSystems recommends that the precision reference source used during pre-acquisition calibration be calibrated at approximately 6 month intervals or an interval that is appropriate to the users situation. The calibration is accomplished by attaching a NIST traceable DVM on the test points provided on the module front panel and adjusting the output of the precision reference to 10.0 Volts.

A.3 High Precision Calibration Procedure

For some low-level applications, for example thermocouples, accurate voltage measurements at millivolt signal levels may be required. Some Silver Bullet signal conditioning modules such as the V243 implement a further level of calibration to achieve accuracies of a few microvolts. These accuracies are achieved by calibrating the internal calibrator and channel-to-channel thermal EMF differences between the actual signal input connector and the common switched ground used for calibration. These second-order correction factors are stored in EEPROM for **each** module when the module is calibrated at the factory. These correction factors can be read by the application software and applied at the time the data is read.

Two second-order corrections are involved in the precision calibration:

1. A gain correction term ϵ_k that represents a second-order correction to the gain based on the hybrid calibrator range k. Note that the gain correction is most significant at high gains since the hybrid calibrator accuracy is most accurate at calibration levels of 100mV and above when compared to 16-bit ADC measurement accuracy of 0.003% . Refer to Table A.1.

2. An offset correction δ_j that represents a second-order correction to the offset for channel j which represents slight thermal EMF differences due to different signal paths from the front connector to the input amplifier and the common switched input ground used in during calibration to determine channel offsets. Note that the offset correction is only important at high gains when measuring low-level signals where μVolt offset differences can be resolved by the ADC.

When using the second-order correction terms, the measured input voltage is given by the expression:

$$\boxed{V_j = \frac{(X_j - b'_j)}{m_j^k}.}$$

Where:

V_j is the *derived* analog input voltage for channel j in Volts.

X_j is the *measured* analog input for channel j in ADC counts

b'_j is the offset for channel j as determined from the calibration run prior to taking data as outlined in Section A.3.2 that includes a second order offset correction term.

m_j^k is the gain for channel j using calibrator range k as determined from the calibration run prior to taking data as outlined in Section A.3.2.

A.3.1 Pre-acquisition Calibration

As noted earlier the equipment should be turned on and temperatures allowed to stabilize for a minimum of 1/2 hour prior to performing the calibration and data collection. It is also recommended that temperature fluctuations be held to a minimum during the warmup, calibration and data acquisition.

The second order calibration is based on gain correction terms which not only correct for the precision of the hybrid calibrator, but also corrects for any deviation of the precision reference from 10.0 Volts. For these reasons the following considerations apply:

- Only the internal precision reference source in the module should be used with this calibration technique (not the common ADC reference source).
- The periodic calibration procedure no longer *requires* calibrating the precision reference source since the second order correction terms correct for any long term drift.
- A switched set of test points for measuring the calibrator output permit a common NIST traceable DVM to be used for a completely automatic periodic calibration.

A.3.2 Determining the offset b_j and slope m_j^k

Prior to taking data a calibration run should be performed to determine the offset b'_j and gain m_j^k for each data channel. These quantities are derived as follows:

$$b'_j = \frac{1}{N}\sum_{i=1}^{N} X_i^j(0) + m_j^k \cdot \delta_j \cdot 10^{-9}$$

> Note: Since the offset correction δ_j is small the published gain can be used in place of m_j^k in the above expression for the offset b'_j with out loss of accuracy.

$$m_j^k = \frac{\frac{1}{N}[\sum_{i=1}^{N} X_i^{jk}(\text{+CAL}) - \sum_{i=1}^{N} X_i^{jk}(\text{-CAL})]}{(E_0^k(\text{+CAL}) - E_0^k(\text{-CAL}))(1 + \epsilon_k \times 10^{-6})}$$

where:

$X_i^j(0)$ is the ith measured ADC counts for channel j with inputs switched to internal ground.

$X_i^{jk}(\pm\text{CAL})$ is the ith measured ADC counts for channel j using calibrator range k.

E_0^k is the published calibrator voltage value for range k.

ϵ_k is the 16-bit signed 2s complement integer “calibrator correction factor” stored in EEPROM for CAL range k scaled by 10^6 (e.g. ±2mV, ±5mV, ±10mV, ...) ranges.

δ_j is the 16-bit signed 2s complement integer “offset correction factor” stored in EEPROM for channel j scaled by 10^9.

A.3.3 Periodic Calibration Procedure

The periodic high precision calibration differs from the general periodic calibration. It is no longer necessary to adjust the precision reference source voltage as any drift in the precision reference is reflected in the second-order gain corrections that are derived and stored in EEPROM during the periodic calibration.

It is recommended that the periodic *gain calibration* be performed at approximately 6 month intervals. It is expected that the *offset periodic calibration* need only be performed once over the life of the module. This calibration is performed at the factory during final testing.

When the system is configured with a common bused cable between the calibrator test points and a NIST traceable DVM which is computer controlled, the periodic gain calibration can be performed automatically without operator intervention.

Second-order Periodic Gain Calibration

The calibrator or gain correction terms ϵ_k for the high precision calibration procedure are computed and stored in EEPROM based on factory measurements using a NIST traceable DVM. The recommended interval for this calibration is approximately 6 month intervals for applications requiring a high level of precision. The user should determine the calibration interval based on the individual situation.

The calibration procedure is to connect a NIST traceable DVM of suitable accuracy to the calibrator output which is available on the front panel of the module. The module is then set up through software to select the desired calibrator output range k and +CAL output using the precision reference source internal to the module being calibrated and selecting the calibrator output to the front panel test points. The DVM reading D^k(+CAL) is recorded. A similar measurement D^k(-CAL) is made. The value of ϵ_k is then computed for each calibrator range k and stored as a signed 16-bit integer in EEPROM as follows:

$$\epsilon_k = \frac{[D^k(\text{+CAL}) - E_0^k(\text{+CAL})] - [D^k(\text{-CAL}) - E_0^k(\text{-CAL})]}{[E_0^k(\text{+CAL}) - E_0^k(\text{-CAL})]} \times 10^6$$

Where:

D^k is the calibrator output voltage measured with an NIST traceable DVM for calibrator range k.

E_0^k is the published calibrator voltage value for range k (e.g. 5mV, 10mV, 20mV, ...).

Note:

> The value of ϵ_k before scaling is a small number ($\ll$1). For the *ideal* calibrator $\epsilon_k = 0$. To store the value of ϵ_k in a computer architecture independent representation we have chosen to store it as a scaled 16-bit 2s complement binary integer which has a range of ±32767.

This gives the correction multiplier $(1 + \epsilon_k)$ a range from 0.967233 to 1.032767 to an accuracy of $1 : 10^6$ (0.0001%).

Offset Correction

The offset calibration is expected to be considerably less sensitive to long term drift and is expected to be stable over the life of the module. This calibration is performed at the factory and the results stored in EEPROM. The calibration can also be performed in the field. For this calibration the input wiring *must* be disconnected and a special shorting connector installed in its place.

The offset correction terms δ_j are computed and stored in EEPROM based on measurements of the differences in offsets for each channel between the offset measured with the inputs shorted using a special connector and when the common internal ground is selected that is used during a pre-acquisition calibration (refer to Figure A.1). These differences are scaled to nanovolts and stored as 2s complement binary integers in EEPROM.

$$\delta_j = (\sum_{i=1}^{N} X_i^j(\text{Input-gnd}) - \sum_{i=1}^{N} X_i^j(\text{Internal-gnd}))\frac{1}{Nm_j} \times 10^9$$

where:

X_i^j(Input-gnd) is the ith measured ADC counts for channel j with the inputs grounded.

X_i^j(Internal-gnd) is the ith measured ADC counts for channel j with inputs switched to internal ground.

m_j is the channel gain. This can be the gain determined from a simple calibration or just the published channel gain since the error introduced by the latter is minimal.

A.4 Error Budget Analysis V243/V208

The following is an error budget analysis of various error sources for the V243/V208 on the 5 millivolt scale (gain=2000) and a temperature range of $\pm 4^\circ C$. It is assumed in this analysis that the high precision calibration method is used. In all cases *typical* (1σ) error estimates are used in this analysis. Note that these may differ from the respective data sheets where *worst case* errors are given.

The following error budget analysis is provided as a **preliminary engineering estimate.**

A.4.1 Systematic Errors

	Typical (1σ)	Accuracy
	Error	±4°C
Precision Reference:		
Temperature Stability	±2.5ppm/°C	±0.05μV
Long Term Drift	±5ppm/°C	±0.025μV
Calibration Accuracy	±5μV on 10V scale	±0.025μV
Calibrator:		
Polarity	±2.4ppm/°C	±0.048μV
Decade (0.001)	±20ppm/°C	±0.40μV
Vernier (0.5)	±0.8ppm/°C	±0.016μV
Instrumentation Front-end:		
Offset	±0.1 + 0.5/Gain (μV/°C)	±0.42μV
Gain Stability	±10ppm/°C	±0.2μV
Total Systematic Error(see note):		±0.619μV

Note: The total systematic error is determined by adding the square root of the sum of the squares of the individual errors.

A.4.2 Measurement Errors

		Typical (1σ) Error	Accuracy ±4$^{\circ}C$
ADC:			
	Diff Non-linearity	0.001%	±0.05μV
	Integral non-linearity	0.0015%	±0.075μV
	Crosstalk	-90dB	±0.16μV
V243 Front-end:			
	Integral non-linearity	0.001%	±0.05μV
	Noise RTI	±0.3μV RMS	±0.3μV
	Cross Talk	-90dB	±0.16μV
	Gain Accuracy after cal		±1.0μV
V243 Total Measurement Error (see note)			±1.037μV

Note: The total measurement error is determined by adding the square root of the sum of the squares of the individual errors.

Appendix B

Connecting Analog Signals and Grounding

Most data acquisition systems involve obtaining data from various transducers that produce analog signals. Often, the signals from the transducers are low level and require various kinds of signal conditioning. Also, the transducers are frequently located some distance from the data acquisition front-end.

This chapter explores some of the issues involved with field wiring, grounding, isolation, signal conditioning, noise, calibration and related *analog* issues.

B.1 Proper Connections to the Sensors

B.1.1 Instrumentation Front-end

–what it does for you and why and what to look for

Measuring Data and Not Noise

B.1.2 Differential vs. Single-ended Input Channels

Two very important concepts that affect the performance of a data acquisition system are *single-ended* and *differential.* A single-ended input channel, as

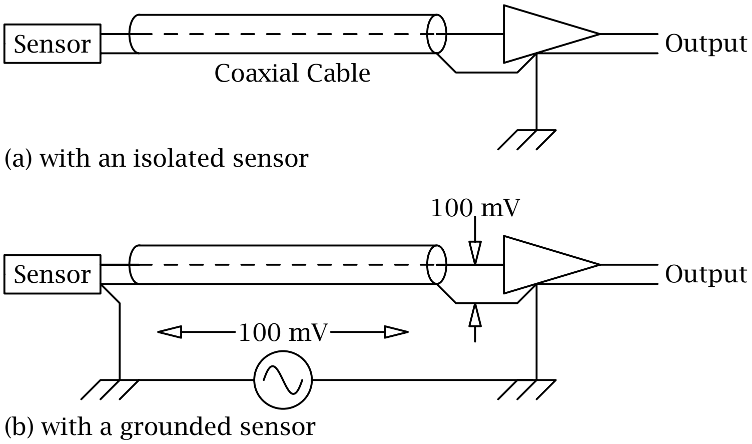


Figure B.1: A single-ended data acquisition input channel

shown in Figure B.1(a), completes the circuit from a sensor to the data acquisition (DAQ) system input circuit via a *signal* wire and a *return* wire. The *return* wire is usually the cable shield and is generally connected to the DAQ system's circuit common—which is generally connected to "ground." In an ideal world, this should not present a problem. *Unfortunately, the world is filled with many noise sources that can interfere with data.* The problem with single-ended input circuits is that the cable shield is part of the signal path—*and any noise voltage developed across the shield adds full-force to the signal!*

Figure B.1(b) shows a sensor that is connected to "ground" and is wired to a grounded single-ended input. If there is 100 millivolt ac potential difference between the sensor's "ground" and the DAQ system's "ground," then *all of the 100 millivolt noise* will be superimposed on the signal. If the sensor is a thermocouple, the noise is likely larger than the dc signal voltage. Even if the sensor has no connections to ground, the data acquisition system must be carefully designed so all connection points are very close to the same potential, or errors will be introduced to the received signal of the various channels. Also, if the input wiring is close to sources of electrical noise, interference may be coupled into the signal path.

A differential-input channel—often called a *balanced input*—is generally connected to a sensor as shown in Figure B.2(a). A *shielded twisted pair* cable is most often used for differential operation. The signal is received by an *instrumentation amplifier.* The primary characteristic of an instrumentation amplifier is that it delivers a signal at its output that is proportional to the *difference* between the voltage on its "+" input and its "-" input.

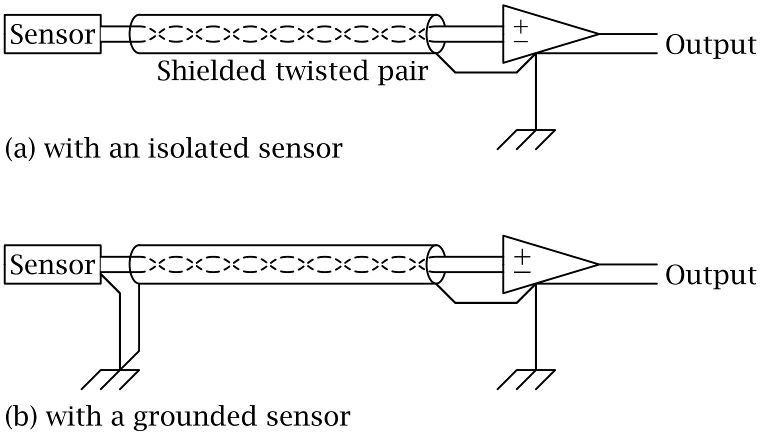


Figure B.2: A differential data acquisition input channel

A real-world example of a differential-input channel is shown in Figure B.2(b). Note that, in this case, the signal circuit is completed *without any signal passing through the shield*. If a noise voltage is impressed across the shield because of a ground potential difference, the effect on the signal will be greatly attenuated. The characteristic of the cable can cause interference to creep into a differential system with long cable runs. The shielded cable must contain a *twisted pair*. Some single-pair cables have both conductors in the same plane instead of having the pair of wires twisted. If this cable is near a strong source of electrical noise, one signal wire will be nearer to the noise than the other, and the noise will not cancel as well. Also, some two-pair cables with separate shields do not contain pairs that are twisted. Systems using such cables can exhibit unacceptable coupling—called crosstalk—between the channels, even though each pair is shielded.

B.1.3 Common Mode and Normal Mode

Common mode rejection ratio—often abbreviated CMRR—describes the effect that unwanted noise between the signal conductors and ground has on the desired input signal. It is called *common* mode because the unwanted signal (often caused by power line noise) is impressed across both conductors of a

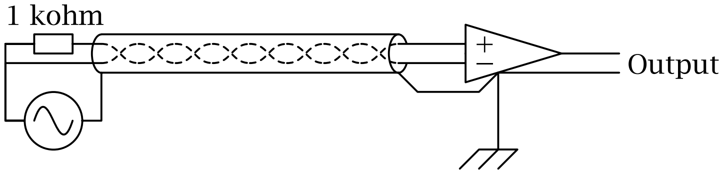


Figure B.3: Measuring common mode rejection ratio

differential pair and—in the ideal case—is canceled out by the balanced system. CMRR is generally measured as shown in Figure B.3. The test voltage is impressed across both conductors of the differential input. Often this test uses a 1000-ohm resistor in one leg to represent the fact that the "real" transducer source may not be perfectly balanced. CMRR is usually expressed in decibels (dB). In this case the dB value represents a logarithmic voltage ratio between the output signal caused by a common-mode voltage and that from a normal-mode voltage. A normal-mode voltage is that impressed *across* the input conductors. For single-ended systems with the shield conductor connected to ground at the instrumentation front-end, the ratio between common mode and normal mode is 1:1—*no common-mode rejection*.

Each 20 dB represents an increase in the common mode rejection ratio by a factor of 10. Therefore a CMRR of 80 dB represents an attenuation of common-mode noise equal to 10,000 to 1. Another important parameter here is the maximum linear input swing on the input circuit. This value is often ±10 volts. With a CMRR of 80 dB, a 10 volt RMS (Root Mean Square) common-mode signal should have an effect on the input signal of 2 millivolts (20/10,000). However, if this is a sine wave, the voltage will reach positive and negative peaks of about 14 volts. This will likely be outside the linear range of the instrumentation amplifier and will result in substantial feed-through during portions of each cycle.

As just discussed, the value of CMRR determines how much *common-mode* noise gets converted to *normal-mode* signal. Some amount of noise may also *enter the system as normal-mode signal*, caused by noise pickup at the transducer, etc. For whatever cause, once an unwanted signal becomes a normal-mode voltage, it can be eliminated only by filtering. This is usually accomplished by low-pass filtering. For slowly changing signals, such as thermocouples, this is often accomplished by one- or two-pole passive (resistor-capacitor) filters connected directly to the input pair before any electronics. The cutoff frequency for these filters is often in the range of 2 to 10 Hertz to provide good normal-mode attenuation to power line frequency and its harmonics.

For fast-changing signals, any attempt to attenuate 50 or 60 Hz power line frequencies would prevent these changes from being monitored by the data acquisition system. In this case, sufficient precautions—such as proper grounding and good CMRR—must be taken to prevent noise from becoming normal-mode in the first place.

B.1.4 Isolation

As indicated earlier, most instrumentation front-ends require that each conductor of a differential input—and the signal conductor of a single-ended input—remain within about ±10 volts of the input common ground. This is not satisfactory for input signals with large common-mode voltage present. An example of this is the measurement from a current shunt that is several hundred volts above ground. An isolated input circuit is appropriate for such an application. This isolation can be built in to the data acquisition system or consist of an isolation block in front of the DAQ system. In either case, the most common practice is to use an isolation amplifier for this purpose. An isolation amplifier usually uses a dc-to-dc-converter-powered “floating” amplifier that produces a high-frequency signal whose duty cycle is proportional to the input voltage. This high-frequency signal is then coupled across an isolation barrier and filtered. The resulting signal is proportional to the input voltage and is coupled via the output amplifier. An isolated input circuit monitoring a “floating” shunt is shown in Figure B.4. The advantage of such an input circuit is that it can accommodate high common-mode voltages. Isolated input channels result in a significant increase in per-channel cost. Frequency response is usually limited to about 30 kilohertz. Isolated input circuits are usually specified only when a non-isolated approach cannot be used.

B.1.5 Grounding—Bad Ground Loops vs. Good Ground Loops

The subject of good grounding practices usually causes more arguments than any other aspects of high-performance data acquisition. Also, it is generally felt that ground loops should be avoided. The purpose of this section is to demystify the issue of grounding and show that there are bad *AND good* ground loops!

A good wiring practice for a voltage-input channel is shown in Figure B.5(a). For this case, the sensor is not grounded and the cable shield is connected to

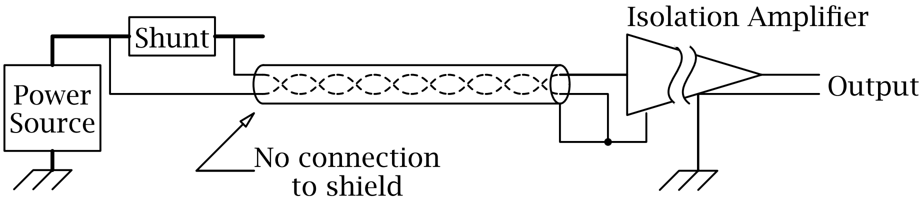


Figure B.4: An isolated data acquisition input circuit

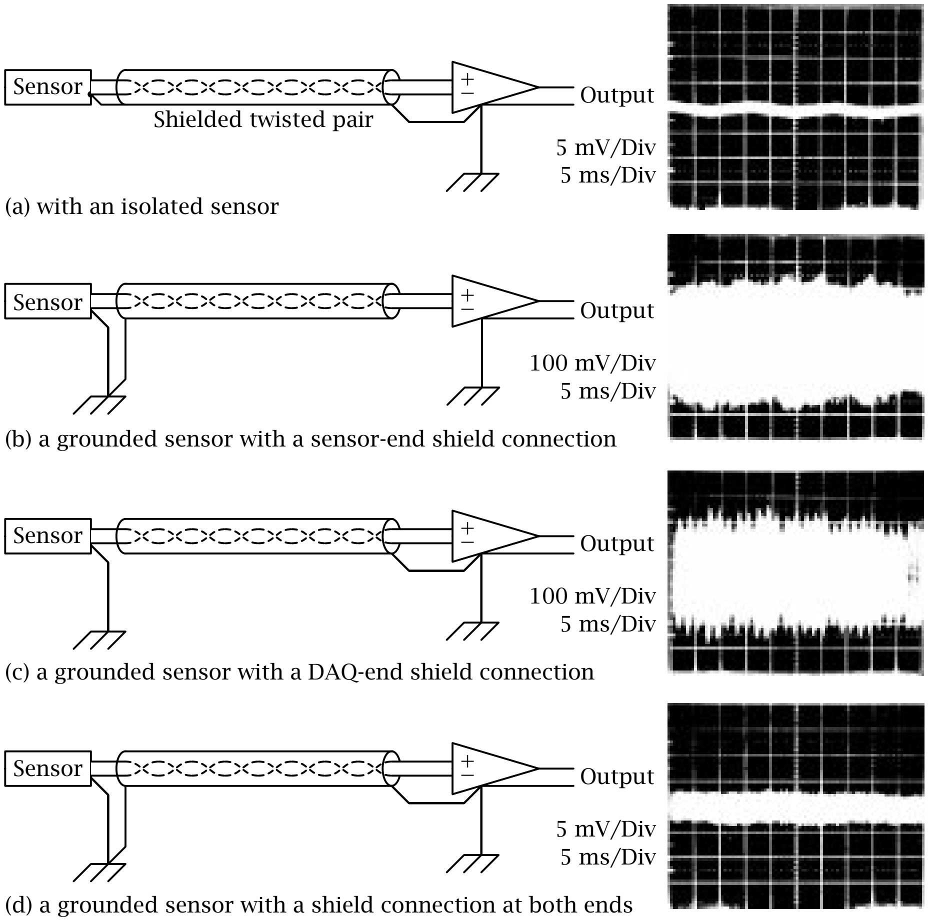
the midpoint of the sensor as well as to the "ground" connection at the data acquisition system. This shield connection also meets the requirement of most instrumentation front-ends.

There must be a return path to the instrumentation common so that the input current (as low as it is) will not cause the input pair to "float" outside the common-mode range. Failure to have this return path is a common cause of DAQ system problems.

This will generally cause data to be collected with very poor linearity.

The situation is more complex if the sensor is grounded. The customary recommendation in this case is to connect the shield at the sensor and not at the DAQ system input *to avoid a ground loop*, as is shown in Figure B.5(b). Another possible method to avoid a ground loop is to leave the shield unconnected at the sensor, as is shown in Figure B.5(c). For most applications, the circuit shown in Figure B.5(d) is recommended, with the shield connected at *both ends*, even though this creates a ground loop. This is the first example of a *good* ground loop. In nearly all cases, this double grounding has been shown to give far superior performance than any configuration with the shield "open" at either end.

The photographs in Figure B.5 show the results of a test with the sensor ground and the instrumentation front-end ground derived from separate outside ac power drops to increase the noise voltage between these "grounds." A 10-ohm resistor was used to simulate the sensor. The measurements were taken from the output of an instrumentation amplifier with unity gain. For the ungrounded sensor, as shown in Figure B.5(a), very low noise is present at the instrumentation amplifier output. Figure B.5(b) shows the result when



(a) with an isolated sensor

(b) a grounded sensor with a sensor-end shield connection

(c) a grounded sensor with a DAQ-end shield connection

(d) a grounded sensor with a shield connection at both ends

Figure B.5: Various shield grounding methods

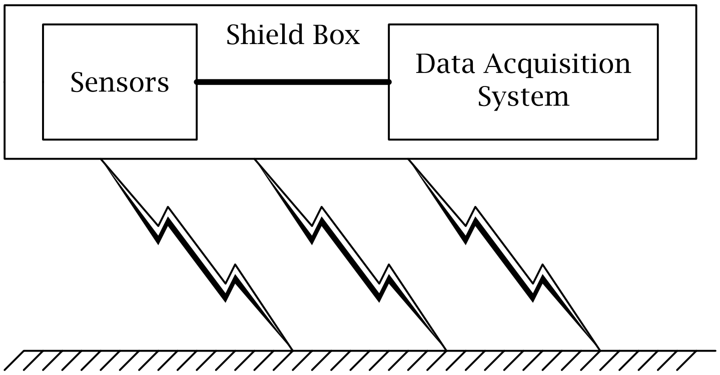


Figure B.6: A data acquisition system at 1 million volts above ground

the shield is connected only at the sensor end. The noise level reached 500 millivolts. Similarly, Figure B.5(c) shows a peak-to-peak noise level of 550 millivolts with the shield connected only at the amplifier end. Finally, when the shield was connected at *both ends*, creating a *good* ground loop, the peak-to-peak noise was reduced to 5 millivolts, as shown in Figure B.5(d). Grounding at *both* ends reduced the noise input by a factor of about *100 to 1*. This configuration contains no filtering. If a single-pole 5 kHz low-pass filter is added, the noise is less than 1 millivolt when both ends are grounded.

How can it be that a ground loop substantially improves performance? A ground loop is generally *bad* if it involves a signal-carrying conductor. An example of this was shown in Figure B.1(b), where the voltage produced by the ground current translated one-to-one into normal-mode voltage because the shield is a "return" for the signal. A ground loop is often *good* if it does not involve a signal-carrying conductor. The high level of noise was seen when the shield was connected at one end is primarily high-frequency “hash” that entered the system through reduced common-mode rejection and nonlinearities in the instrumentation amplifier at high frequencies. The cable capacitance and other factors greatly reduce the transmission of this noise when the shield is connected to the circuit elements at both ends. Indeed, with a connection to ground at both ends, current flows through the shield, particularly at power-line frequencies, and the signal conductors act as secondary windings of a transformer with the shield as the primary. The effect of this transformer action is greatly reduced because these voltages cancel out in the differential-input instrumentation amplifier. The double grounding cannot be applied if there is a substantial potential difference between the two circuit commons. For this case, an isolated input channel, as shown in Figure B.4, is appropriate. Generally speaking, if a "grounded" sensor and "grounded" instrumentation input produce a ground potential difference that prevents the grounding of the shield at both ends, input isolation must be provided.

Keeping the ground system as unipotential as possible is another very important aspect of reducing noise pickup in a data acquisition system. KineticSystems has supplied data acquisition chassis for research laboratory Van de Graff generators that are operating at 1 million volts above ground and are transmitting digital data to ground-connected hardware via fiber optics. Refer to Figure B.6. How do these systems operate without excessive noise pickup? Just as people don't notice that the earth's surface is spinning at speeds up to 1,000 miles per hour because all of their surroundings are moving at the same speed, a data front-end and its sensors that are at the same potential and well shielded from a noisy environment can perform quite well. Note that the important factor is that the sensors, the wiring and the instrumentation front-end are at nearly the same potential.

If the data system uses more than one equipment rack, these racks should be bonded to each other directly by screws or by a large conductive strap. The rack system should be connected to a good ground—often electrical conduit. If there are wire ducts or conduit carrying the signal cables, the usual recommendation is that these be bonded to the ground reference for the sensors *and* the racks that contain the data front-end. Note that this creates another ground loop, usually the *good* kind. This approach is controversial when the sensors are in a rather hostile electrical environment. The concern often expressed is that ground bonding at both ends will cause the electrical noise at the sensors to be transmitted to the data system ground and create more interference. Generally, the noise reduction resulting from the sensors and front-end being at nearly the same potential will far outweigh the introduction of noise by the ground loop. An additional benefit of ground bonding is that it will reduce the chance of damage to the electronics in the presence of lightning or other voltage transients. The lower the electrical impedance between the various parts of the circuit, the lower the potential difference in the event of a large voltage transient.

Even if the sensors are not connected to any part of an electrical circuit or ground, and the connection is as shown in Figure B.2(a), the preferred technique is to bond the grounds at the sensors and DAQ front-end, using the input cable conduit, wire tray, or a # 8 AWG or heavier wire. This will mini-

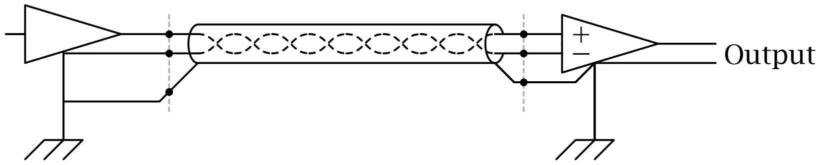


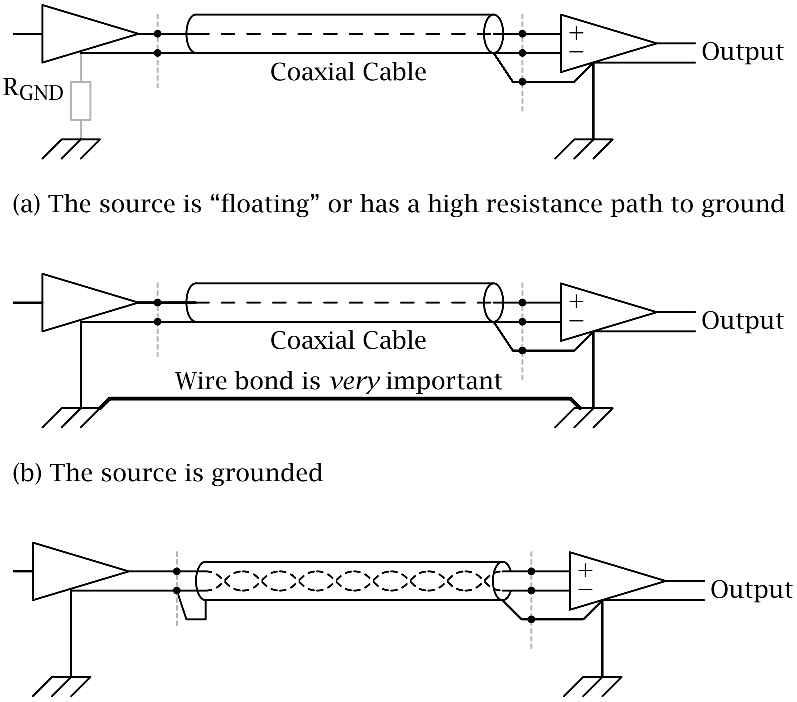
Figure B.7: Driving a differential input channel from an unbalanced source

mize the effect of any electrostatic coupling of noise to the sensors. Again, this is to keep the entire data acquisition front-end, including the sensors, in as unipotential an environment as possible. A similar approach involves the use of double-shielded cables, where the inner shield is connected as in Figure 4.2(a) and the outer shield is connected to ground at the sensors and to the front-end equipment chassis ground. Again, the guiding principle is to keep *all* parts of the analog subsystem moving at the same potential, just like associated objects are spinning together on the surface of the earth.

Other approaches can be used if the primary common-mode (signal-to-ground) interference is primarily high frequency in nature. One configuration uses a trifilar transformer, which is a tightly coupled three-winding transformer, with one winding in series with each of the two signal conductors and the third winding in series with the guard connection. This is quite effective in enhancing common-mode rejection at high frequencies. Another approach is to use a capacitor to connect the circuit ground to the shield at the instrumentation front-end. This provides a high-frequency ground while reducing the current caused by power-line frequencies. The effectiveness of the capacitor depends upon the particular situation. Also, some front-ends provide a guard signal for connection to the shield. The guard voltage is derived from a special instrumentation amplifier output that monitors the common-mode voltage and produces a signal to cancel it.

Another question that is often asked involves the correct cabling and ground connection when the signal source is an amplifier output instead of a passive sensor. If properly wired, this output circuit can be single-ended and produce a good signal-to-noise ratio. If the amplifier output circuit drives a two-contact-plus-shield-type connector, the connections are shown in Figure B.7. This provides a good balance because the output impedance of an instrumentation or operational amplifier is generally less than 1 ohm at low frequencies.

Some signal conditioning chassis contain BNC single-contact connectors at



(a) The source is “floating” or has a high resistance path to ground

(b) The source is grounded

(c) Using twisted pair cable

Figure B.8: Connecting drivers with coaxial outputs to differential inputs

their output and are intended to be used with coaxial cable. This can present problems in correctly wiring a coaxial cable to a differential-input front-end. If the output shield connection on the signal conditioning unit is isolated from ground or has a resistance path to ground of 1000 ohms or greater, then the connections shown in Figure B.8(a) should be used. If the shield conductor is grounded at the source, then the diagram shown in Figure B.8(b) can be followed to prevent a ground loop in a signal-carrying conductor. This approach may not be satisfactory unless the data acquisition system contains a high-frequency filter. Also, it is *very* desirable that the ground frames of the chassis associated with the signal transmitter and receiver are mounted in the same rack or nearby racks and are electrically bonded together so that the ground noise between them is minimized. A good wiring alternative, particularly if the two units are not in the same rack, is to convert the cable to a shielded-twisted-pair type *as close to the source as possible* as shown in Figure B.8(c).

B.2 Noise

The presence of noise in data can render that data meaningless. In some cases, data can be recovered through post processing. However, once noise has been added to the data, information is generally lost. Some of the sources of noise are:

- low frequency thermal drift,
- line frequency interference,
- induced RF signals, and
- harmonic distortion.

Every effort should be taken to eliminate the generation of noise, rather than to eliminate it later by filtering or other techniques. Most system noise problems can be solved by good installation practices.

Noise may be introduced external to the data acquisition system, or it may be induced by the system itself. Some sources of external noise are:

sensors	improper installation of the sensor
cabling	improper shielding and grounding (see Section B.1.5)

B.2.1 Input Bandwidth Considerations

One of the primary sources of noise within the data acquisition system is the instrumentation amplifier (IA) itself. This is true because the input to the IA is interfaced to the sensor and its wiring, and the lowest signal level in the system is usually at this amplifier's input terminals. Careful consideration should be given to matching the instrumentation amplifier to the system needs. For example, wide band amplifiers typically exhibit larger sources of error than narrower band amplifiers. Also, when the amplifier bandwidth significantly exceeds the signal bandwidth, increased measurement errors can result from the noise. Therefore, the amplifier's bandwidth should not greatly exceed that required for the desired signal and sampling rate used. With wider amplifier bandwidth, dc performance is poorer and generated noise is greater.

B.2.2 Multiple Gain Stages

One common problem is caused by incorrect allocation of gain between amplification stages. In general, whenever there are multiple gain stages in the signal path, the gain should be maximized in the first stage of an amplifier-filter combination. This has the effect of reducing the noise contribution of the later amplifier stages. When the signal path includes a low-pass filter, pre-filter gain should be maximized. The limiting condition is that the signal, with all of its frequency components, must not overload the pre-filter amplification stage(s).

B.2.3 Limiting Noise

In many applications sensors are installed at relatively long distances from the instrumentation, typically several meters to hundreds of meters. Further, these sensors frequently generate rather low level signals. For example thermocouples, strain gages, RTDs, etc. generate signals in the millivolt range. The application environments are often electrically noisy with large motors, arc furnaces, etc. in relative proximity to the sensors. These environments can have large amounts of induced noise from either 50 or 60 Hz powerlines or high frequency noise from sources such as arc furnaces, induction heaters, and even Radio Frequencies (RF) from near by radio and TV stations and hand held transceivers.

For low bandwidth signals (e.g. slow thermocouples), noise limiting can be a relatively simple such as a low-pass RC filter at the instrumentation input. By using the shielded twisted pair wiring techniques and good instrumentation amplifiers with differential input (Section B.1.2), the corner frequency of the noise limiting filter can be set relatively high. The twisted pair shielded wiring coupled with good CMRR of the instrumentation amplifier (Section B.1.3) should minimize any powerline pickup. The input filter should limit high frequency noise reaching the instrumentation amplifier. In some cases an active filter with good attenuation at the power line frequency may be desirable.

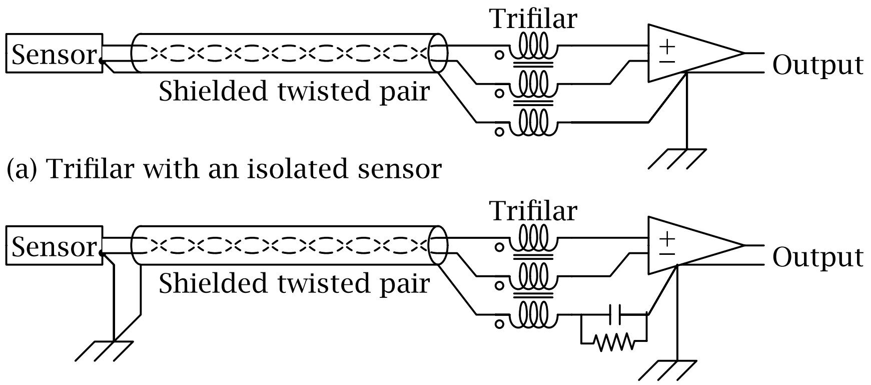
B.2.4 Wideband Noise Limiting and Trifilar Transformers

In applications where good accuracy is important, it is necessary to minimize high frequency common mode noise at the input to the instrumentation amplifier. This requirement is due to the fact that the CMRR of instrumentation amplifiers is typically very good at low frequencies (e.g. 60 Hz), but at high frequencies the common-mode rejection generally is quite poor. Further, in very noisy environments, there can be significant levels of common mode RF energy which can drive the instrumentation amplifier into a non-linear region, causing signal distortion as well as d.c. offset from even small non-linearities. In short the sensor lead acts as an antenna!

A unique method of minimizing the effects of high frequency noise is through the use of the trifilar transformer. By transformer action it prevents most of the high-frequency common-mode noise from reaching the input nodes of the instrumentation amplifier. The use of trifilar transformers is illustrated in Figure B.9.

Figure B.9 (a) illustrates the use of a trifilar transformer with an isolated sensor. Figure B.9 (b) illustrates the use with a *grounded* sensor. The capacitor value is chosen so that the transformer begins functioning where the instrumentation amplifier CMRR starts to deteriorate *(typically at about 50 KHz)* and remains effective into the megahertz region. The resistor provides a ground return for the sensor signal in event that the sensor ground is not present.

A dc shield ground at the instrumentation amplifier is undesirable because of the potential of ground loops, however for the trifilar to do any good in blocking common mode high frequency noise to the instrumentation amplifier a path for the high frequency noise current must be provided from the shield to the instrumentation ground. The 0.001 μf capacitor between the shield and instrumentation ground is provided for this purpose.



(a) Trifilar with an isolated sensor

(b) Trifilar with a grounded sensor

Figure B.9: Trifilar transformer for limiting high frequency noise.

“the index goes here”