SVM2608

4-Channel, 100 kSamples/s Analog-to-Digital Converter

USER'S MANUAL

P/N: 82-0066-000
Released February 23, 2007

VXI Technology, Inc.

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

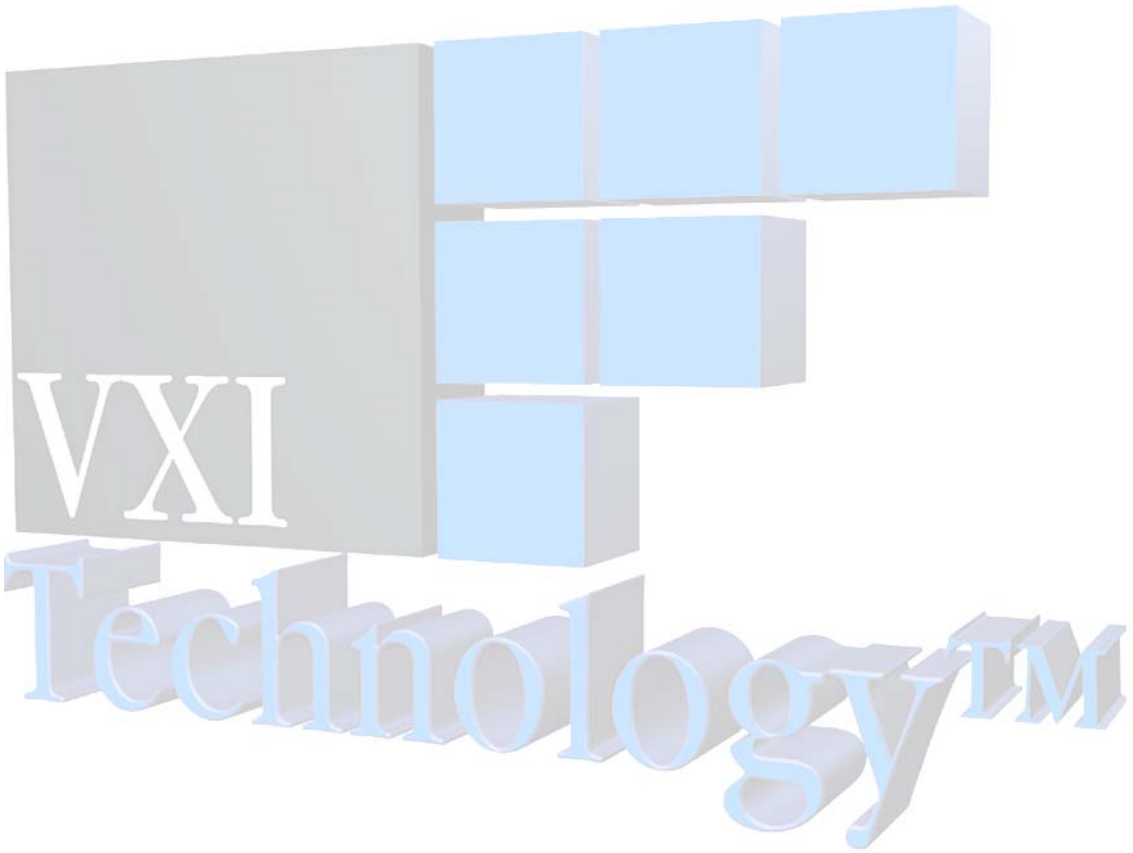


TABLE OF CONTENTS

CERTIFICATION

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

WARRANTY

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

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

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

LIMITATION OF WARRANTY

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

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

RESTRICTED RIGHTS LEGEND

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

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

DECLARATION OF CONFORMITY

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

MANUFACTURER'S NAME	VXI Technology, Inc.
MANUFACTURER'S ADDRESS	2031 Main Street Irvine, California 92614-6509-6509
PRODUCT NAME	4-Channel, 100 kSamples/s Analog-to-Digital Converter
MODEL NUMBER(S)	SVM2608
PRODUCT OPTIONS	All
PRODUCT CONFIGURATIONS	All

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

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

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

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

February 2007

CE

Steve Mauga, QA Manager

GENERAL SAFETY INSTRUCTIONS

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

Service should only be performed by qualified personnel.

TERMS AND SYMBOLS

These terms may appear in this manual:

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

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

These symbols may appear on the product:

ATTENTION - Important safety instructions

Frame or chassis ground

Indicates that the product was manufactured after August 13, 2005. This mark is placed in accordance with *EN 50419, Marking of electrical and electronic equipment in accordance with Article 11(2) of Directive 2002/96/EC (WEEE)*. End-of-life product can be returned to VTI by obtaining an RMA number. Fees for take-back and recycling will apply if not prohibited by national law.

WARNINGS

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

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

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

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

WARNINGS (CONT.)

Avoid Electric Shock

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

Ground the Product

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

Operating Conditions

To avoid injury, electric shock or fire hazard:

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

Improper Use

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

SUPPORT RESOURCES

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

VXI Technology
World Headquarters

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

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

VXI Technology
Cleveland Instrument Division

5425 Warner Road
Suite 13
Valley View, OH 44125

Phone: (216) 447-8950
Fax: (216) 447-8951

VXI Technology
Lake Stevens Instrument Division

VXI Technology, Inc.
1924 - 203 Bickford
Snohomish, WA 98290

Phone: (425) 212-2285
Fax: (425) 212-2289

Technical Support

Phone: (949) 955-1894
Fax: (949) 955-3041
E-mail: support@vxitech.com

Visit http://www.vxitech.com for worldwide support sites and service plan information.

SECTION 1

INTRODUCTION

INTRODUCTION

The SVM Series leverages off VXI Technology's line of high-density modular VXIbus instruments, but are optimized for the VMEbus. All SVM instruments are designed to provide all the features of test instrumentation in other platforms such as GPIB or VXI. These features are achieved in hardware rather than in a driver. This approach to the interface design guarantees the user that all communications to the module occur in microseconds, as opposed to several milliseconds, considerably improving system throughput. The board is equipped with a microprocessor which significantly increases the module's functionality and task performing capabilities.

The SVM2608 is a ruggedized circuit card designed for insertion into a convection-cooled VME chassis. It is a double height VME size module (6U) of single slot width and conforms to all physical requirements as specified by VME specifications. The VME interface is compatible with the VME32/64x configuration with two 160-pin (32 x 5) backplane connectors (P0 – P1). The SVM2608 consists of four low-speed (100 kSamples/s) channels and, with the addition of the -01 Option, can include two, high-speed (20 MHz) channels.

OVERVIEW

The SVM2608 is a precision, four channel digitizer capable of capturing data on all four channels simultaneously either in FIFO (or "*real-time*") mode or Linear (or "*burst"*) mode. A processor enables the user to perform a variety of calculations with the data acquired. Each channel is also capable of measuring voltage and resistance. All four channels can measure voltage at the same time, but resistance can only be measured one channel at a time. Resistance can be measured in two different modes: 2-wire or 4-wire. Both modes use a local current source to inject a current into the resistor under test and then measure voltage across the resistor.

All four channels are independent of one another. The front end of each channel has both a variable gain amplifier and an attenuator, thus allowing for full ADC scale measurements of signals from 1 V to 100 V. Before being digitized, the signal can be passed through a Low Pass Filter (LFP) with a cut-off frequency of 20 kHz. The ADC is a 16-bit converter capable of taking as many as 100 kSamples/s on a scale of -10 V to +10 V. To compensate for offset and gain variation in the ADCs, each channel has two 12-bit DACs that are used to calibrate the offset and the gain on each channel ADC. These calibrations are performed at the factory using precision voltage reference sources.

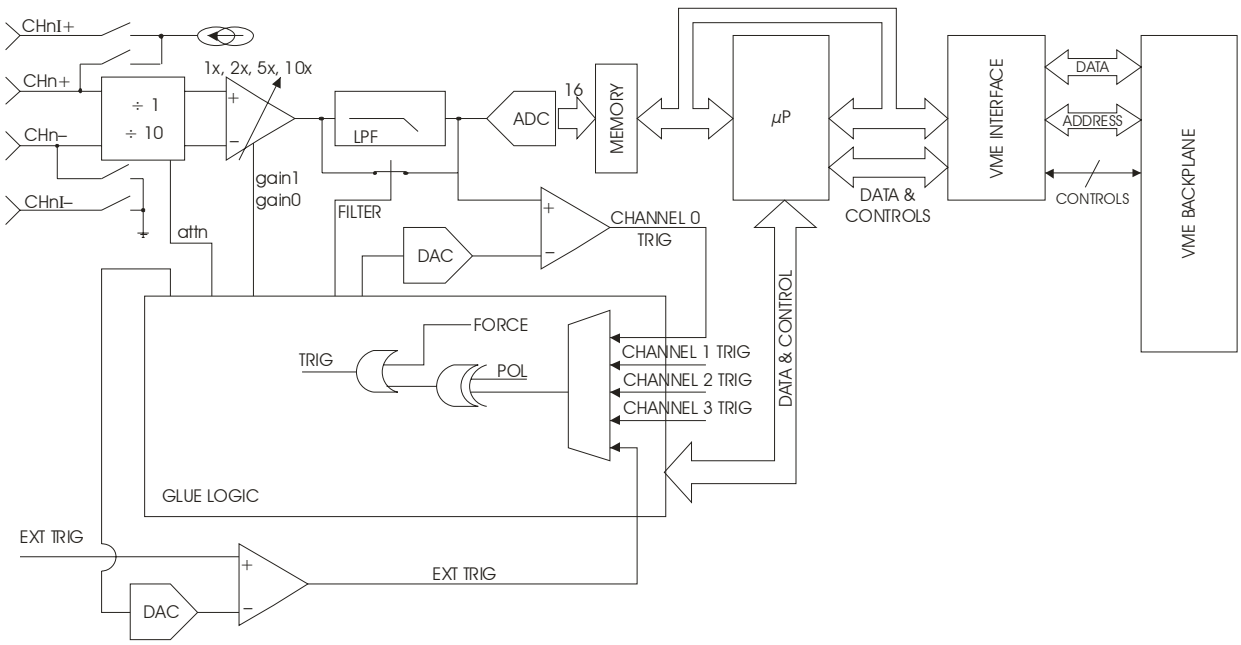


FIGURE 1-1: SVM2608 BLOCK DIAGRAM

The acquisition process is controlled by two FPGAs that allow for greater flexibility along with higher speed and precision during the digitizing process. As the data is digitized, it is placed in memory. It is then available to the user through the VME interface. Each channel has its own channel memory that can store up to one million samples of data. This data is also made available to the microprocessor for data processing. The samples are stored as words (16 bits). The first sample of a channel is located at the channel's base address at Offset 0 (0x000000 for Channel 0, 0x200000 for Channel 1, 0x400000 for Channel 2 and 0x600000 for Channel 3). The next sample is located at Offset 2 (0x000002 for Channel 0, etc.) and the third sample is located at Offset 4, etc.

In order to provide better resolution for the measurement, the input signal is amplified accordingly to generate a ±10 V_{P-P} signal at the input of the ADC. Thus, on different scales, the weight of a bit of digitized data will be different:

SCALE (V)	Bit Weight (μV/count)
1	30.518
2	61.035
5	152.588
10	305.176
20	610.352
50	1525.879

The following equation is used to determine the bit weight at a specified scale:

Bit Weight = Full Scale / 32,768

For example, the Bit Weight of the 10 V range is:

10.0 volts / 32768 = 0.0003051757813 V/count
≈ 305.176 μV/count

The counts stored in memory are 16 bits **SIGNED** integers. The most significant bit represents the SIGN. Thus, the hex number 0x4000 and the hex number of 0xC000 represent the same signal amplitude but in opposite directions, where 0x4000 represents a positive signal while 0xC000 is a negative signal with the same amplitude.

To translate a raw count value into a voltage, multiply the raw count value by the bit weight. The following example shows this conversion for a SVM2608 using the 10.0 V range:

A reading of 0x4000 = 16,384 counts
Voltage = Counts * Bit Weight
Voltage = 16384 counts* 305.176 µV/count
≈ 5.0 volts.

A reading of 0xC000 = -16,384 counts
Voltage = -16384 counts * 305.176 µV/count
≈ -5.0 volts.

Similarly, for the 5 volt range:

A reading of 0x4000 = 16,384 counts
Voltage = 16,384 counts * 152.588 µV/count
≈ 2.5 volts.

Data acquisition can be made in two modes: ***Linear*** or ***FIFO***. In *FIFO* mode, data can only be read from a fixed address (FIFO register), while in *Linear* mode, data can be read from any address in the memory space of a channel. Linear mode also offers a two more options for acquisition: ***Pre-Trigger*** and ***Delayed Trigger***.

In FIFO mode, data can be retrieved while the acquisition is still in progress. However, if the memory is not read and the acquisition continues running, new incoming data will overwrite older data and the older data will be lost. It is also NOT possible to run a measurement command in FIFO mode.

Acquiring Data

To acquire data, a channel must first be Armed. When a channel is armed, it starts its local Sample Clock and waits for a Trigger Event to begin sampling. The channel must remain Armed for the entire duration of the acquisition process. Clearing the ARM bit will reset the internal state-machines and stop acquisition. Data capturing starts when a Trigger Event occurs. A trigger event can be caused by an external trigger source, the signal under test or forced by setting a bit in a register.

Triggering

An external signal, other than one of the sampled signals, can be used to trigger any or all of the channels. This external signal is compared to a threshold level set by a local DAC and a high-speed comparator is used to generate an External Trigger signal.

The signal under test can also be used to trigger an acquisition. The signal is compared to a threshold level set by a local DAC and a high-speed comparator is used to generate a Channel Trigger signal. Each channel has its own DAC and its own comparator, thus, each channel can generate a Trigger signal independent of the other channels. Acquisition on any channel can be triggered by any other Channel Trigger signal (Channel 0 can be triggered by *Channel 0 Trigger***,** *Channel 1 Trigger***,** *Channel 2 Trigger***,** or *Channel 3 Trigger*) even if the other channels are not armed and are not acquiring any data. Only one channel can be the trigger source at any time and the trigger sources cannot be AND'ed or OR'ed together.

In absence of a Trigger signal, the acquisition can be forced by setting a control bit, the FORCE bit. Forcing an acquisition on a channel only starts acquisition on that channel. Each channel has its own corresponding FORCE bit.

Linear Mode

In Linear mode, the total number of samples collected (also referred to as *Sample Points*) is determined by the value programmed into the Sample Points register. The first sample (also referred to as *Sample Zero*) is stored in memory when a Trigger Event occurs. Sample Zero is from the value read from the channel's base address at offset zero (0x000000 for Channel 0, 0x200000 for Channel 1, 0x400000 for Channel 2 and 0x600000 for Channel 3).

Pre-Trigger

In Linear mode, it is also possible to store samples that occur before a Trigger Event. When a channel is armed and the Pre-Trigger register is programmed with a value other than zero, that channel will begin sampling immediately, without waiting for an External Trigger. After it stores the number of samples specified in the Pre-Trigger register (also referred to as *Pre-Trigger Points*), it begins monitoring the Trigger Event. Until the Trigger Event occurs, the channel continues sampling and storing. When the Trigger Event occurs, Sample Zero is stored. After the Trigger Event, the number of data points collected is determined by the following equation:

AFTER TRIGGER POINTS = SAMPLE POINTS – PRETRIGGER POINTS

When the user reads from offset zero of a channel, the data returned is Sample Zero followed by Sample Zero + 1, etc. The Pre-Trigger Points can be read from the top of that channel's memory. For example, if 0x100 Pre-Trigger Points were sampled on Channel 0 after the acquisition is completed, the samples can be retrieved from locations 0x1FFE00, 0x1FFE02 …0x1FFFFE with the data at 0x1FFFFE being the last Pre-Trigger Sample before the trigger event.

Delayed Trigger

In Linear mode, it is also possible to delay storing Sample Zero by a number of sample clocks, where a sample clock is defined by the Sample Clock Rate register. The number of sample clocks an acquisition is delayed (also referred to as *Delayed Points*) is programmed in the Delayed Trigger register. Samples are taken and stored immediately when the Trigger Event occurs, but Sample Zero will be stored only after the specified number of Delayed Points passes. Data stored during the Delayed period can be viewed by the user at the top of the memory space of the respective channel (same as in Pre-Trigger mode as above), assuming that the following condition is observed:

SAMPLE POINTS < 1 MSamples

As opposed to the Pre-Trigger acquisition, the number of samples taken *after* the Trigger Event is not affected by the number of samples taken *before* it:

AFTER TRIGGER POINTS = SAMPLE POINTS

If the following condition is met:

DELAYED POINTS < 1 M – SAMPLE POINTS

Then all the samples collected before the Trigger Event are available to the user.

FIFO Mode

In FIFO mode, the user can retrieve data from the board as acquisition progresses. The memory behaves as a FIFO: data is written into a circular buffer with new data overwriting older data when the buffer is full. A Threshold Flag is available to monitor the status of the buffer and prevent overwriting the data or under-reading it.

The Sample Points register that is used in Linear mode to determine the amount of data to be captured is used in FIFO mode to determine the size associated with the FIFO Threshold Flag. When the number of samples stored in memory equals the number of points set in the Sample Points register, the FIFO Threshold flag is asserted. In this manner, the user can wait until a certain number of samples are captured before they download data from the board. If the user fails to retrieve the data from the card in time and new data overwrites older data, then the FIFO Overrun flag is asserted. Conversely, if the user attempts to read more data than has been stored, the FIFO Underrun flag is asserted. The FIFO Threshold flag is cleared when data is read from the board and the total amount of "new (unread)" data in the buffer is less than the THRESHOLD value. The FIFO Overrun and Underrun flags are cleared only when a new acquisition is initiated.

Calibrations

Due to the nature of the semiconductors and passive components, not all parts have exactly the same characteristics. Slight differences exist from component to component. While these inconsistencies are unavoidable, they do not affect the basic functionality of the electronic instrumentation. The precision of the instrument, however, can be altered by these variances.

One way to eliminate these slight variations is to use expensive, precision parts or to perform a rigorous parts selection procedure to ensure consistency. These measures, however, would dramatically increase the cost of the board. Another way to compensate for offset and gain variations is to take a number of measurements using precision calibrated instruments of known voltage and resistance. Their known values are then compared against the values attained for each channel and the difference is used to adjust future measurements. These adjustments are called *calibrations*. They are performed at the factory using approved calibration sources.

Test Bus

The SVM2608 is capable of performing a self-test to check for functionality and accuracy. Using a local voltage reference source and local resistance references, basic function tests can be performed. Four different voltage reference sources are available on the board: ±9.45 V and ±0.945 V. Two Resistance References are available: 128 Ω and 81.92 kΩ. Two different signal generators can also be used for different tests: a RAMP generator and a PULSE generator. Any of these locally generated test sources can be placed on the internal Test Bus (TB). The Test Bus can then be connected to the input of any or all of the channels. Only one of the test signals can be connected to the Test Bus at one time. The test sources can be connected to the Test Bus using microprocessor commands. The Test Bus is also available to the user for monitoring on pins 24 and 13 of the Front Panel Connector. (See ***Front Panel Interface Wiring*** for more detail.)

The self-test is performed by sending a command to the microprocessor, instructing it to run the self-test (see *Microprocessor Commands*). When the microprocessor runs the self-test, a Test Result is returned (see the description of the Self Test Command for a more detailed description).

Commands

The SVM2608 is equipped with a processor. While the processor is not directly involved in the acquisition process, its presence on the board significantly enhances the capabilities of the SVM2608 digitizer.

The user can choose to download the data on to a CPU and perform custom data processing, or they can instruct the on-board microprocessor to perform one or several predefined calculations

sets. (See *Microprocessor Commands* for more details on available commands.) The command is sent to the microprocessor via the Command register. Since there are four independent channels on the board, each of them can take a different command and each of them has its own command register. The result of the microprocessor calculation is returned in the Result register for the corresponding channel.

The data stored in the channel memory is *raw* data. When the microprocessor performs a resistance calculation, it uses *calibrated* data, meaning that the microprocessor takes the calibrations for the Local Current Sources (see above) values into consideration. The raw data the user downloads from the board represents calibrated voltage measurements. The result calculated by the microprocessor and placed in the Result register when a Resistance Measurement command is issued is based on calibrated Current measurements. While the user can perform calibrated *voltage* measurements by simply reading the raw data, the calculations for *resistance* cannot be accurately performed by the user as they do not have the calibrated *current* values (the exact values injected in the resistance under test by the board's current source). Although it is possible for the user to read the calibration values (see the ***Calibration Commands*** section) and use the raw data to perform all the calibrated measurements on their own, the manufacturer encourages the use of the microprocessor's capabilities to perform all calibrated resistance calculations.

Option -01

With the addition of the SVM2608-01 option, two additional channels are available with a sample rate of 20 MHz and 12-bits of resolution. This option may be purchased at the same time as the SVM2608 or is factory upgradeable.

The high-speed channels available on Option -01 function independently. The front end of each channel has both a variable gain amplifier and an attenuator, similar to the low-speed channels. A 5 MHz low-pass filter (LPF) is available on these channels, as opposed to the 20 kHz LPF found on the low-speed channels. The ADC converter is a 12 bits converter capable of taking as many as 20 MSamples/s on a scale of -2 V to +2 V. To compensate for offset and gain variation in the ADCs, each channel has two 12 bits DACs that are used to calibrate the offset and the gain on each ADC channel. These calibrations are performed at the factory using precision voltage reference sources. A block diagram for Option -01 is provided on the following page.

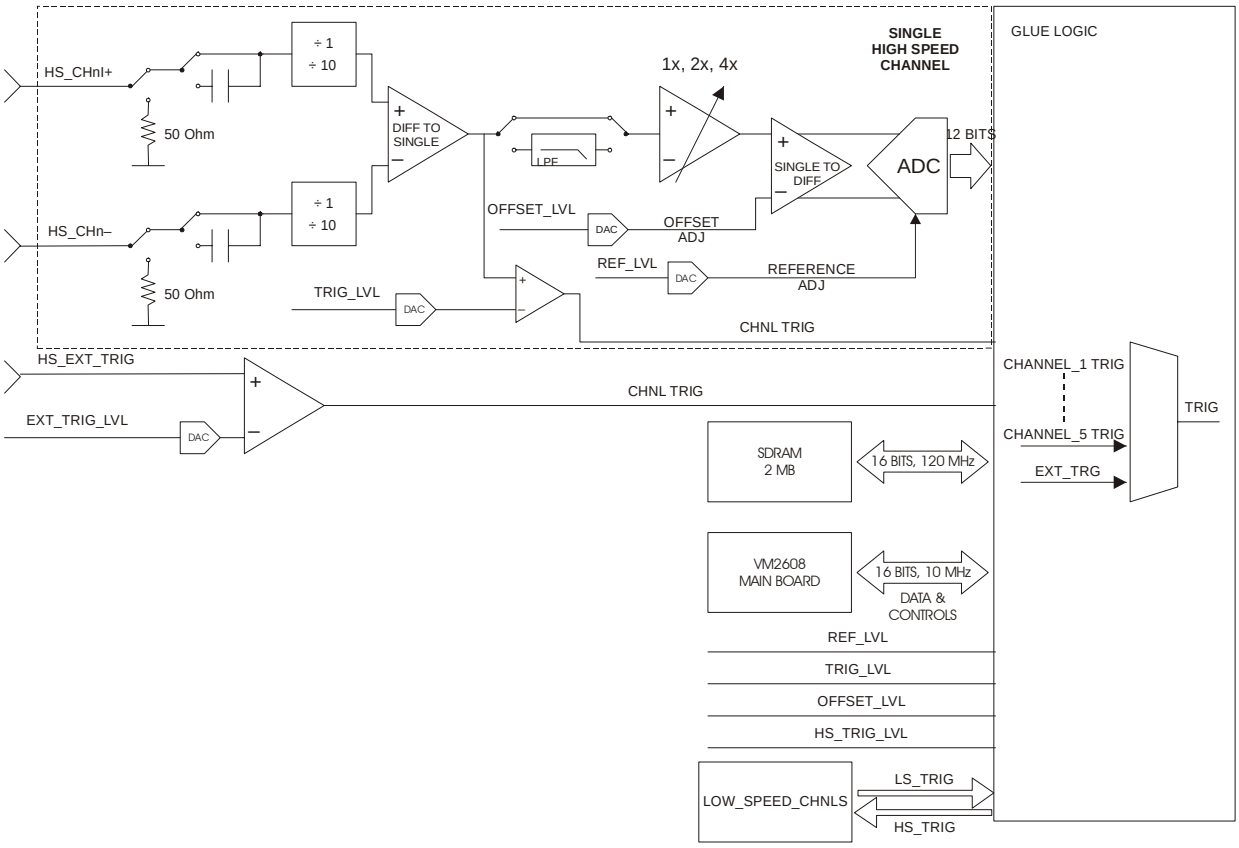


FIGURE 1-2: SVM2608 BLOCK DIAGRAM

PHYSICAL DESCRIPTION

The SVM2608 has a protective coating applied to it to ensure that the effects of environmental hazards are minimized. This coating endows the modules with resistance to salt sprays, moisture, dust, sand, and explosive environments, as the polymer coating provides a hermetic seal. The module is designed to withstand the stress and rigors of shock and vibration, allowing the module to be deployed in a variety of applications without concern for damage due to the surrounding physical environment. The following table details the environmental specifications of this module.

TABLE 1-1: SVM2608 ENVIRONMENTAL SPECIFICATIONS

SVM ENVIRONMENTAL SPECIFICATIONS	
CLASSIFICATION	MIL-T-28800E Type III, Class 5, Style E or F
TEMPERATURE	Meets functional shock requirements of MIL-T-28800E, Type III, Class 5
OPERATIONAL	-20°C to 65°C
NON-OPERATIONAL	-40°C to 71°C
HUMIDITY	5% to 95% (non-condensing)
ALTITUDE	
OPERATIONAL	Sea level to 15,000 ft (4,570 m)
SUSTAINED STORAGE	Sea level to 40,000 ft (12,190 m)
RANDOM VIBRATION	Three axis, 30 minutes total, 10 minutes per axis
OPERATIONAL	0.27 g_{rms} total from 5.0 Hz to 55.0 Hz
NON-OPERATIONAL	2.28 g_{rms} total from 5.0 Hz to 55.0 Hz
FUNCTIONAL SHOCK	Half sine, 30 g, 11 ms duration
SALT ATMOSPHERE	> 48 hrs operation
SAND AND DUST	> 6 hrs operation in a dust environment of 0.3 g/ft^3 blowing at 1750 ft/min

The SVM2608 has two indicator LEDs located on its front panel. The A/E (Access/Error) LED flashes green when read/write commands are being sent to the module. Should the SVM2608 receive an error, the LED glows red. This LED can be overridden by the user by setting the Sysfail bit in the *Reset, Sys Fail Control, Interrupt Levels Register*. The P/F (Power/Fail) LED glows green indicates the status of a processor driven self-test. If the self-test is successful, the P/F LED glows green, if the test fails, however, the LED will glow red. Both the A/E and P/F LEDs can be programmed to glow red when a fail condition occurs.

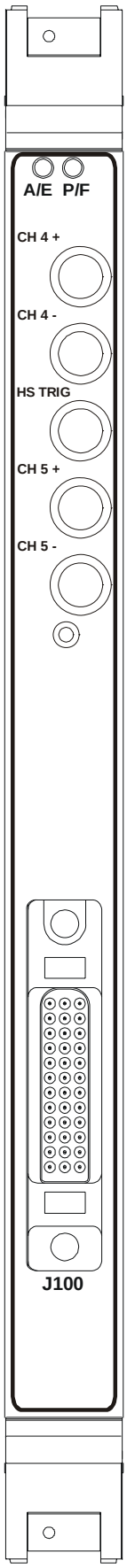


FIGURE 1-3: SVM2608 FRONT PANEL

FRONT PANEL INTERFACE WIRING

Front-panel connector, J101, contains all the instrument signals for the Channels 0 through 3.

PIN NUMBER	SIGNAL	PIN NUMBER	SIGNAL	PIN NUMBER	SIGNAL
1	GND	13	TB-	25	GND
2	CH1I-	14	GND	26	CH3I-
3	CH1-	15	GND	27	CH3-
4	CH1+	16	GND	28	CH3+
5	CH1I+	17	GND	29	CH3I+
6	GND	18	GND	30	GND
7	GND	19	EXTTRIGIN	31	GND
8	CH0I-	20	GND	32	CH2I-
9	CH0-	21	GND	33	CH2-
10	CH0+	22	GND	34	CH2+
11	CH0I+	23	GND	35	CH2I+
12	GND	24	TB+	36	GND

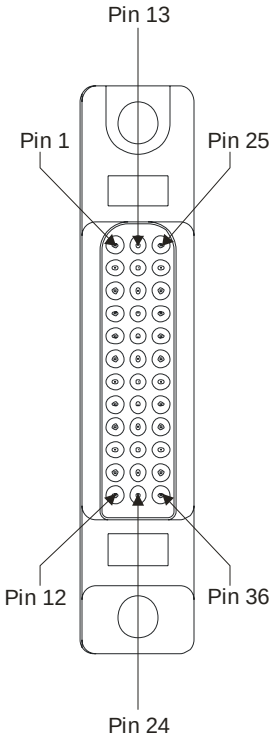


FIGURE 1-4: SVM2608 PIN LOCATIONS

Note	The SMA connectors associated with the high-speed channels are labeled on the front panel and are capable of being triggered by a different external trigger source than the low-speed channels.

SVM2608 SPECIFICATIONS

GENERAL SPECIFICATIONS	
NUMBER OF CHANNELS	
	4
SAMPLING RATE	
Range	0.59 Samples/s to 100.0 kSamples/s
Resolution	100 ns
Accuracy	1%
INPUT POWER	
+5 V dc	500 mA
+12 V dc	300 mA
-12 V dc	300 mA
VMEBUS INTERFACE	
Address mode	A32
Data transfer mode	D16 or D32
MEMORY	
	1 MSamples per channel
VOLTAGE MEASUREMENT	
Range	±1.0 V, ±2.0 V, ±5.0 V, ±10.0 V, ±20.0 V, ±50.0 V
Resolution	$1/2^{15}$ of full scale
Accuracy	1%
INPUT IMPEDANCE	
20 & 50 V Range	200 kΩ
1, 2, 5 & 10 V Range	> 10 MΩ
RESISTANCE MEASUREMENT*	
Range	100 Ω, 1 kΩ, 10 kΩ, 100 kΩ, 1 MΩ, 100% over range*
Resolution	1/6,553.6 of scale
Accuracy	1%
***Note**: *Resistance measurements can only be made one channel at a time.*	
All resistance measurements can be made accurately up to +199% of the set range.	
INPUT FILTER	
	20 kHz (-3 dB)
TRIGGER LEVELS	
Internal	
Range	trigger level determined by selected voltage range (see *Voltage Measurement* above)
Trigger level resolution	$Range/2^{11}$
Level accuracy	1%
External	
Range	-10 V to +10 V
Trigger level resolution	4.88 mV
Level accuracy	1%
DELAYED TRIGGER	
Range	0 samples to $(2^{32} -1)$ samples
Resolution	1 sample
TIMEOUT	
Range	10 µs to 227 hrs
Resolution	10 µs to 100 s
WARM-UP TIME	
	10 min
MTBF	
	80,000 hrs (based on 20% Ground Mobile, 80% Ground Fixed environment, T ≥ 52°C)

OPTION 1 - SVM2608-01	
NUMBER OF CHANNELS	
	2
VMEBUS INTERFACE	
Address Mode	A32
Data Transfer Mode	D16 or D32
SAMPLING RATE	
Range	7.15 Samples/s to 20.0 MSamples/s
Resolution	8.333 ns
Accuracy	1%
MEMORY	
	1 MSamples per channel
VOLTAGE MEASUREMENTS	
Range	±0.5 V, ±1.0 V, ±2.0 V, ±5.0 V, ±10.0 V, ±20.0 V
Resolution	Range/2^{11}
Accuracy	1%
INPUT IMPEDANCE	
	1 MΩ or 50 Ω (software selectable)
INPUT MODE	
	ac or dc (software selectable)
INPUT FILTER	
	5 MHz (-3 dB)
TRIGGER LEVELS	
Internal	
Range	trigger level determined by selected voltage range (see *Voltage Measurement* above)
Trigger level resolution	Full scale/2^{11}
Level accuracy	1%
External	
Range	-10 V to +10 V
Trigger level resolution	4.88 mV
Level accuracy	1%
DELAYED TRIGGER	
Range	0 samples to (2^{24} -1) samples
Resolution	1 sample
TRIGGER ACCURACY	
	1%
EXTERNAL TRIGGER ACCURACY	
	1%
TIMEOUT	
Range	10 µs to 227 hrs
Resolution	10 µs to 100 s
WARM-UP TIME	
	10 min
MTBF	
	80,000 hrs (based on 20% Ground Mobile, 80% Ground Fixed environment, T ≥ 52°C)

SECTION 2

PREPARATION FOR USE

INTRODUCTION

When the SVM2608 is unpacked from its shipping carton, the contents should include the following items:

(1) SVM2608 4 Channel 100 kSamples/s Analog-to-Digital Converter Module
(1) SVM2608 User's Manual (this manual)

All components should be immediately inspected for damage upon receipt of the unit. Installation instructions for the module are discussed in the following pages of this section.

The chassis the SVM2608 is installed in should be checked to ensure that it is capable of providing adequate power and cooling. Once it is found that the chassis meets these specifications, the SVM2608 should be examined. If the module is found to be in good condition, the base address of the SVM2608 and the backplane jumpers of the chassis may be configured. After setting the base address and chassis jumpers, the SVM2608 may be installed into an appropriate 6U VMEbus mainframe in any slot other than slot zero.

CALCULATING SYSTEM POWER AND COOLING REQUIREMENTS

It is imperative that the chassis provide adequate power and cooling for this module. Referring to the chassis operation manual, confirm that the power budget for the system (the chassis and all modules installed therein) is not exceeded and that the cooling system can provide adequate cooling.

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

SETTING THE CHASSIS BACKPLANE JUMPERS

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

SETTING THE BASE ADDRESS

The base address of the SVM2608 is determined by using the offset value (OV), set by two rotary switches located on the top edge of the interface card (Figure 2-1), and multiplying it by 2^{24} (or 16,777,216) to get the base address in A32 address space. The switches are labeled with positions 0 through F. The switch located at **S3** corresponds to the Most Significant Bit (**MSB**) and **S2** corresponds to the Least Significant Bit (**LSB**). (Note, **S1** is not used in the determination of the base address.) To set the OV to **25**, first convert the decimal number to the hexadecimal value of **0x19**. Next, set switch S3 to **1**, and then set switch S2 to **9**. See Figure 2-2. Two conversion examples are provided on the following pages.

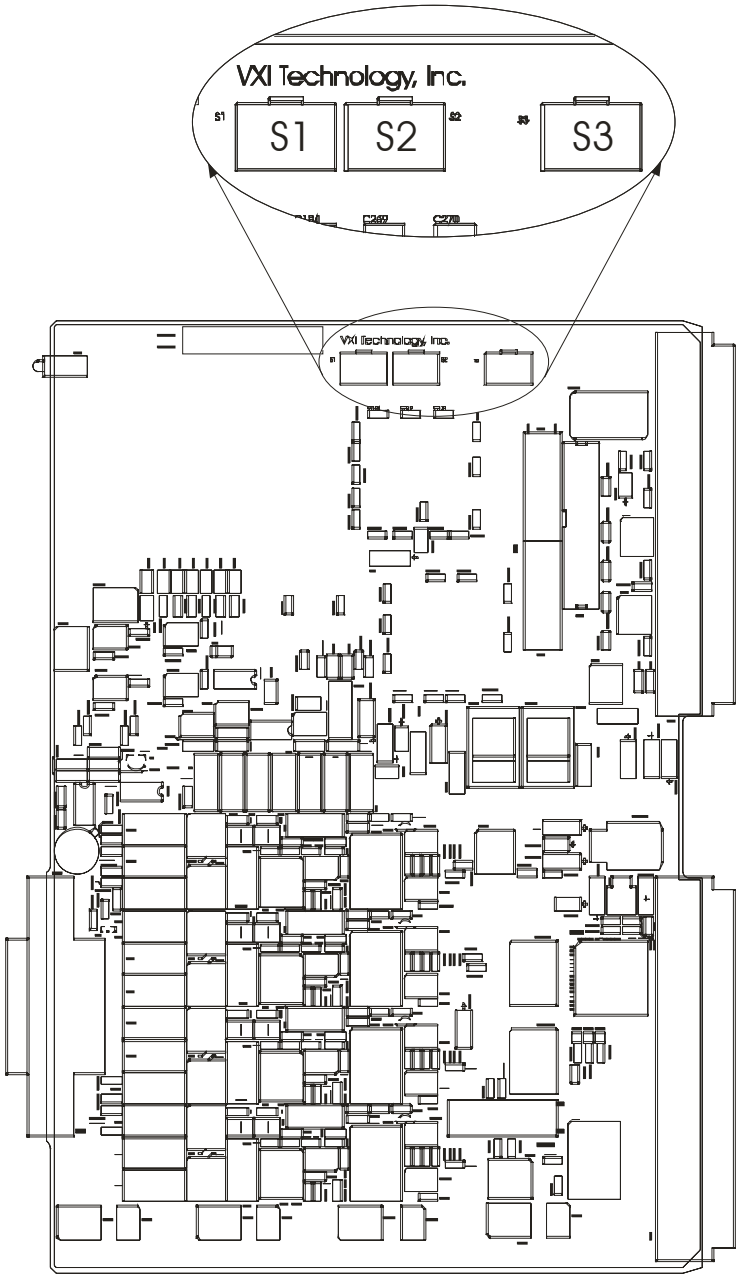


FIGURE 2-1: ROTARY SWITCH LOCATIONS

Example 1

OV (decimal)	Divide by 16		MSB	LSB	
25	25 / 16	=	1	w/ 9 remaining	*Divide the decimal value by 16 to get the MSB and the LSB.*
		=	0001	1001	*The 1 is the MSB, and the remainder of 9 is the LSB.*
		=	1	9	*Convert to hexadecimal. Set the back switch to 1 and the front switch to 9.*

FIGURE 2-2: OFFSET VALUE EXAMPLE 1

Here is another way of looking at the conversion:

OV = (S3 x 16) + S2
OV = (1 x 16) + 9
OV = 16 + 9
OV = 25

The base address is then determined by using the following formula:

A32 Base Address = Offset Value * 0x1000000 *(or 16,777,216)*

In this case:

A32 Base Address = 0x19 * 0x1000000 *(or 16,777,216)*
A32 Base Address = 0x19000000

Example 2

OV (decimal)	Divide by 16		MSB	LSB	
200	200 / 16	=	12	w/ 8 remaining	*Divide by 16.*
		=	1100	1000	*Convert to MSB and LSB.*
		=	C	8	*Convert to hexadecimal. Set the back switch to C and the front switch to 8.*

FIGURE 2-3: OFFSET VALUE EXAMPLE 2

Therefore, the base address in this example is:

A32 Base Address = 0xC8 * 0x1000000 *(or 16,777,216)*
A32 Base Address = 0xC8000000

This information is used to write to the registers of the SVM2608. (See Section 3 for more details on SVM2608 registers.)

MODULE INSTALLATION/REMOVAL

Before installing an SVM2608 module into a 6U VME mainframe, make sure that the mainframe is powered down. Insert the module into the base unit by orienting the module so that the flanges at the edge of the module can be inserted into the slot of the base unit. Position the flanges so that they fit into the module slot groove. Once the module is properly aligned, push the module back and firmly insert it into the backplane connector. The retaining screws can then be used to secure the module in the chassis.

To remove the module, power down the mainframe and remove all cabling from the module. The retaining screws can then be loosened. The ejector handles can then be used as to assist in the removal of the module.

SECTION 3

PROGRAMMING

INTRODUCTION

The SVM2608 modules are VMEbus register-based devices for high-speed D16 or D32 data retrieval. Register-based programming is a series of **reads** and **writes** directly to the module registers. This eliminates the time for command parsing thus increasing speed.

DEVICE MEMORY MAPS

Function Offset

The function offset helps define where in A32 space a WRITE or READ operation is performed. The offsets are defined as follows:

Function	Decimal Value	Hexadecimal Value
CH0 Data	0	0x000000
CH1 Data	2097152	0x200000
CH2 Data	4194304	0x400000
CH3 Data	6291456	0x600000
CH4 (Option -01)	8388608	0x800000
CH5 (Option -01)	10485760	0xA00000
Registers	12582912	0xC00000
Reserved	14680064	0xE00000

CH0 – 5 Data These addresses are used to store data.

Registers These addresses are the A32 memory registers. They are used to program the settings for each channel, collect FIFO data, collect results or sent commands to the microprocessor.

Reserved These addresses are reserved for future use.

Register Offset

The register offset is located within the module's A32 address space. When data is sent to a register address, the address that is written to is the sum of the module base address, the function offset and the register offset:

Register Address = Module Base Address + Function Offset + Register Offset

Table 3-1 shows the A32 map of the SVM2608 registers.

TABLE 3-1: SVM2608 A32 REGISTER MAP

Note	*MS = Most Significant*	*LS = Least Significant*

OFFSET	WRITE FUNCTION	READ FUNCTION
0x00	Sysfail Control, Interrupt Levels	Sysfail Control, Interrupt Levels
0x02	Force Trigger/Start	Force Trigger/Start
0x04	Reserved	Reserved
0x06	External Trigger Level	External Trigger Level
0x08	Control (Channel 0)	Control (Channel 0)
0x0A	Trigger Level (Channel 0)	Trigger Level (Channel 0)
0x0C	Sample Rate (Channel 0) – (MS)	Sample Rate (Channel 0) – (MS)
0x0E	Sample Rate (Channel 0) – (LS)	Sample Rate (Channel 0) – (LS)
0x10	Sample Points (Channel 0) – (MS)	Sample Points (Channel 0) – (MS)
0x12	Sample Points (Channel 0) – (LS)	Sample Points (Channel 0) – (LS)
0x14	Pre-Trigger Points (Channel 0) – (MS)	Pre-Trigger Points (Channel 0) – (MS)
0x16	Pre-Trigger Points (Channel 0) – (LS)	Pre-Trigger Points (Channel 0) – (LS)
0x18	Trigger Delay (Channel 0) – (MS)	Trigger Delay (Channel 0) – (MS)
0x1A	Trigger Delay (Channel 0) – (LS)	Trigger Delay (Channel 0) – (LS)
0x1C	Timeout (Channel 0)	Timeout (Channel 0)
0x1E	Interrupt Enable (Channel 0)	Interrupt Enable (Channel 0)
0x20	Reserved	Interrupt Status (Channel 0)
0x22	Command Register (Channel 0)	Command Register (Channel 0)
0x24	Reserved	FIFO Data (Channel 0) – (MS)
0x26	Reserved	FIFO Data (Channel 0) – (LS)
0x28	Reserved	Result Register (Channel 0) – (MS)
0x2A	Reserved	Result Register (Channel 0) – (LS)
0x2C	Reserved	Result Register (Channel 0) – (MS)
0x2E	Reserved	Result Register (Channel 0) – (LS)
0x30	Control (Channel 1)	Control (Channel 1)
0x32	Trigger Level (Channel 1)	Trigger Level (Channel 1)
0x34	Sample Rate (Channel 1) – (MS)	Sample Rate (Channel 1) – (MS)
0x36	Sample Rate (Channel 1) – (LS)	Sample Rate (Channel 1) – (LS)
0x38	Sample Points (Channel 1) – (MS)	Sample Points (Channel 1) – (MS)
0x3A	Sample Points (Channel 1) – (LS)	Sample Points (Channel 1) – (LS)
0x3C	Pre-Trigger Points (Channel 1) – (MS)	Pre-Trigger Points (Channel 1) – (MS)
0x3E	Pre-Trigger Points (Channel 1) – (LS)	Pre-Trigger Points (Channel 1) – (LS)
0x40	Trigger Delay (Channel 1) – (MS)	Trigger Delay (Channel 1) – (MS)
0x42	Trigger Delay (Channel 1) – (LS)	Trigger Delay (Channel 1) – (LS)
0x44	Timeout (Channel 1)	Timeout (Channel 1)
0x46	Interrupt Enable (Channel 1)	Interrupt Enable (Channel 1)
0x48	Reserved	Interrupt Status (Channel 1)
0x4A	Command Register (Channel 1)	Command Register (Channel 1)
0x4C	Reserved	FIFO Data (Channel 1) – (MS)
0x4E	Reserved	FIFO Data (Channel 1) – (LS)
0x50	Reserved	Result Register (Channel 1) – (MS)
0x52	Reserved	Result Register (Channel 1) – (LS)
0x54	Reserved	Result Register (Channel 1) – (MS)

OFFSET	WRITE FUNCTION	READ FUNCTION
0x56	Reserved	Result Register (Channel 1) – (LS)
0x58	Control (Channel 2)	Control (Channel 2)
0x5A	Trigger Level (Channel 2)	Trigger Level (Channel 2)
0x5C	Sample Rate (Channel 2) – (MS)	Sample Rate (Channel 2) – (MS)
0x5E	Sample Rate (Channel 2) – (LS)	Sample Rate (Channel 2) – (LS)
0x60	Sample Points (Channel 2) – (MS)	Sample Points (Channel 2) – (MS)
0x62	Sample Points (Channel 2) – (LS)	Sample Points (Channel 2) – (LS)
0x64	Pre-Trigger Points (Channel 2) – (MS)	Pre-Trigger Points (Channel 2) – (MS)
0x66	Pre-Trigger Points (Channel 2) – (LS)	Pre-Trigger Points (Channel 2) – (LS)
0x68	Trigger Delay (Channel 2) – (MS)	Trigger Delay (Channel 2) – (MS)
0x6A	Trigger Delay (Channel 2) – (LS)	Trigger Delay (Channel 2) – (LS)
0x6C	Timeout (Channel 2)	Timeout (Channel 2)
0x6E	Interrupt Enable (Channel 2)	Interrupt Enable (Channel 2)
0x70	Reserved	Interrupt Status (Channel 2)
0x72	Command Register (Channel 2)	Command Register (Channel 2)
0x74	Reserved	FIFO Data (Channel 2) – (MS)
0x76	Reserved	FIFO Data (Channel 2) – (LS)
0x78	Reserved	Result Register (Channel 2) – (MS)
0x7A	Reserved	Result Register (Channel 2) – (LS)
0x7C	Reserved	Result Register (Channel 2) – (MS)
0x7E	Reserved	Result Register (Channel 2) – (LS)
0x80	Control (Channel 3)	Control (Channel 3)
0x82	Trigger Level (Channel 3)	Trigger Level (Channel 3)
0x84	Sample Rate (Channel 3) – (MS)	Sample Rate (Channel 3) – (MS)
0x86	Sample Rate (Channel 3) – (LS)	Sample Rate (Channel 3) – (LS)
0x88	Sample Points (Channel 3) – (MS)	Sample Points (Channel 3) – (MS)
0x8A	Sample Points (Channel 3) – (LS)	Sample Points (Channel 3) – (LS)
0x8C	Pre-Trigger Points (Channel 3) – (MS)	Pre-Trigger Points (Channel 3) – (MS)
0x8E	Pre-Trigger Points (Channel 3) – (LS)	Pre-Trigger Points (Channel 3) – (LS)
0x90	Trigger Delay (Channel 3) – (MS)	Trigger Delay (Channel 3) – (MS)
0x92	Trigger Delay (Channel 3) – (LS)	Trigger Delay (Channel 3) – (LS)
0x94	Timeout (Channel 3)	Timeout (Channel 3)
0x96	Interrupt Enable (Channel 3)	Interrupt Enable (Channel 3)
0x98	Reserved	Interrupt Status (Channel 3)
0x9A	Command Register (Channel 3)	Command Register (Channel 3)
0x9C	Reserved	FIFO Data (Channel 3) – (MS)
0x9E	Reserved	FIFO Data (Channel 3) – (LS)
0xA0	Reserved	Result Register (Channel 3) – (MS)
0xA2	Reserved	Result Register (Channel 3) – (LS)
0xA4	Reserved	Result Register (Channel 3) – (MS)
0xA6	Reserved	Result Register (Channel 3) – (LS)
0xA8	Control (Channel 4)	Control (Channel 4)
0xAA	Trigger Level (Channel 4)	Trigger Level (Channel 4)
0xAC	Sample Rate (Channel 4) – (MS)	Sample Rate (Channel 4) – (MS)
0xAE	Sample Rate (Channel 4) – (LS)	Sample Rate (Channel 4) – (LS)
0xB0	Sample Points (Channel 4) – (MS)	Sample Points (Channel 4) – (MS)
0xB2	Sample Points (Channel 4) – (LS)	Sample Points (Channel 4) – (LS)
0xB4	Pre-Trigger Points (Channel 4) – (MS)	Pre-Trigger Points (Channel 4) – (MS)

OFFSET	WRITE FUNCTION	READ FUNCTION
0xB6	Pre-Trigger Points (Channel 4) – (LS)	Pre-Trigger Points (Channel 4) – (LS)
0xB8	Trigger Delay (Channel 4) – (MS)	Trigger Delay (Channel 4) – (MS)
0xBA	Trigger Delay (Channel 4) – (LS)	Trigger Delay (Channel 4) – (LS)
0xBC	Timeout (Channel 4)	Timeout (Channel 4)
0xBE	Interrupt Enable (Channel 4)	Interrupt Enable (Channel 4)
0xC0	Reserved	Interrupt Status (Channel 4)
0xC2	Command Register (Channel 4)	Command Register (Channel 4)
0xC4	Reserved	FIFO Data (Channel 4) – (MS)
0xC6	Reserved	FIFO Data (Channel 4) – (LS)
0xC8	Reserved	Result Register (Channel 4) – (MS)
0xCA	Reserved	Result Register (Channel 4) – (LS)
0xCC	Reserved	Result Register (Channel 4) – (MS)
0xCE	Reserved	Result Register (Channel 4) – (LS)
0xD0	Control (Channel 5)	Control (Channel 5)
0xD2	Trigger Level (Channel 5)	Trigger Level (Channel 5)
0xD4	Sample Rate (Channel 5) – (MS)	Sample Rate (Channel 5) – (MS)
0xD6	Sample Rate (Channel 5) – (LS)	Sample Rate (Channel 5) – (LS)
0xD8	Sample Points (Channel 5) – (MS)	Sample Points (Channel 5) – (MS)
0xDA	Sample Points (Channel 5) – (LS)	Sample Points (Channel 5) – (LS)
0xDC	Pre-Trigger Points (Channel 5) – (MS)	Pre-Trigger Points (Channel 5) – (MS)
0xDE	Pre-Trigger Points (Channel 5) – (LS)	Pre-Trigger Points (Channel 5) – (LS)
0xE0	Trigger Delay (Channel 5) – (MS)	Trigger Delay (Channel 5) – (MS)
0xE2	Trigger Delay (Channel 5) – (LS)	Trigger Delay (Channel 5) – (LS)
0xE4	Timeout (Channel 5)	Timeout (Channel 5)
0xE6	Interrupt Enable (Channel 5)	Interrupt Enable (Channel 5)
0xE8	Reserved	Interrupt Status (Channel 5)
0xEA	Command Register (Channel 5)	Command Register (Channel 5)
0xEC	Reserved	FIFO Data (Channel 5) – (MS)
0xEE	Reserved	FIFO Data (Channel 5) – (LS)
0xF0	Reserved	Result Register (Channel 5) – (MS)
0xF2	Reserved	Result Register (Channel 5) – (LS)
0xF4	Reserved	Result Register (Channel 5) – (MS)
0xF6	Reserved	Result Register (Channel 5) – (LS)
0xF8	Reserved	Reserved
0xFA	Reserved	Reserved
0xFC	Reserved	Reserved
0xFE	External Trigger Level for High-Speed Channel	External Trigger Level for High-Speed Channel

DATA(BYTE) ORDERING

When a pair of 16-bit registers is read as a 32-bit register, the content of the register marked MS is placed on the VME Bus on D31 - D16 and the content of the register marked LS is placed on D15 - D0. Similarly, when a pair of 16-bit registers is written as a 32-bit register, the register marked MS is loaded with the data present on the VME Bus on D31 - D16 and the register marked LS is loaded with the data present on D15 - D0. All other registers should be addressed as 16-bit registers to prevent any malfunctioning.

With a variety of systems and bridges that move the data between different bus types (i.e. VME to PCI, VXI to PCI, etc.), in order to assist the user in determining how data is ordered, a known floating point value of 0.12345678901234 is loaded at Power-Up in the Result Register for all channels. Channel 0 values are listed as an illustration:

0x3FBF	is written at address	0xC00028
0x9ADD	is written at address	0xC0002A
0x3746	is written at address	0xC0002C
0xF4C6	is written at address	0xC0002E

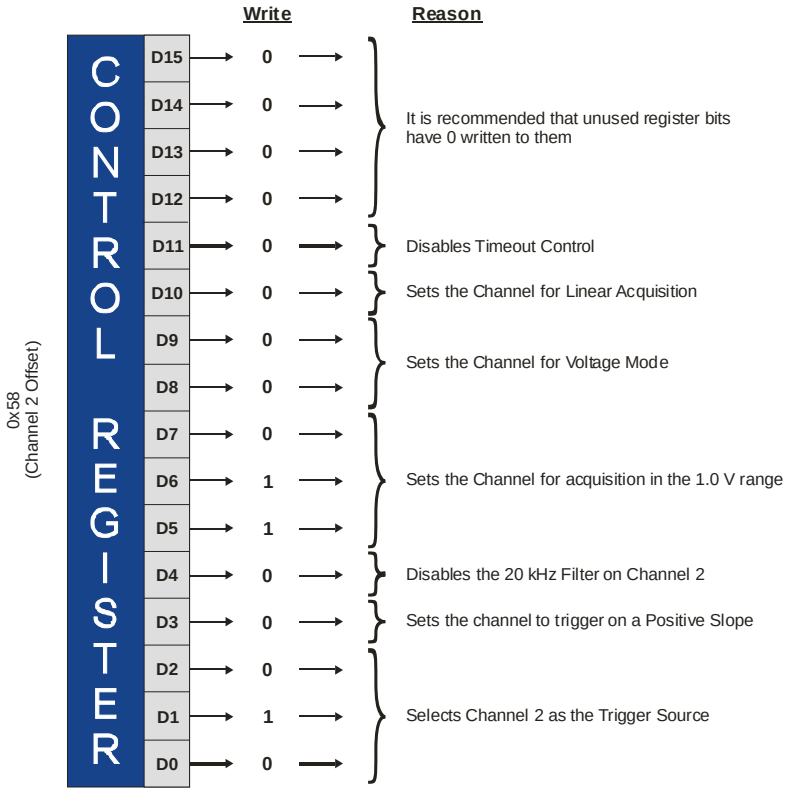
By reading the value from these addresses, the user can identify the type of DATA(BYTE) swapping that takes place in the system and modify their code accordingly. An example of how to do the swapping is presented in Appendix A.

DETERMINING THE REGISTER ADDRESS

A user wishes to set Channel 2 to the 1.0 V range. Data is to be captured linearly without the use of the low pass filter or timeout control and will trigger from the positive edge of data sent to Channel 2. To accomplish this, the user will access the Control Register for Channel 2 at register offset 0x0058. To determine the register address, this value must be added to the base address and A32 address of the module. In this example, it is assumed that the base address switches are set to 0x19, yielding a base address of 0x19000000. Since the user must write to a register, the function offset is 0x00C00000.

Register Address = Module Base Address + Function Offset + Register Offset
= 0x19000000 + 0x00C00000 + 0x00000058
= 0x19C00058

By observing the bits in the Control Register, it can be determined what data value should be sent:



The binary values are then converted into a hexadecimal format:

Binary	Hexadecimal
0000 0000 0110 0010	0x0062

This determines the data value required for the aforementioned settings.

ACCESSING THE REGISTERS

With both D16 and D32 data transfer available, the user can write either 16 or 32 bits of data to the registers. To change the settings of the module, it is only necessary to write a 16- or 32-bit integer to the specified register with the new configuration.

All registers, as defined in the following section, are 16-bit registers. A 32-bit write can be made to registers that are located in consecutive addresses. The consecutive 16-bit registers that can be accessed as 32-bit registers are:

Sample Rate Register (0x0C, 0x34, 0x5C, 0x84, 0xAC, 0xD4)

Sample Points Register (0x10, 0x38, 0x60, 0x88, 0xB0, 0xD8)

Pre-Trigger Points Register (0x14, 0x3C, 0x64, 0x8C, 0xB4, 0xDC)

Trigger Delay Register (0x18, 0x40, 0x68, 0x90, 0xB8, 0xE0)

FIFO data (0x24, 0x4C, 0x74, 0x9C, 0xC4, 0xEC)

Result Register (0x28 & 0x2C, 0x50 & 0x54, 0x78 & 0x7C, 0xA0 & 0xA4, 0xC8 & 0xCB, 0xF0 & 0xF4)

NOTE	*Reading 32 bits from a 16-bit register may generate a **BERR** on the VME bus.* Writing 32 bits to a 16-bit register may generate a **BERR** on the VME bus or may corrupt data in another register.

DESCRIPTION OF REGISTERS

The following pages describe the registers found in the SVM2608 Register Map for A32 address space that starts at 0x00C0000. When multiple channels registers have the same functions, the offsets appear in parenthesis separated by commas with Channel 0 being listed first, followed by Channel 1, etc. For example, the description used by the Control Register Bit is applicable to all six channels at offsets 0x08 for Channel 0, 0x30 for Channel 1, 0x58 for Channel 2 and 0x80 for Channel 3, 0xA8 for Channel 4 and 0xD0 for Channel 5. This is indicated in the register description by using the following notation: (0x08, 0x30, 0x58, 0x80, 0xA8, 0xD0). Unless otherwise noted, register descriptions apply to all channels (Channels 0 – 5).

Reset, Sys Fail Control, Interrupt Levels Register (0x00) — Read & Write		
D15	Unused	This bit is reserved for future use.
D14	SYSFAILCTL	**System Fail Control** - This bit controls whether or not the sysfail line will be masked. 0 = Card can assert sysfail line. 1 = Sysfail line is masked and card cannot assert sysfail line. P_{on} state = 0
D13 – D3	Unused	These bits are reserved for future use.

Reset, Sys Fail Control, Interrupt Levels Register (0x00) — Read & Write		
D2 – D0	INTLVL2 - 0	**Interrupt Level** - These bits determine the interrupt service level. 111 = Interrupt level 7 110 = Interrupt level 6 101 = Interrupt level 5 100 = Interrupt level 4 011 = Interrupt level 3 010 = Interrupt level 2 001 = Interrupt level 1 000 = No interrupt

Force Trigger, Start Register (0x02) — Read & Write		
D15 – D13	HS_TRIGSRC2 - 0	**High-Speed Trigger Source** – These bits select a trigger source for the high-speed channels. 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3 100 = Invalid state 101 = Invalid state 110 = External 111 = Invalid state **Note**: These bits are only utilized by high-speed Channels 4 and 5. These bits are unused for Channels 0 – 3.
D12	EXT TRIG SLOPE	**External Trigger Slope** – This bit sets the slope of the external trigger for low-speed Channels 0 - 3. 0 = Positive 1 = Negative **Note**: This bit is only utilized by high-speed Channels 4 and 5. This bit is unused for Channels 0 – 3.
D11 – D6	FTRIG5 - 0	**Force Trigger** - All of the channels have the ability to be triggered via software when in the arm mode. Acquisition begins when trigger is forced. These bits need to be reset to '0' in order to allow subsequent triggers (it is the transition of a bit from 0 to 1 that forces a trigger). One bit is assigned to each channel as follows: D6 for Channel 0 D7 for Channel 1 ⋮ D11 for Channel 5 Having one bit per channel allows multiple channels to be triggered simultaneously. 0 = Do not force trigger 1 = Force software trigger P_{on} state =0

Force Trigger, Start Register (0x02) — Read & Write		
D5 – D0	START5 – 0	**Acquisition Armed** - These bits control whether or not the specified channel is to be armed for an acquisition. A channel must remain ARMED for the entire duration of the acquisition process. Clearing an ARM bit will reset the internal state-machines and stop the acquisition. One bit is assigned to each channel as follows: D0 for Channel 0 D1 for Channel 1 ⋮ D5 for Channel 5 Having one bit per channel allows multiple channels to be triggered simultaneously. 0 = Channel not armed for acquisition 1 = Channel armed and ready for acquisition P_{on} state = 0

Reserved (0x04)		
D15 – D0	Reserved	These bits are reserved for future use.

External Trigger Level (0x06) — Read & Write		
D15 – D12	Unused	These bits are reserved for future use.
D11 – D0	External Trigger Level	Sets the level at which the module triggers from an external source.

Control Register (0x08, 0x30, 0x58, 0x80, 0xA8, 0xD0) — Read & Write		
D15 – D14	Unused	These bits are reserved for future use.
D13	AC/DC Coupling	**AC/DC Select** - This bit selects between ac and dc coupling for high-speed Channels 4 – 5. 0 = AC 1 = dc P_{on} state= 0 **Note**: This bit is only utilized by high-speed Channels 4 and 5. This bit is unused for Channels 0 – 3.
D12	1 MΩ/50 Ω	**1 MΩ /50 Ohms** - Selects between the 1 MΩ and 50 Ω 0 = 1 MΩ 1 = 50 Ω P_{on} state= 0 **Note**: This bit is only utilized by High-Speed Channels 4 and 5. This bit is unused for Channels 0 – 3.
D11	TIMEOUTCTL	**Timeout Control** - This bit controls whether or not a timeout condition will cause the timeout bit to be set in the interrupt status register. 0 = Disable timeout status bit 1 = Enable timeout status bit P_{on} state = 0

Control Register (0x08, 0x30, 0x58, 0x80, 0xA8, 0xD0) — Read & Write		
D10	LINEAR/FIFO	**Linear or FIFO Memory Mode** - Determines whether the data is captured in Linear (burst) or FIFO (real time) acquisition mode. 0 = Linear mode 1 = FIFO mode P_{on} state = 0
D9 - D8	2WIREOHMS/ 4WIREOHMS	**Function Setting** - The digitizer is capable of measuring voltages, resistances in 2-wire mode or resistances in 4-wire mode. Before taking a measurement, allow for at least 5 ms for internal circuits to settle after making changes. 00 = Voltage mode 01 = 4-wire resistance mode 10 = 2-wire resistance mode 11 = Invalid state **Note**: These bits are only utilized by Channels 0 – 3. These bits are unused by Channels 4 and 5.
D7 – D5	ATTN-GAIN1-GAIN0	**Attenuation and Gain Setting** - Valid attenuation settings are x1 and x10. Valid gain settings are 1, 2, 5, and 10. Note that there is only one current source shared among all four channels, therefore, only one resistance measurement can be made at any one time. Before taking a measurement, allow for at least 5 ms for internal circuits to settle after making changes. VOLTAGE MODE (for channels 0-3) 011 = 1.0 V 010 = 2.0 V 001 = 5.0 V 000 = 10.0 V 110 = 20.0 V 101 = 50.0 V 100 = Invalid state 111 = Invalid state VOLTAGE MODE (for channels 4-5) 011 = 0.5 V 001 = 1.0 V 000 = 2.0 V 111 = 5.0 V 101 = 10.0 V 100 = 20.0 V 010 = Invalid state 110 = Invalid state RESISTANCE MODE (Valid for channels 0-3) 011 = 100 Ω; 2 mA current source setting 000 = 1 kΩ; 2 mA current source setting 000 = 10 kΩ; 200 μA current source setting 000 = 100 kΩ; 20 μA current source setting 000 = 1 MΩ; 2 μA current source setting

Control Register (0x08, 0x30, 0x58, 0x80, 0xA8, 0xD0) — Read & Write		
D4	FILTER	**20 kHz (Channels 0-3)/5 MHz (Channels 4-5) LPF Control** – This bit enables/disables the low pass frequency filter for the low-speed channels and high-speed channels, respectively. . Before taking a measurement, allow for at least 5 ms for internal circuits to settle after making changes. 0 = Filter off 1 = Filter on P_{on} state = 0
D3	TRGSLOPE	**Input Trigger Source Slope** – This bit sets the slope of the input trigger for low-speed Channels 0 - 3. 0 = Positive 1 = Negative P_{on} state = 0
D2 – D0	TRIGSRC2 - 0	**Trigger Source Control** - Once the trigger is armed, the acquisition can be triggered via software (FTRIG), any of the six channels or an external trigger. 000 = Channel 0 001 = Channel 1 010 = Channel 2 011 = Channel 3 100 = Channel 4 101 = Channel 5 110 = External 111 = External High-Speed

Trigger Level (0x0A, 0x32, 0x5A, 0x82, 0xAA, 0xD2) — Read & Write		
D15 – D12	Unused	This is reserved for future use.
D11 – D0	TRGLVL	**Trigger Level Threshold Setting** – These bits set the trigger threshold (12-bit value). 0x800 = 0 V 0x000 = -ve full scale 0xFFF = +ve full scale

Sample Rate (0x0C, 0x34, 0x5C, 0x84) — Read & Write		
D15 – D9	Unused	These bits are reserved for future use.
D8 – D0	SAMPRAT24 – 16	**Sample Interval** – These bits set the sample rate. Bit Weight = 100 ns/bit Minimum Value = 100 Maximum Value = $2^{24} – 1$

Sample Rate (0x0E, 0x36, 0x5E, 0x86) — Read & Write		
D15 – D0	SAMPRAT15 – 0	**Sample Interval** – These bits set the sample rate. Bit Weight = 100 ns/bit Minimum Value = 100 Maximum Value = $2^{24} – 1$

Sample Rate, High-Speed (0xAC, 0xD4) — Read & Write		
D15 – D9	Unused	These bits are reserved for future use.
D8 – D0	SAMPRAT24 – 16	**Sample Interval** – These bits set the high speed sample rate. Bit Weight = 8.333 ns/bit Minimum Value = 6 Maximum Value = $2^{24} – 1$

Sample Rate, High-Speed (0xAE, 0xD6) — Read & Write		
D15 – D0	SAMPRAT15 – 0	**Sample Interval** – These bits set the high speed sample rate. Bit Weight = 8.333 ns/bit Minimum Value = 6 Maximum Value = $2^{24} – 1$

Sample Points (0x10, 0x38, 0x60, 0x88, 0xB0, 0xD8) — Read & Write		
D15 – D4	Unused	These bits are reserved for future use.
D3 – D0	SIZE19 – 16	**Waveform Capture Size** - In Linear mode, this sets the number of data points to be captured in a single sweep. In FIFO mode, this sets the threshold for generating a service request interrupt or status bit. The maximum size is 1 MSamples (2 Mbytes of data).

Sample Points (0x12, 0x3A, 0x62, 0x8A, 0xB2, 0xDA) — Read & Write		
D15 – D0	SIZE15 – 0	**Waveform Capture Size** - In Linear mode, this sets the number of data points to be captured in a single sweep. In FIFO mode, this sets the threshold for generating a service request interrupt or status bit. The maximum size is 1 MSamples (2 Mbytes of data).

Pre-Trigger Points (0x14, 0x3C, 0x64, 0x8C, 0xB4, 0xDC) — Read & Write		
D15 – D4	Unused	These bits are reserved for future use.
D3 – D0	PRE19 – 16	**Size of Pre-Trigger Data** – In Linear mode, this sets the number of data points to be stored before the trigger occurs. In FIFO mode, this register is not used. Minimum Value = 0 Maximum Value = (Capture Size) – 1

Pre-Trigger Points (0x16, 0x3E, 0x66, 0x8E, 0xB6, 0xDE) — Read & Write		
D15 – D0	PRE15 – 0	**Size of Pre-Trigger Data** – In Linear mode, this sets the number of data points to be stored before the trigger occurs. In FIFO mode, this register is not used. Minimum Value = 0 Maximum Value = (Capture Size) – 1

Trigger Delay (0x18, 0x40, 0x68, 0x90, 0xB8, 0xE0) — Read & Write		
D15 – D0	DELAY31 – 16	**Trigger Delay** - The trigger delay value is used to set the number of samples between trigger occurrence and storing of data. Minimum Value = 0 Maximum Value = $2^{32} – 1$

Trigger Delay (0x1A, 0x42, 0x6A, 0x92, 0xBA, 0xE2) — Read & Write		
D15 – D0	DELAY15 – 0	**Trigger Delay** - The trigger delay value is used to set the time between trigger occurrence and storing of data. Minimum Value = 0 Maximum Value = $2^{32} - 1$

Timeout (0x1C, 0x44, 0x6C, 0x94, 0xBC, 0xE4) — Read & Write		
D15 – D13	TOSEL2 – 0	**Timeout Select** - This sets the resolution of the timeout counter. It will determine how long the device waits between start and trigger events before setting the timeout status bits. 000 - 10 μs 001 - 100 μs 010 - 1 ms 011 - 10 ms 100 - 100 ms 101 - 1.0 s 110 - 10.0 s 111 - 100.0 s
D12 – D0	TO12 – 0	**Timeout Counter Setting** - This value multiplied by the timeout select value yields the actual timeout setting.

Interrupt Enable (0x1E, 0x46, 0x6E, 0x96, 0xBE, 0xE6) — Read & Write		
D15 – D0	IM15 – 0	**Interrupt Enable** - Enables or disables interrupt generation for corresponding bit in the interrupt status register. 0 = Disabled 1 = Enabled P_{on} state = 0

Interrupt Status (0x20, 0x48, 0x70, 0x98, 0xC0, 0xE8) — Read Only		
D15 – D0	INT15 – 0	**Interrupt Status Register** - If a corresponding bit in the interrupt enable register is set, these bits indicate if an interrupt occurred. Otherwise, they operate as status bits. 15 - Real time trigger input value 14 - This bit is reserved for future use 13 - This bit is reserved for future use 12 - Settled (range change ready) 11 - Result ready 10 - FIFO overrun 9 - FIFO underrun 8 - FIFO @ threshold 7 - ARM not ready - ERROR if ARM too soon (warning) 6 - Triggered - cleared when acquisition is complete 5 - Capture Complete 4 - Function busy 3 - Self test busy 2 - Self test passed 1 - CMD error 0 - Timeout occurred

Command Register (0x22, 0x4A, 0x72, 0x9A, 0xC2, 0xEA) — Read and Write		
D15 – D0	CMD15 – 0	**Command Register** - Writing to this register instructs the microprocessor to perform the specified function. If this command performs a calculation, the data is returned into the corresponding result register. For a detailed description of the commands, refer to the *Microprocessor Commands* section.

FIFO Data (0x24, 0x4C, 0x74, 0x9C, 0xC4, 0xEC) — Read Only		
D15 – D0	FIFODATA31 – 16	**FIFO Data** - Two registers are provided for retrieving FIFO data. This allows for 32-bit transfer. One sample of data per 16-bit register.

FIFO Data (0x26, 0x4E, 0x76, 0x9E, 0xC6, 0xEE) — Read Only		
D15 – D0	FIFODATA15 – 0	**FIFO Data** - Two registers are provided for retrieving FIFO data. This allows for 32-bit transfer. One sample of data per 16-bit register.

Results Register (0x28, 0x50, 0x78, 0xA0, 0xC8, 0xF0) — Read Only		
D15 – D0	RESULT63 – 48	**Result Data** - When a process data command is issued to the microprocessor, bits 63 through 48 of the 64-bit floating point result is returned to this register. A status bit in the interrupt register is set.

Results Register (0x2A, 0x52, 0x7A, 0xA2, 0xCA, 0xF2) — Read Only		
D15 – D0	RESULT47 – 32	**Result Data** - When a process data command is issued to the microprocessor, bits 47 through 32 of the 64-bit floating point result is returned to this register. A status bit in the interrupt register is set.

Results Register (0x2C, 0x54, 0x7C, 0xA4, 0xCB, 0xF4) — Read Only		
D15 – D0	RESULT31 – 16	**Result Data** - When a process data command is issued to the microprocessor, bits 31 through 16 of the 64-bit floating point result is returned to this register. A status bit in the interrupt register is set.

Results Register (0x2E, 0x56, 0x7E, 0xA6, 0xCE, 0xF6) — Read Only		
D15 – D0	RESULT15 – 0	**Result Data** - When a process data command is issued to the microprocessor, bits 15 through 0 of the 64-bit floating point result is returned to this register. A status bit in the interrupt register is set.

Reserved Registers (0xF8 – 0xFC)		
D15 – D0	Unused	This is reserved for future use.

External Trigger Level - High-Speed Channels (0xFE) — Read & Write		
D15	HS_EXT_ TRIG_SLOPE	**High-Speed Input Trigger Source Slope** – This bit sets the slope of the input trigger for high-speed Channels 4 and 5. 0 = Positive 1 = Negative P_{on} state = 0
D14 – D12	Unused	These bits are reserved for future use.
D11 – D 0	HS_EXT_TRIG 11 - 0	**High-Speed External Trigger** – Sets the level at which the high-speed module triggers from an external source.

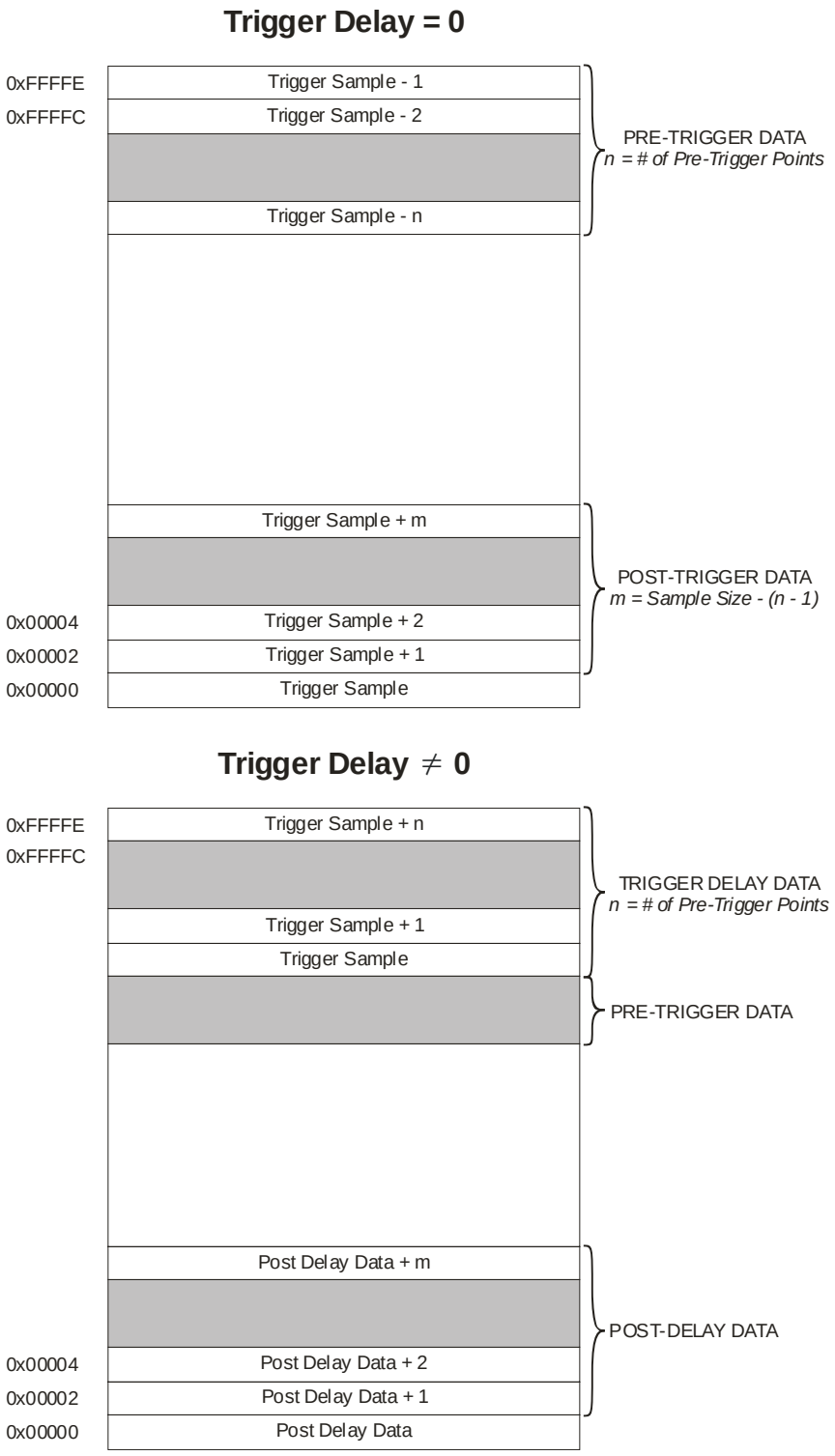


FIGURE 3-1: MEMORY MAP

MICROPROCESSOR COMMANDS

Unless otherwise specified, commands are issued to each channel's *command register*. After issuing a command, the user must wait until the command is executed before issuing a new command to the same channel. The module sets the Command Register to all zeros when a command is finished executing. Issuing a new command before a command completes will yield unpredictable results and may place the board into an unknown state. It is acceptable, however, to issue a command to a different channel without waiting for the current channel to finish execution.

Measurement Commands

The following is a list of the measurement commands:

NOTE	The currents generated by the current sources are listed here for REFERENCE only. The board is not designed to be a ***precision*** current source. These current sources are used for resistance measurements, but all the calculations are adjusted to the internally calibrated values.

0x0001 = Peak Voltage Calculation
0x0002 = DC Voltage Calculation
0x0003 = RMS Voltage Calculation
0x0004 = Peak-to-Peak Voltage Calculation
0x0005 = 100 Ω Range Resistance Measurement (2-wire)
0x0006 = 1 kΩ Range Resistance Measurement (2-wire)
0x0007 = 10 kΩ Range Resistance Measurement (2-wire)
0x0008 = 100 kΩ Range Resistance Measurement (2-wire)
0x0009 = 1 MΩ Range Resistance Measurement (2-wire)
0x000A = Auto-range Resistance Measurement (2-wire)
0x000B = 100 Ω Range Resistance Measurement (4-wire)
0x000C = 1 kΩ Range Resistance Measurement (4-wire)
0x000D = 10 kΩ Range Resistance Measurement (4-wire)
0x000E = 100 kΩ Range Resistance Measurement (4-wire)
0x000F = 1 MΩ Range Resistance Measurement (4-wire)
0x0010 = Auto-Range Resistance Measurement (4-wire)
0x0011 = Perform Self-Test
0x0012 = 1 V Range Voltage Measurement
0x0013 = 2 V Range Voltage Measurement
0x0014 = 5 V Range Voltage Measurement
0x0015 = 10 V Range Voltage Measurement
0x0016 = 20 V Range Voltage Measurement
0x0017 = 50 V Range Voltage Measurement ***(for Channels 0-3 Only)***
0x0019 = Auto Range Voltage Measurement
0x001A = Minimum Voltage Calculation
0x001B = Resistance Calculation
0x001C = Set current source for 100 Ω / 1 kΩ measurement (≈ 2 mA)
0x001D = Set current source for 10 kΩ measurement (≈ 0.2 mA)
0x001E = Set current source for 100 kΩ measurement (≈ 0.02 mA)
0x001F = Set current source for 1 MΩ measurement (≈ 0.002 mA)
0x0020 = Correct setup with calibrations (set offset & gain DACs and calibrated trigger levels based on calibration values)
0x0021 = 0.5 V Range Voltage Measurement ***(for Channels 4-5 Only)***

0x0022 = 100 Ω Range Resistance Measurement - Offset Method (2-wire)
0x0023 = 1 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0024 = 10 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0025 = 100 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0026 = 1 MΩ Range Resistance Measurement - Offset Method (2-wire)
0x0027 = Auto-Range Resistance Measurement - Offset Method (2-wire)
0x0028 = 100 Ω Range Resistance Measurement - Offset Method (4-wire)
0x0029 = 1 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002A = 10 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002B = 100 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002C = 1 MΩ Range Resistance Measurement - Offset Method (4-wire)
0x002D = Auto-Range Resistance Measurement - Offset Method (4-wire)
0x002E = 100 Ω Range Resistance Measurement - Dynamic Method (2-wire)
0x002F = 1 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0030 = 10 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0031 = 100 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0032 = 1 MΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0033 = Auto-Range Resistance Measurement - Dynamic Method (2-wire)
0x0034 = 100 Ω Range Resistance Measurement - Dynamic Method (4-wire)
0x0035 = 1 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0036 = 10 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0037 = 100 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0038 = 1 MΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0039 = Auto-Range Resistance Measurement - Dynamic Method (4-wire)

Captured Data Calculations

The following commands perform calculations on previously captured data:
:

0x0001 = Peak Voltage
0x0002 = DC Voltage
0x0003 = RMS Voltage
0x0004 = Peak-to-Peak Voltage
0x001A = Minimum Voltage Calculation
0x001B = Generic Resistance Calculation

The user sets the ***Sampling Rate***, ***Sampling Points***, ***Measurement*** (voltage, resistance), ***Range*** and ***Trigger Event*** (signal or forced), collects the data. Once the data collection is complete, one of the above commands is issued. The microprocessor performs the calculation based on the captured data and returns the result in the Results Register.

Note that after data is collected, calculations can be performed without doing additional data captures. So, for example, it is possible to calculate dc voltage and then calculate peak-to-peak voltage on the same data collection.

Resistance Measurement – Offset Method

The value returned by the resistance measurement offset commands (0x0022 through 0x002D) is calculated using two current values. A voltage is measured when a current (I) is applied to the circuit (V_{ON}) as well as when the current is not applied (V_{OFF}). The offset resistance value is then calculated using the following formula:

$$\text{Offset Resistance Measurement} = \frac{V_{ON} - V_{OFF}}{I}$$

This might be useful when trying to measure a resistance in the presence of a voltage.

Resistance Measurement – Dynamic Method

The dynamic measurement resistance commands (0x002E – 0x0039) are used to determine a resistance value for a given current while taking the non-linear nature of the current versus voltage curve produced by diodes. The measurement is performed using two currents. The initial current, I_1, produces the initial voltage, V_1. A second, lower current, I_2, creates a second voltage, V_2. The dynamic resistance measurement value is calculated using the following formula:

$$\text{Dynamic Resistance Measurement} = \frac{V_1 - V_2}{I_1 - I_2}$$

When dynamic resistance is measured in the lowest current range (highest resistance range), no "lower" current value exists. In this instance, current is turned off for the second measurement ($I_2 = 0$). In effect, this measurement is the same as an offset resistance measurement.

Self Test Command

The Perform Self Test (0x0011) command instructs the processor to perform an internal test using the on-board reference voltage and on-board reference resistors. The purpose of this test is to verify the functionality and accuracy of the system. A failure is indicated if the measurement is not within 0.8% of the correct value.

This command can be sent to each channel independently to perform a self test on that channel, the result of the self test is placed at the base offset for each channel (i.e. 0x000000 for Channel 0, 0x200000 for Channel 1, etc.). Each one of the 32 bits indicates the FAILURE (bit = 1) or SUCCESS (bit = 0) of a test as follows:

For Channels 0 – 3:

Bit 0 - Measures +0.945 V on the 1 V scale
Bit 1 - Measures -0.945 V on the 1 V scale
Bit 2 - Measures +0.945 V on the 2 V scale
Bit 3 - Measures -0.945 V on the 2 V scale
Bit 4 - Measures +0.945 V on the 5 V scale
Bit 5 - Measures -0.945 V on the 5 V scale
Bit 6 - Measures +9.45 V on the 10 V scale
Bit 7 - Measures -9.45 V on the 10 V scale
Bit 8 - Measures +9.45 V on the 20 V scale
Bit 9 - Measures -9.45 V on the 20 V scale
Bit 10 - Measures +9.45 V on the 50 V scale
Bit 11 - Measures -9.45 V on the 50 V scale
Bit 12 - Measures 128 Ω on the 100 Ω scale
Bit 13 - Measures 128 Ω on the 1 kΩ scale

Bit 14 - Measures 128 Ω on the 10 kΩ scale
Bit 15 - Measures 81.92 kΩ on the 100 kΩ scale
Bit 16 - Measures 81.92 kΩ on the 1 MΩ scale
Bits 17 – 31 are not used and read as "0".

For Channels 4-5:

Bit 0 - Measures +0.117 V on the 0.5 V scale
Bit 1 - Measures -0.117 V on the 0.5 V scale
Bit 2 - Measures +0.945 V on the 1 V scale
Bit 3 - Measures -0.945 V on the 1 V scale
Bit 4 - Measures +0.945 V on the 2 V scale
Bit 5 - Measures -0.945 V on the 2 V scale
Bit 6 - Measures +0.945 V on the 5 V scale
Bit 7 - Measures -0.945 V on the 5 V scale
Bit 8 - Measures +9.45 V on the 10 V scale
Bit 9 - Measures -9.45 V on the 10 V scale
Bit 10 - Measures +9.45 V on the 20 V scale
Bit 11 - Measures -9.45 V on the 20 V scale
Bit 12 – 31 are not used and read as "0".

Example 1

Reading 0x0000000C after a Self Test command indicated a problem during Self Test 0 and Self Test 1, measuring ±0.945 V on the 2 V scale, but no problem on all other tests. This would indicate that the front end programmable gain amplifier has a problem on the x5 (gain = 5) setting.

Example 2

Reading 0x00000000 after a Self Test operation would indicate that all Self Tests passed successfully.

Preset Setting Measurement Commands

The following measurement commands are executed by the microprocessor using pre-set, factory-programmed settings:

0x0005 = 100 Ω Range Query (2-wire)
0x0006 = 1 kΩ Range Query (2-wire)
0x0007 = 10 kΩ Range Query (2-wire)
0x0008 = 100 kΩ Range Query (2-wire)
0x0009 = 1 MΩ Range Query (2-wire)
0x000A = Auto-range Query (2-wire)
0x000B = 100 Ω range query (4-wire)
0x000C = 1 kΩ range query (4-wire)
0x000D = 10 kΩ range query (4-wire)
0x000E = 100 kΩ range query (4-wire)
0x000F = 1 MΩ range query (4-wire)
0x0010 = Auto-range query (4-wire)
0x0012 = 1 V Measurement
0x0013 = 2 V Measurement
0x0014 = 5 V Measurement
0x0015 = 10 V Measurement

0x0016 = 20 V Measurement
0x0017 = 50 V Measurement
0x0018 = 100 V Measurement
0x0019 = Auto Voltage Measurement
0x0021 = 0.5 V Range Voltage Measurement ***(for Channels 4-5 Only)***
0x0022 = 100 Ω Range Resistance Measurement - Offset Method (2-wire)
0x0023 = 1 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0024 = 10 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0025 = 100 kΩ Range Resistance Measurement - Offset Method (2-wire)
0x0026 = 1 MΩ Range Resistance Measurement - Offset Method (2-wire)
0x0027 = Auto-Range Resistance Measurement - Offset Method (2-wire)
0x0028 = 100 Ω Range Resistance Measurement - Offset Method (4-wire)
0x0029 = 1 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002A = 10 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002B = 100 kΩ Range Resistance Measurement - Offset Method (4-wire)
0x002C = 1 MΩ Range Resistance Measurement - Offset Method (4-wire)
0x002D = Auto-Range Resistance Measurement - Offset Method (4-wire)
0x002E = 100 Ω Range Resistance Measurement - Dynamic Method (2-wire)
0x002F = 1 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0030 = 10 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0031 = 100 kΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0032 = 1 MΩ Range Resistance Measurement - Dynamic Method (2-wire)
0x0033 = Auto-Range Resistance Measurement - Dynamic Method (2-wire)
0x0034 = 100 Ω Range Resistance Measurement - Dynamic Method (4-wire)
0x0035 = 1 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0036 = 10 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0037 = 100 kΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0038 = 1 MΩ Range Resistance Measurement - Dynamic Method (4-wire)
0x0039 = Auto-Range Resistance Measurement - Dynamic Method (4-wire)

The pre-set values are as follows:

SAMPLE POINTS	1667
SAMPLE RATE	10 µs
RANGE/TYPE (2- or 4-wires resistance)	Implicit in the command definition.
FILTER	This setting is unaltered by the processor and remains at the value set previously.
TRIGGER EVENT	FORCED TRIGGER

When one of these commands is issued, the microprocessor performs a Data Capture, it calculates the result and returns the result in the corresponding result register. This gives a measurement over one 60 Hz power line cycle to help reject power line noise.

Calibration Commands

The following commands are used at the factory to calibrate the equipment and are presented here for reference only.

WARNING: Calibration commands should only be executed by qualified personnel. If you want to perform your own calibrations please contact factory for more information.

0x0020	Correct setup with calibrations
0x1001	Store Calibration data in non-volatile memory.
0x1002	Read a Cal Gain DAC (lower 12 bits of 16-bit value).
0x1003	Set a Cal Gain DAC (lower 12 bits of a 16-bit value).
0x1004	Read a Cal Offset DAC (lower 12 bits of 16-bit value).
0x1005	Set a Cal Offset DAC (lower 12 bits of 16-bit value).
0x1006	Read a Trigger DAC Gain calibration value (32-bit value about 0x00010000)
0x1007	Set a Trigger DAC Gain calibration value.
0x1008	Read a Trigger DAC Offset calibration value.
0x1009	Set a Trigger DAC Offset calibration value.
0x100A	Read an external Trigger DAC gain calibration value (32-bit value about 0x00010000)
0x100B	Set an external Trigger DAC gain calibration value.
0x100C	Read an external trigger DAC offset calibration value (signed 16-bit value).
0x100D	Set an external Trigger DAC Offset value.
0x100E	Read an ohms calibration gain calibration value (32-bit value about 0x00010000).
0x100F	Set an ohms calibration gain calibration value.
0x1010	Read an ohms calibration offset calibration value (signed 16-bit value).
0x1011	Set an ohms calibration offset calibration value
0x1012	Read a 128 Ω gain, 32-bit 0x00010000 nominal.
0x1013	Set 128 Ω gain (32-bit 0x00010000).
0x1014	Read 81.92k gain (32-bit 0x00010000).
0x1015	Set 81.92k gain (32-bit 0x00010000).
0x1016	Read 9.45 V gain (32-bit 0x00010000).
0x1017	Set 9.45 V gain (32-bit 0x00010000).
0x1018	Read -9.45 V gain (32-bit 0x00010000).
0x1019	Set -9.45 V gain (32-bit 0x00010000).
0x101A	Read 0.945 V gain (32-bit 0x00010000).
0x101B	Set 0.945 V gain (32-bit 0x00010000).
0x101C	Read -0.945 V gain (32-bit 0x00010000).
0x101D	Set -0.945 V gain (32-bit 0x00010000).
0x101E	Read 2 wire Ω offset (32-bit 0x00010000would subtract 1 Ω).
0x101F	Set 2 wire Ω offset (32-bit 0x00010000 would subtract 1 Ω).
0x1020:	Read the serial number, a 32-bit value.
0x1021:	Set the serial number, a 32-bit value
0x1022:	Read the chip IDs, a 16-bit value, lower four for U1, next four for U20.
0x1023:	Return the software revision, 16-bit value with an implied decimal two digits from the right (i.e. 100 is Rev. 1.00.)
0x1024:	Read the error queue.
0x1025	Read V Positive 0.1177 calibration gain value, calibration value.
0x1026	Set the V Positive 0.1177 calibration gain value, calibration value.
0x1027	Read V Negative 0.1177 calibration gain value, calibration value.

0x1028 Set the V Negative 0.1177 calibration gain value, calibration value.

All the calibration commands use the channel's base address at offset zero (0x000000 for Channel 0, 0x200000 for Channel 1, 0x400000 for Channel 2 and 0x600000 for Channel 3) for communication with the processor. This is where the user places the data it wants the processor to write when it issues a "Set …" command. This is also where the processor places the data for the user to read when the user issues a "Read …" command.

When the user issues a "Set …" command, the value is read by the processor from the channel's base address at offset zero and it is stored in its local *volatile* calibration memory. Each calibration value has its own location in the volatile calibration memory.

The 0x1001 command takes the values from the *volatile* calibration memory and stores them in *non-volatile* memory. If the values in the non-volatile calibration memory have been changed by the user, issuing this command will permanently erase the values set by the factory and replace them with the new values set by the user.

The 0x1000 command reloads the values stored in the *non-volatile* memory into the *volatile* calibration memory. This operation is executed by the microprocessor at power-up automatically. This is useful for the user in the event that the calibration memory is accidentally changed and the user wants to recall the factory preset values from the non-volatile memory.

Error Processing

Each channel has its own ERROR queue. Sending the 0x1024 command to a channel's command register will return an error code to the channel's base address at offset zero. If several errors occur at the same time or if the user does not read the error queue to clear the error codes, the error codes will accumulate in a queue. The error queue is five positions deep. Only the first error in the queue is returned when a "Read the error queue command" is sent. When there are no more errors in the queue, the error code returned is 0x0000. Here is the list of error codes returned by the processor:

NO_ERROR	0x0000
ERROR_UNKNOWN_COMMAND	0x0001
ERROR_PRE_GT_SIZE	0x0002
ERROR_RESISTANCE_OVER_RANGE	0x0101
ERROR_UNSTABLE_RESISTANCE	0x0102
ERROR_UNSTABLE_VOLTAGE	0x0103
ERROR_INVALID_RES_CH	0x0104
ERROR_INVALID_CH	0x0105
ERROR_MULTIPLE_TEST_SOURCES	0x0201
ERROR_NONVOL_READ	0x0202
ERROR_NONVOL_WRITE	0x0203
ERROR_NONVOL_DEFAULTED	0x0204
ERROR_FLASH_BURN	0x0205
ERROR_HS_NONVOL_READ	0x0206
ERROR_HS_NONVOL_WRITE	0x0207
ERROR_HS_NONVOL_DEFALTED	0x0208
ERROR_INTERNAL_SOFTWARE	0x0911
ERROR_INTERNAL_RANGE	0x0912
ERROR_QUEUE_OVFL	0xFFFF

The aforementioned errors are reported for the following reasons:

NO_ERROR	*There are no errors in the queue.*
ERROR_UNKNOWN_COMMAND	*An unknown command was sent to the microprocessor.*
ERROR_PRE_GT_SIZE	*The value programmed in the Pre-Trigger Points register is greater than the value programmed in the Sample Points register.*
ERROR_RESISTANCE_OVER_RANGE	*The user attempted to measure a resistance greater than the range.*
ERROR_UNSTABLE_RESISTANCE	*In* Auto-Range Resistance *mode, the resistance is not stable.*
ERROR_UNSTABLE_VOLTAGE	*In* Auto-Range Voltage *mode, the voltage is not stable.*
ERROR_MULTIPLE_TEST_SOURCES	*The user attempted to connect several sources on the Test Bus.*
ERROR_NONVOL_READ	*There was an error when trying to read the non-volatile calibration memory.*
ERROR_NONVOL_WRITE	*There was an error when trying to write to the non-volatile calibration memory.*
ERROR_NONVOL_DEFAULTED	*There was an error when trying to read the default values from the non-volatile calibration memory.*
ERROR_FLASH_BURN	*Error trying to burn data in the FLASH (a common error here would occur when loading the firmware for U1 and issuing the write flash commands for U20, or vice versa).*
ERROR_INTERNAL_SOFTWARE	*The microprocessor encountered an internal software error.*
ERROR_QUEUE_OVFL	*There were more then five errors in the ERROR queue.*

Diagnostic Commands

The following commands in combination with the Self-Test command (see measurement commands section above) help diagnose problems with the board.

0x3000	Read the self-test register.
0x3001	Read the self-test relay register.
0x3002	Read the switch register.
0x3003	Read the trigger inputs.
0x3004	Read the calibrated reference value: 0.945 V.
0x3005	Read the calibrated reference value: -0.945 V.
0x3006	Read the calibrated reference value: 9.45 V.
0x3007	Read the calibrated reference value: -9.45 V.
0x3008	Read the calibrated reference value: 128 Ω.
0x3009	Read the calibrated reference value: 81.92 kΩ.
0x300A	Read self-test measurement values.
0x300B	Turn SYSFAIL ON (for diagnostic purposes only).
0x300C	Turn SYSFAIL OFF (just for diagnostic purposes).
0x300D	Read the calibrated reference value: 0.1177 V.
0x300E	Read the calibrated reference value: -0.1177 V.
0x300F	Unused.
0x3010	Control the HS self-test relays – all OFF.
0x3011	Turn Channel 4 self-test on (Channel 5 OFF).
0x3012	Turn Channel 5 self-test on (Channel 4 OFF).
0x3013	Turn Channel 4 and Channel 5 self-test on (not normal).

FLASH Memory Programming Commands

The following commands can be used to change the content of the FLASH memory. The FLASH memory stores the board's software (executed by the microprocessor) and firmware (what programs the two FPGAs on the board). To prevent accidental writings of the FLASH, a sequence of three commands is necessary to perform a write to it.

To change the microprocessor software, the new file that needs to be programmed into FLASH is uploaded at offset address 0x000000. Then, to change the ***software***, the following three commands must be issued in this particular order. Issuing them in a different order will not produce any results:

0x5501, 0xAA01, 0x5501

After issuing each of the above commands, the user must wait until the command is executed before issuing the next one. When the second 0x5501 command is done, it means the new data has been moved into FLASH memory.

To change the ***firmware*** for the *first* FPGA (U1), the new file that needs to be programmed into FLASH is uploaded at offset address 0x000000. Then, the following three commands must be issued in this particular order. Issuing them in a different order will not produce any results:

0x5502, 0xAA02, 0x5502

After issuing each of the above commands, the user must wait until the command is executed before issuing the next one. When the second 0x5502 command is done, it means the new data has been moved into FLASH memory.

To change the ***firmware*** for the *second* FPGA (U20), the new file that needs to be programmed into FLASH is uploaded at offset address 0x000000. Then the following three commands must be issued in this particular order. Issuing them in a different order will not produce any results:

0x5503, 0xAA03, 0x5503

After issuing each of the above commands, the user must wait until the command is executed before issuing the next one. When the second 0x5503 command is done, it means the new data has been moved into FLASH memory.

To change the ***firmware*** for the *third* FPGA (U37), the new file that needs to be programmed into FLASH is uploaded at offset address 0x000000. Then the following three commands must be issued in this particular order. Issuing them in a different order will not produce any results:

0x5504, 0xAA04, 0x5504

After issuing each of the above commands, the user must wait until the command is executed before issuing the next one. When the second 0x5503 command is done, it means the new data has been moved into FLASH memory.

Changes become effective the next time the module powers up.

WARNING: ANY COMMANDS NOT LISTED HERE ARE RESERVED FOR FACTORY USE AND SHOULD NOT BE USED UNDER ANY CIRCUMSTANCES.

EXAMPLES

Example 1: Setting the Channel 2 and 4 Sample Rate to 123 ms (8.13 kHz)

The sample rate clock for an individual low-speed channel (Channels 0 – 3) is generated by dividing a 0.1 µs (10 MHz) reference clock, generated by an on-board oscillator, by the value present in the Sample Rate register of the respective channel. For the high-speed channels (Channels 4 – 5), the reference clock is 8.333 ns (120 MHz).

123 ms = $123*10^{-3}$ s
For Channel 2, divide 123 ms by the 0.1 µs reference clock:

$$\frac{123\times10^{-3}\,\text{s}}{100\times10^{-9}\,\text{s}} = 1{,}230{,}000$$

For Channel 4, divide 123 ms by the 8.333 ns reference clock:

$$\frac{123\times10^{-3}\,\text{s}}{8.333\times10^{-9}\,\text{s}} \cong 14{,}760{,}590$$

To set the Sample Rate to 123 ms, the reference clock must be divided by 1,230,000 for Channel 2 and 14,760,590 for Channel 4. In hexadecimal format, these values correspond to 0x12C4B0 and E13A8E, respectively. The Sample Rate register for Channel 2 is composed of two 16 bits registers located at offsets 0xC0005C (the MS – Most Significant bits, bits D24 - 16) and 0xC0005E (the LS – Least Significant bits, bits D15 - D0). The Sample Rate register for Channel 4 is similar, starting at offset 0xAC.

Method 1: Make two 16 bits writes.

For low-speed Channel 2:
Write 0x0012 to Base address + 0xC0005C
Write 0xC4B0 to Base address + 0xC0005E

For high-speed Channel 4:
Write 0x00E1 to Base address + 0xC000AC
Write 0x3A8E to Base address + 0xC000AE

Method 2: Make one 32 bits write.

For low-speed channels 0 – 3:
Write 0x0012C4B0 to Base address + 0xC0005C

For high-speed channels 4 – 5:
Write 0x00E13A8E to Base address + 0xC000AE

Example 2: Setting Channel 2 to Acquire 200,000 Samples

The number of samples acquired in Linear mode by a channel is determined by the value programmed in the Sample Points register of the respective channel. In hexadecimal format, 200,000 corresponds to 0x30D40. The Sample Points register for Channel 2 is composed of two 16-bit registers located at offsets 0xC00060 (the MS, bits D19 - D16) and 0xC00062 (the LS, bits D15 - D0).

Method 1: Make two 16 bits writes.

Write 0x0003 @ Base address + 0xC00060

Write 0x0D40 @ Base address + 0xC00062

Method 2: Make one 32 bits write.

Write 0x00030D40 @ Base address + 0xC00060

Example 3: Setting Channel 2 to Pre-acquire 100,000 Samples

The acquisition of samples starts when a trigger point is met or when a trigger is forced by setting the corresponding Force bit. However, samples can be collected before the occurrence of the trigger point. These samples are called Pre-Trigger Points. The number of Pre-Trigger Points to be acquired is determined by the value programmed in the Pre-Trigger Points register of the respective channel.

In hexadecimal format, 100,000 corresponds to 0x186A0. The Pre-Trigger Points register for Channel 2 is composed of two 16-bit registers located at offsets 0xC00064 (the MS, bits D19 - D16) and 0xC00066 (the LS, bits D15 - D0).

Method 1: Make two 16 bits writes.

Write 0x0001 @ Base address + 0xC00064
Write 0x86A0 @ Base address + 0xC00066

Method 2: Make one 32 bits write.

Write 0x000186A0 @ Base address + 0xC00064

The total number of samples acquired is set by the Sample Points, and the number of samples stored after the trigger event is:

(Sample Points) – (Pre-Trigger Points)

Note	Pre-Trigger Points must be less than Sample Points (Pre-Trigger Points < Sample Points).

Example 4: Setting Channel 2 to Delay Acquisition by 1,500,000 Samples

The acquisition of samples starts when a trigger point is met or when a trigger is forced by setting the corresponding Force bit. If the acquisition is to be triggered by a trigger event (signal trigger, external trigger or forced trigger) the first sample is collected either immediately, within one sample clock period from the moment when the trigger occurs or after a Delay Period. The Delay Period is determined by the value programmed in the Trigger Delay register of the respective channel. The Delay Period is based on the Sample Rate value. If the value in the Trigger Delay register is set to zero, the sampling starts immediately after the trigger event. If the Trigger Delay register is set to 100 (0x64), for example, then Sample Zero is taken 100 sample clocks after the trigger event.

In hexadecimal format, 1,500,000 corresponds to 0x16E360. The Trigger Delay register for Channel 2 is composed of two 16-bit registers, located at offsets 0xC00068 (the MS, bits D31 - D16) and 0xC0006A (the LS, bits D15 - D0).

Method 1: Make two 16 bits writes.

Write 0x0016 @ Base address + 0xC00068
Write 0xE360 @ Base address + 0xC0006A

Method 2: Make one 32 bits write.

Write 0x0016E360 @ Base address + 0xC00068

If the Sample Rate register from Example 1 (0x0012C4B0 corresponding to a sample rate of 123 ms) and the Trigger Delay in Example 4 above are set on the SVM2608, the first sample is taken 1,500,000 x 123 ms = 184,500 seconds (51.25 hours!!!) after the trigger event.

Example 5: Setting Channel 2 and 4 Timeout Register to Timeout after 2.5 s

The Timeout function is designed to prevent the SVM2608 from encountering indefinite time periods where the module waits for an external trigger event. The clock for the timeout counter is generated by dividing the reference clock (0.1 µs (10 MHz) for the low-speed Channels 0 – 3 and 8.333 ns (120 MHz) for high-speed channels 4 and 5), generated by an on-board oscillator, by a preset value determined by bits 15 - 13 of the Timeout register of the respective channel. This is called the Timeout Base Clock (or TOSEL). Bits 12 - 0 in the TIMEOUT register load a counter (Timeout Counter) that determines the number of Timeout Base Clocks after which the timeout flag is set.

To determine the minimum Timeout Base Clock for ***Channel 2***, divide the desired *Timeout* value by the maximum value allowed in the *Timeout Counter register* and round it up to the next highest value available:

TIMEOUT BASE CLOCK $\geq$ TIMEOUT VALUE / $(2^{13} - 1)$ = 2.5 s / 8191 $\approx$ 0.305 ms

The nearest Timeout Base Clock available, which is greater than or equal to 0.305 ms, is 1 ms. Set the TOSEL bits in the Timeout register to '010'.

To determine the value for the Timeout Counter, divide the desired *Timeout* value by the *Timeout Base Clock*:

TIMEOUT COUNTER = TIMEOUT / TIMEOUT BASE CLOCK
= 2.5 s / 1 ms
= 2.5 s / 10^{-3} s
= 2,500

The value that needs to be programmed in the Timeout Counter is 2,500 = 0x9C4
The value which needs to be programmed into the Timeout register is determined by multiplying the *Timeout Base* by 2^{13} and adding it to the *Timeout Counter* value:

TIMEOUT REGISTER = TIMEOUT BASE * 2^{13} + TIMEOUT COUNTER
= 2 * 8192 + 2500
= 18884
= 0x49C4

To program the Timeout value for Channel 2, write 0x49C4 at address 0xC00044.

The Timeout Counter starts counting down from the value it is loaded with after the channel is Armed, all the Pre-Trigger samples (if any were specified) are acquired and the Delay Points (if any were specified) are acquired. It stops counting as soon as either a trigger event occurs or when it reaches zero. Terminating (aborting) the acquisition by resetting the ARM bit also stops the Timeout counter.

If the Timeout Counter reaches zero, the Timeout Occurred flag is asserted.

APPENDIX A

APPENDIX A

DATA SWAPPING EXAMPLE

An example is provided below detailing how data might be swapped in to get a REAL number when the data is read “swapped”. In order to make the code easier to understand, only the portion that shuffles the data is presented here. The VME functions that actually perform the reading of the data are the responsibility of the user.

```
byte  tempData[16];
int   regRdata[16];
doublerealData;

union
{     char  data[8];
      doublevalue;
}dblData;


      // read results
      data = VmeRead16(channel, VmeAddress+0x28, &Api_Result);
      printf("MSB1:  %02x\n", data);
      regRdata[0] = data;
      data = VmeRead16(channel, VmeAddress+0x2a, &Api_Result);
      printf("MSB0:  %02x\n", data);
      regRdata[1] = data;
      data = VmeRead16(channel, VmeAddress+0x2c, &Api_Result);
      printf("LSB1:  %02x\n", data);
      regRdata[2] = data;
      data = VmeRead16(channel, VmeAddress+0x2e, &Api_Result);
      printf("LSB0:  %02x\n", data);
      regRdata[3] = data;

      //    now shuffle the bytes
      dblData.data[0] = (byte)((regRdata[3] & 0xff00) >> 8);
      dblData.data[1] = (byte)((regRdata[3] & 0x00ff) >> 0);
      dblData.data[2] = (byte)((regRdata[2] & 0xff00) >> 8);
      dblData.data[3] = (byte)((regRdata[2] & 0x00ff) >> 0);
      dblData.data[4] = (byte)((regRdata[1] & 0xff00) >> 8);
      dblData.data[5] = (byte)((regRdata[1] & 0x00ff) >> 0);
      dblData.data[6] = (byte)((regRdata[0] & 0xff00) >> 8);
      dblData.data[7] = (byte)((regRdata[0] & 0x00ff) >> 0);
      realData = dblData.value;
```

INDEX