VM4018

PRECISION ANALOG COMPARATOR / INTERRUPTER

USER'S MANUAL

82-0064-000
Rev. August 25, 2003

VXI Technology, Inc.

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

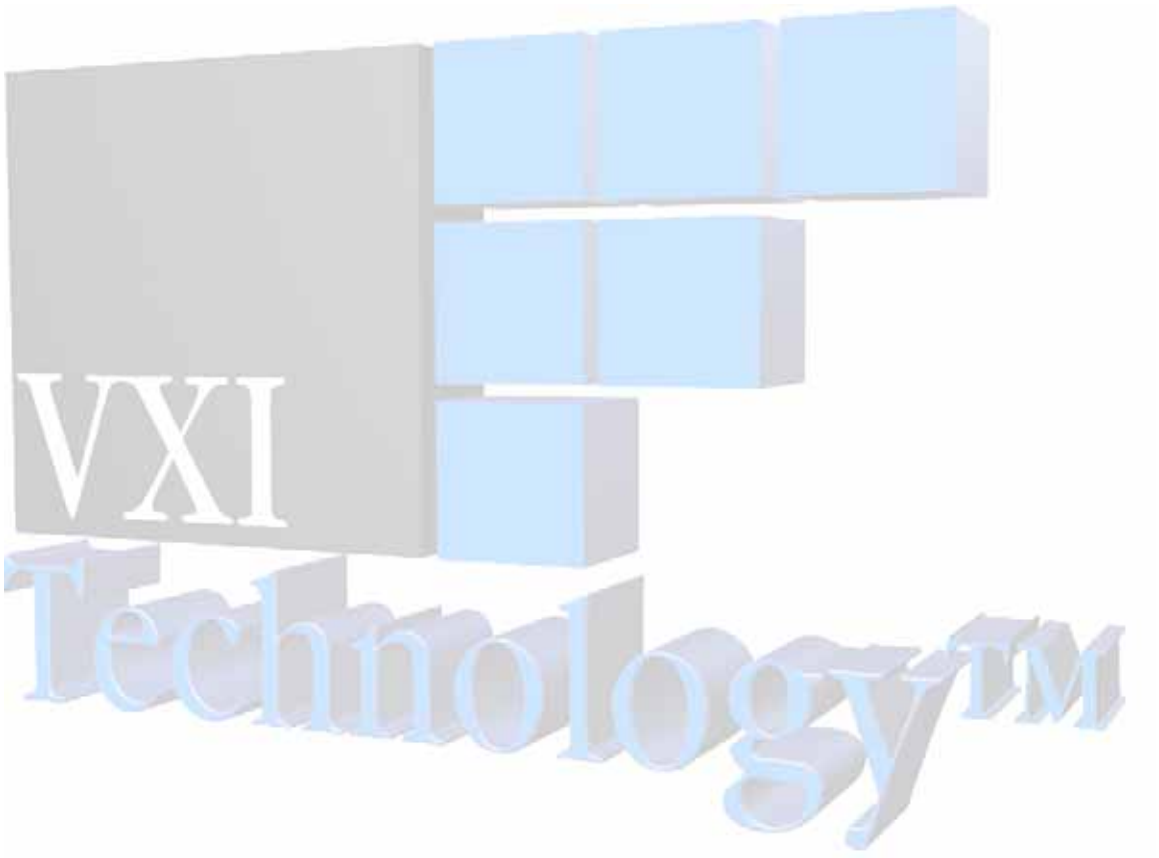

VXI
Technology™

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 three years 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
PRODUCT NAME	Precision Analog Comparator/Interrupter
MODEL NUMBER(S)	VM4018
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:*

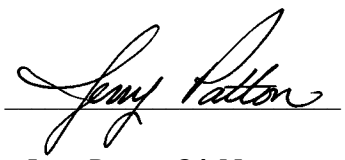
SAFETY	EN61010 (2001)
EMC	EN61326 (1997 w/A1:98) Class A CISPR 22 (1997) Class A VCCI (April 2000) Class A ICES-003 Class A (ANSI C63.4 1992) AS/NZS 3548 (w/A1 & A2:97) Class A FCC Part 15 Subpart B Class A EN 61010-1:2001

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.

August 2003

CE



Jerry Patton, 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

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

VXI Technology, Inc.
7525 Granger Road, Unit 7
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

See http://www.vxitech.com for worldwide support sites.

SECTION 1

INTRODUCTION

OVERVIEW

The VM4018 is a high-performance analog comparator module with sixteen input channels per VMIP daughter module. Each input channel can be programmed for the following ranges: ±1.0 V, ±2.5 V, ±10.0 V, ±25.0 V, ±100.0 V and ±250.0 V. Each input is compared against a reference voltage derived from an independent 12-bit DAC (*digital-to-analog converter*). The VM4018 is an ideal device for "go/no-go" testing, where a device fails if the voltage outputs exceed a threshold or window, or even in control applications where a device or test needs to be shut down if a voltage level is exceeded. Using an analog comparator/interrupter in certain applications, as opposed to a traditional DMM/switch approach, considerably improves the overall throughput of the test while substantially reducing cost.

Each input signal is digitally debounced for a programmed time ranging from approximately 12.8 μs to 0.42 s. This prevents input signal noise from causing undesired interrupts. After debounce, the signal may be inverted via serial word command to select the input transition edge of interest (rising or falling edge) and masked to prevent unused channels from causing interrupts.

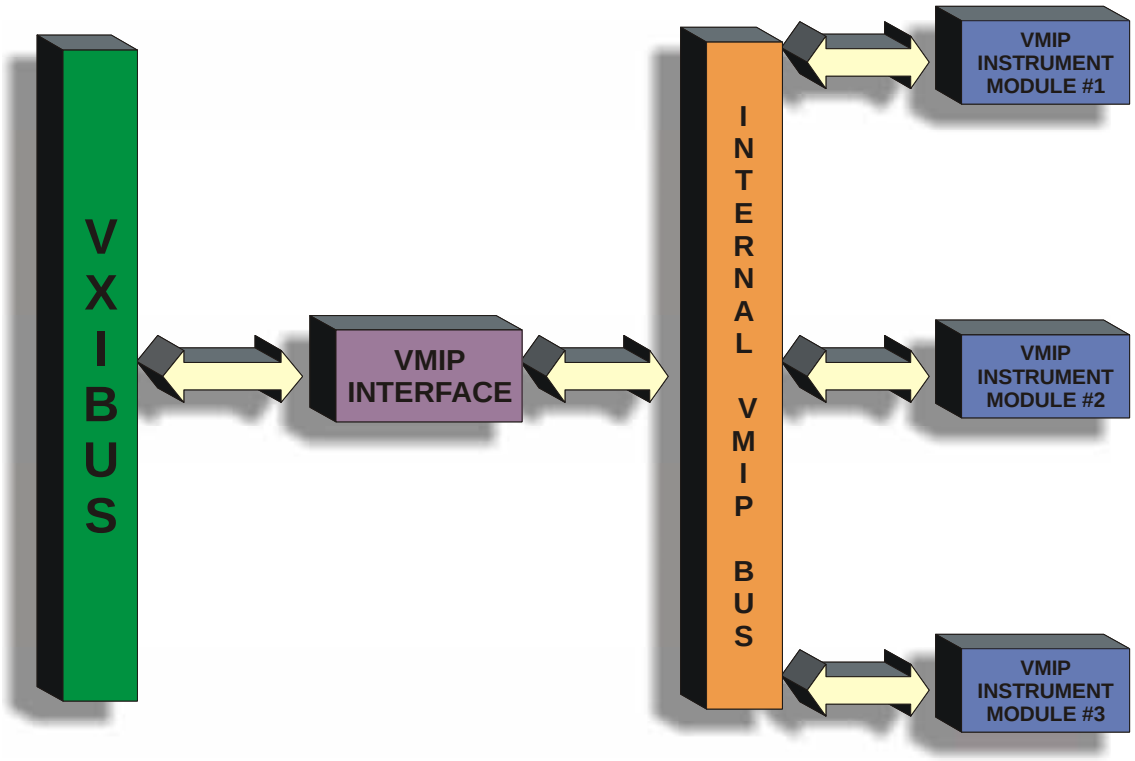
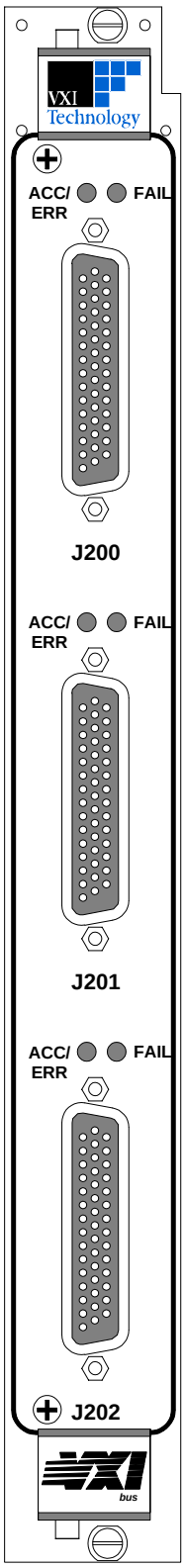


FIGURE 1-1 VMIP™ PLATFORM



Regardless of whether the VM4018 is configured with other VM4018 modules or with other VMIP modules, each group of sixteen channels is treated as an independent instrument in the VXIbus chassis and, as such, each group has its own FAIL and ACCESS/ERROR LEDs.

The FAIL LED is a Power/Fault indicator. When a normal power up condition exists, the FAIL LED will illuminate green. When a fault condition occurs during power up, the FAIL LED will illuminate red. The ACC/ERR LED indicates communication status. When a successful access (ACC) occurs, the LED will blink green during data transfer and command/query operations. In the event of an unrecognized command, or other data related error (ERR), the LED will illuminate red. If there is no command/query activity and there are no errors, the ACC/ERR LED will be extinguished. The normal state of the LEDs on a properly functioning idle instrument is for the FAIL LED to be green and the ACC/ERR LED to be unlit.

DESCRIPTION

The VM4018 is a high-performance analog comparator module with sixteen input channels per VMIP daughter module. Each input channel can be programmed for the following ranges: ±1.0 V, ±2.5 V, ±10.0 V, ±25.0 V, ±100.0 V and ±250.0 V. Each input is compared against a reference voltage derived from an independent 12-bit DAC.

Each input signal is digitally debounced for a programmed time ranging from approximately 12.8 μs to 0.42 s. This prevents input signal noise from causing undesired interrupts. After debounce, the signal may be inverted via serial word command to select the input transition edge of interest (rising or falling edge) and masked to prevent unused channels from causing interrupts.

All the masked inputs are OR'd together to produce a single interrupt signal. This interrupt signal is used to generate a VXIbus interrupt, a TTL trigger bus event or to generate front panel interrupt outputs. Special logic will latch the first input to cross its threshold, into the First Latched Register. This records the originating input. The First Latched Register can be cleared by querying the Latched Register contents using the word serial command FETCh:LATched?, or by querying the data via Pseudo Register Access with the INHOUSE:CLEAR_LATCH set to *1* or *ON*. (See Section 4 for more details on the aforementioned commands.)

The state of each channel's debounced input and the inverted and masked status may be read directly in the user-defined area of the VXIbus registers, as can the First Latched register. This information may also be retrieved using the message-based word serial interface.

The VM4018 may be configured to utilize the TTLTRIG lines as outputs and will toggle when an interrupt event occurs in the modules. Setting a bit to 1 enables the trigger line, setting a bit to 0 disables the corresponding line. This can be configured by the user by using the OUTPut:TTLTrg SCPI command.

The block diagram of Figure 1-3 shows the overall functionality of the VM4018 Analog Comparator instrument.

FIGURE 1-2 FRONT PANEL LAYOUT

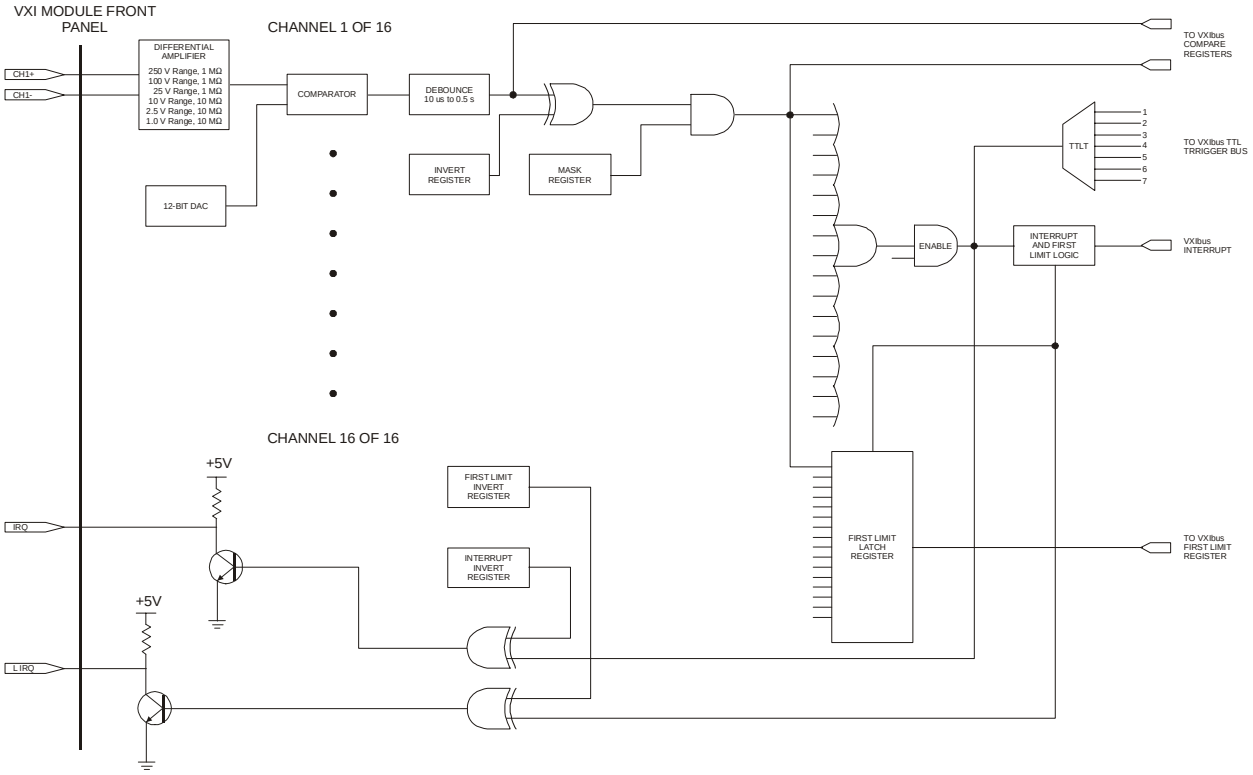


FIGURE 1-3 VM4018 BLOCK DIAGRAM

TABLE 1-1 VM4018 GENERAL SPECIFICATIONS

GENERAL SPECIFICATIONS	
CHANNELS	
	16
INPUT RANGE	
	±1.0 V, ±2.5 V, ±10.0 V, ±25.0V, ±100 V, ±250 V Resolution: 0.05% of range within 5°C range
INPUT THRESHOLD	
	0.05% of range within 5°C range
WARM-UP TIME	
	30 minutes
INPUT TYPE	
	Differential, may be configured for single-ended by grounding the negative input
INPUT IMPEDANCE	
≤ 10 V	≥ 1 MΩ
≥ 25 V	1 MΩ
INPUT POLARITY	
	Rising or Falling Edge
DEBOUNCE TIME	
	12.8 μs to 0.41943045 s, 6.4 μs resolution
IRQ AND LATCHED IRQ OUTPUT	
	Open Collector Driver, 200 mA max. sink Internally pulled up to +5 V with 10 k resistor
ACCURACY	
	±0.05% of Range + Gain/Offset Error
GAIN/OFFSET ERROR	
	±1 count of 12-bit DAC
VXI INTERFACE	
	Message-based word serial interface Direct register access in the user defined area of the VXIbus register map
LOGICAL ADDRESSING	
	Static or Dynamic Configuration
RAW DATA REGISTER	
	Logical Address + 20H
MASKED DATA REGISTER	
	Logical Address + 28H
FIRST LATCHED REGISTER	
	Logical Address + 30H
POWER REQUIREMENTS	
	+5 V @ 1.43 A -5.2 V @ 0.17 A +24V @ 0.05 A -24V @ 0.05 A
COOLING REQUIREMENTS	
	See Power Cooling Table

CONNECTOR PIN OUTS / PIN LOCATIONS

The connector used in the VM4018 is a commonly available 44-pin high density DSUB receptacle connector. A mating solder cup pin connector from AMP is included; crimp type connectors are available from a variety of sources (see the following page for two such sources).

TABLE 1-2 ANALOG COMPARATOR PIN OUTS

SIGNAL	PIN NUMBER	SIGNAL	PIN NUMBER
CHANNEL 1 +	1	GROUND	23
CHANNEL 1 -	2	CHANNEL 11 +	24
GROUND	3	CHANNEL 11 -	25
CHANNEL 4 +	4	CHANNEL 14 +	26
CHANNEL 4 -	5	CHANNEL 14 -	27
CHANNEL 7 +	6	GROUND	28
CHANNEL 7 -	7	CHANNEL 16 +	29
GROUND	8	CHANNEL 16 -	30
CHANNEL 10 +	9	CHANNEL 3 +	31
CHANNEL 10 -	10	CHANNEL 3 -	32
CHANNEL 13 +	11	GROUND	33
CHANNEL 13 -	12	CHANNEL 6 +	34
GROUND	13	CHANNEL 6 -	35
IRQ OUTPUT	14	CHANNEL 9 +	36
GROUND	15	CHANNEL 9 -	37
CHANNEL 2 +	16	GROUND	38
CHANNEL 2 -	17	CHANNEL 12 +	39
GROUND	18	CHANNEL 12 -	40
CHANNEL 5 +	19	CHANNEL 15 +	41
CHANNEL 5 -	20	CHANNEL 15 -	42
CHANNEL 8 +	21	GROUND	43
CHANNEL 8 -	22	LATCHED IRQ OUT	44

The pin locations are shown in Figure 2-2.

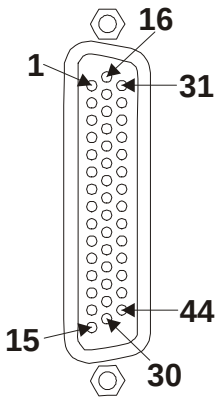


FIGURE 2-2 VM4018 PIN LOCATIONS

MATING CONNECTORS

The mating connector for the VM4018 is available from the following companies:

ITT Cannon (www.ittcannon.com)

P/N ZDBA44P	Connector
P/N ZD110238-1009	Connector Pin
P/N 995-2000-022	Crimp Tool
P/N 980-0004-804	Insertion/Extraction Tool

Positronic Industries, Inc. (www.positronic.com)

P/N ODD44M1FY0C	Connector
P/N 9507	Crimp Tool
P/N 9502-4	Contact Positioner
P/N M81969/1-04	Insertion/Extraction Tool

SECTION 2

PREPARATION FOR USE

INSTALLATION

When the VM4018 is unpacked from its shipping carton, the contents should include the following items:

(1) VM4018 Precision Analog Comparator/Interrupter VXIbus module
(1) VM4018 Precision Analog Comparator/Interrupter Module User's Manual (this manual)

All components should be immediately inspected for damage upon receipt of the unit.

The chassis should be checked to ensure that it is capable of providing adequate power and cooling for the VM4018. Once the chassis is found adequate, the VM4018's logical address and the backplane jumpers of the chassis should be configured prior to the VM4018's installation. After the chassis is assessed to be in good condition, it may be installed into an appropriate C-size or D-size VXIbus chassis in any slot other than slot zero.

INSTRUMENT SENSITIVITY

In comparison to the VM4016, the VM4018 is considerably more sensitive to input noise and other environmental factors. Special care needs to be taken when setting the debounce time, as a signal with noise near the debounce threshold may cause the instrument to send interrupts indefinitely, making the VM4018 unresponsive. To attain the precision specified, the instrument needs to remain within ±5°C of its calibration temperature. The VM4018 is calibrated at a nominal factory temperature of 25°C. The instrument should be allowed to warm up for 30 minutes before use.

CALCULATING SYSTEM POWER AND COOLING REQUIREMENTS

The power and cooling requirements of the VM4018 are given in the Specifications section of Chapter 1 in this manual. It is imperative that the chassis provide adequate power and cooling for this module. Referring to the chassis User's 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 airflow at the specified backpressure.

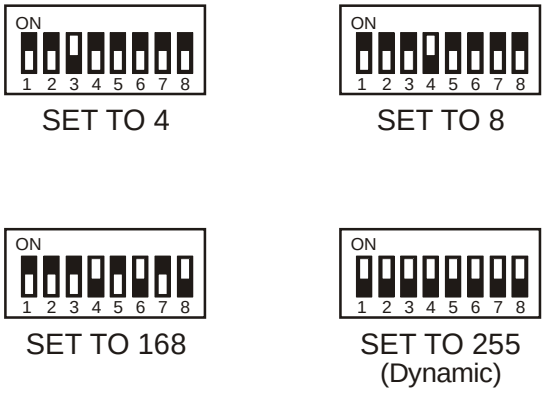
It should be noted that if the chassis cannot provide adequate power to the module, the instrument might not perform to specification or possibly not operate at all. In addition, if adequate cooling is not provided, the reliability of the instrument will be jeopardized and permanent damage may occur.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

SETTING THE CHASSIS BACKPLANE JUMPERS

Please refer to the chassis User's Manual for further details on setting the backplane jumpers.

SETTING THE LOGICAL ADDRESS

The logical address of the VM4018 is set by a single 8-position DIP switch located near the module's backplane connectors (this is the only switch on the module). The switch is labeled with positions 1 through 8 and with an ON position. A switch pushed toward the ON legend will signify a logic 1; switches pushed away from the ON legend will signify a logic 0. The switch located at position 1 is the least significant bit while the switch located at position 8 is the most significant bit. See Figure 2-1 for examples of setting the logical address switch.



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

FIGURE 2-1 LOGICAL ADDRESS SWITCH SETTING EXAMPLES

The VMIP may contain three separate instruments and will allocate logical addresses as required by the VXIbus specification (revisions 1.3 and 1.4). The logical address of the instrument is set on the VMIP carrier. The VMIP logical addresses must be set to an even multiple of 4 *unless dynamic addressing is used*. Switch positions 1 and 2 must always be set to the OFF position. Therefore, only addresses of 4, 8, 12, 16, ...252 are allowed. The address switch should be set for one of these legal addresses and the address for the second instrument (the instrument in the center position) will automatically be set to the switch set address plus one; while the third instrument (the instrument in the lowest position) will automatically be set to the switch set address plus two. If dynamic address configuration is desired, the address switch should be set for a value of 255 (All switches set to ON). Upon power-up, the slot 0 resource manager will assign the first available logical addresses to each instrument in the VMIP module.

If dynamic address configuration is desired, the address switch should be set for a value of 255. Upon power-up, the slot 0 resource manager will assign logical addresses to each instrument in the VMIP module.

SECTION 3

PROGRAMMING

INTRODUCTION

The VM4018 is a VXIbus message-based device whose command set is compliant with the SCPI programming language. All module commands are sent over the VXIbus backplane to the module. Commands may be in upper, lower or mixed case. All numbers are sent in ASCII decimal unless otherwise noted.

The SCPI programming language is a tree-structured language based on IEEE STD 488.2 Specifications. It utilizes the IEEE STD 488.2 Standard commands and the device dependent commands are structured to allow multiple branches off the same trunk to be used without repeating the trunk. To use this facility, terminate each branch with a semicolon. For example, **OFFSet:GAIN** and **OFFSet:OFFSet** are both branches off the **CALibration:** trunk and can be combined as follows:

```
CALibration:OFFSet:GAIN <value>;OFFSet <value>
```

The above command is the same as the these two commands:

```
CALibration:OFFSet:GAIN <value>
CALibration:OFFSet:OFFSet <value>
```

See the SCPI Manual, Volume 1: Syntax & Style, Section 6, for more information.

The SCPI commands in this section are listed in upper and lower case. Character case is used to indicate different forms of the same command. Keywords can have both a short form and a long form (some commands only have one form). The short form uses just the keyword characters in uppercase. The long form uses the keyword characters in uppercase plus the keyword characters in lowercase. Either form is acceptable. Note that there are no intermediate forms. All characters of the short form or all characters of the long form must be used. Short forms and long forms may be freely intermixed. The actual commands sent can be in upper case, lower case or mixed case (case is only used to distinguish short and long form for the user). As an example, these commands are all correct and all have the same effect:

```
CALibration:OFFSet:GAIN <value>
calibration:offset:gain <value>
CALIBRATION:OFFSET:GAIN <value>
CAL:OFFSet:GAIN <value>
CAL:OFFS:GAIN <value>
cal:offs:gain <value>
```

The following command is **not** correct because it uses part of the long form of **CALibration**, but not all the characters of the long form:

```
calib:offs:gain <value>
```

Incorrect syntax - extra "ib"- only cal or calibration is correct

All of the SCPI commands also have a query form unless otherwise noted. Query forms contain a question mark (?). The query form allows the system to ask what the current setting of a parameter is. The query form of the command generally replaces the parameter with a question mark (?). Query responses do not include the command header. This means only the parameter is returned: no part of the command or "question" is returned.

NOTATION

Keywords or parameters enclosed in square brackets ([ ]) are optional. If the optional part is a keyword, the keyword can be included or left out. Omitting an optional parameter will cause its default to be used.

Parameters are enclosed by angle brackets (< >). Braces ({ }), or curly brackets, are used to enclose one or more parameters that may be included zero or more times. A vertical bar (|), read as "or", is used to separate parameter alternatives.

EXAMPLES OF SCPI COMMANDS

FETCH:CONDITIONED?

The *FETCh:CONDitioned?* query returns the 16-bit value that represents the current conditioned (masked and inverted) state of the inputs. It is important to note that this information is also available at the VXIbus register level at offset 0x28.

FETCh:CONDitioned? ***No query parameters.***

EXAMPLES

Command/Query	Response	Description/Explanation
`FETCh:CONDitioned?`	`0`	*Returns the state of the conditioned (masked and inverted) inputs*

FETCH:LATCHED?

The *FETCh:LATChed?* query returns a 16-bit value that reports the active signals in the First Latched register. The First Latched register records the active signals when the first new input channel crossed its threshold. It is important to note that the above information is also available at the VXIbus register level at offset 0x30.

FETch:LATChed? ***No query parameters***

EXAMPLES

Command/Query	Response	Description/Explanation
`FETCh:LATChed?`	`1`	*Returns the active signals in the First Latched register (Channel 1)*
`FETC:LATC?`	`3`	*Returns the active signals in the first Latched Registers (Channels 1 and 2)*

FETCH:RAW?

The *FETCh:RAW?* query returns the 16-bit value that represents the current unconditioned (unmasked and non-inverted) state of the inputs. It is important to note that the above information is also available at the register level at offset 0x20.

FETCH:RAW? ***No query parameters***

EXAMPLES

Command/Query	Response	Description/Explanation
`FETCh:RAW?`	`1`	*Returns the state of the unconditioned (unmasked and non-inverted) inputs. (Channel 1)*
`FETC:RAW?`	`65535`	*All 16 channels crossed the program threshold.*

INHOUSE:CLEAR_LATCH

The *INHOUSE:CLEAR_LATCH* command determines whether the first latched information will be cleared when the information is read by the FETch:LATChed? word serial command or if PSEUDO is set and a register read of the first latched information occurs. The information will not be cleared if a hardware register read is used. When the first latched information is cleared, all the following reads will return a value of 0 until a new first latched event occurs. It is important to note that all letters in the command must be provided, as there is no short form for this command.

INHOUSE:CLEAR_LATCH <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INHOUSE:CLEAR_LATCH 1`		*Clears the first latched information on a read*
`INHOUSE:CLEAR_LATCH?`	`1`	*Returns 1 stating that the first latched information will be cleared on a read*
`FETC:LATC?`	`1`	*Reading the first latched information. This also clears the latched information*
`FETC:LATC?`	`0`	*Reading the first latched information returns a value of 0, once the clearing of the latch information was enabled (assuming no further latching occurred.)*

INHOUSE:PSEUDO

The *INHOUSE:PSEUDO* command controls the use of the register interface. When PSEUDO is set to true, it specifies that the pseudo register interface should be used. When PSEUDO is false, this specifies that the hardware register interface should be used. The value set takes effect the next time the unit powers up (i.e., it does not take effect immediately). The pseudo register interface allows the use of REG_ENABLE and CLEAR_LATCH capability from the registers. Although the hardware register interface is much faster than the pseudo register interface, it lacks the above two features. It is important to note that when the module is shipped from the factory, pseudo is set to 1. It is also important to note that all letters of the command must be provided, as there is no short form for this command.

INHOUSE:PSEUDO <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INHOUSE:PSEUDO 1`		*Sets the pseudo register interface ON (The unit must be power cycled before the change will take effect.)*
`INHOUSE:PSEUO?`	`1`	*Returns 1 that states that the register interface is set to pseudo*

INHOUSE:REGINT

The *INHOUSE:REGINT* command controls the type of module's response to an interrupt acknowledge cycle [ack cycle]. When REGINT is false, the module uses reqt | reqf (request true | request false), provided the latched interrupt bit is set in the SRE.

It is important to note that all the letters of the command must be provided, as there is no short form for this command.

INHOUSE:REGINT <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INHOUSE:REGINT 1`		*Sets the type of module interrupt response to one backplane interrupts for every first latched event.*
`INHOUSE:REGINT?`	`1`	*Sets the type of module interrupt response as two backplane interrupts of reach latched event*
`INHOUSE:REGINT 0`		*Sets the type of module interrupts response as two backplane interrupts for each latched event.*
`INHOUSE:REGINT?`	`0`	*Returns the type of module interrupt response as 0*

INHOUSE:REG_ENABLE

The *INHOUSE:REG_ENABLE* command controls the masking for REGINT. REG_ENABLE 0 means that backplane interrupts cannot be generated. If the REG_ENABLE is 1, then backplane interrupts can be generated. If PSEUDO is set, then a write to the register at offset 0x38 also controls the masking. Enable or disable capabilities are provided in the pseudo register interface to allow a complete register interface. It is important to note that all letters of the command must be provided, as there is no short form for this command.

INHOUSE:REG_ENABLE <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INHOUSE:REG_ENABLE 1`		*Enables the REGINT type interrupt generation*
`INHOUSE:REG_ENABLE?`	`1`	*Returns 1 to state that backplane interrupting is currently enabled*
`INHOUSE:REG_ENABLE 0`		*Disabling REGINT interrupt generation*
`INHOUSE:REG_ENABLE?`	`0`	*Returns 0 to state that backplane interrupting is currently disabled*

INPUT:DEBOUNCE

The *INPut:DEBounce* command sets the time period for the digital debounce circuitry. This command affects all sixteen channels of the instrument. It is important to note that the debounce resolution is 6.4 µs.

INPut:DEBounce <value> ***Where <value> ranges from 12.8 µs (0.0000128 s) to 0.4194304 s.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:DEBounce 12.8e-6`		*Sets the input debounce time for all channels to 0.0000128 s. This will not allow a signal to generate an interrupt unless the input signal crosses the signal for more than 12.8e-6 seconds*
`INP:DEB 0.4`		*Sets input debounce time to 0.4 seconds*
`INP:DEB?`	`0.4`	*Returns the input debounce time as 0.4 seconds*

INPUT:MASK

The *INPut:MASK* command enables or disables input channels from generating interrupts or recording data in the conditional register. If a channel is programmed to be ON (or 1), then it is enabled to generate interrupts. If a channel is programmed to be OFF (or 0), then it cannot generate VXIbus interrupts.

INPut:MASK <state>,<channel_list>

Where <state> is 0 | OFF | 1 | ON.

Where <channel_list> is standard channel list format supporting Channels 1 through 16.

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:MASK ON,(@1:8)`		*Enables Channel 1 though 8 to generate interrupts*
`INPut:MASK? 3`	`1`	*Reports that Channel 3 is enabled for voltage comparison*
`INP:MASK? 9`	`0`	*Reports that Channel 9 is not enabled for voltage comparison*

INPUT:MASK:INTERRUPT

The *INPut:MASK:INTerrupt* command enables or disables interrupt generation when changing MASK values. When set to 0 (the *RST state), interrupts are temporarily disabled whenever MASK values are changed. When set to 1, interrupts are generated even as MASK values are changed.

INPut:MASK:INTerrupt <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:MASK:INTerrupt 0`		*Disables interrupt generation*
`INP:MASK:INT`	`0`	*Reports that interrupt generation is disabled*

INPUT:OFFSET

The *INPut:OFFSet* command sets the input threshold for a channel or group of channels, over which the input signal must cross to cause an interrupt event. This command sets the value in the 12-bit DAC to which the input signal is compared. It is important to note that the actual input offset value is affected by the *INPut:RANGe* command, as the response has been normalized to ±10 V range. The actual input offset for the allowable ranges are as follows:

Range	Entered Threshold	Actual Threshold
±1.0	x	0.1x
±2.5	x	0.25x
±10.0	x	1.0x
±25.0	x	2.5x
±100.0	x	10.0x
±250.0	x	25.0x

INPut:OFFSet <voltage_level>,<channel_list>

Where <voltage_level> ranges from -10.00 V to +10.00 V. For the ±10 V range.

Where <channel_list> is the standard channel list format supporting Channels 1 through 16.

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:RANGe 100,(@5:10)`		*Sets the input range for Channels 5 through 10 to ±100 V*
`INPut:OFFSet -5.0,(@5:10)`		*Sets the input offset for Channels 5 through 10 to -50 V*
`INP:OFFS? 9`	`-5.000`	*Returns the normalized input offset -50 V for Channel 9*

INPUT:POLARITY

The *INPut:POLarity* command selects the input polarity for one or more channels. When a channel is programmed for normal polarity, an interrupt will be generated when the input voltage is greater than the programmed input offset for the channel. The invert polarity will cause an interrupt when the input voltage is less than the programmed input offset for the channel.

INPut:POLarity <polarity>,<channel_list>

Where <polarity> is either NORMal or INVerted

Where <channel_list> is the standard channel list format supporting Channels 1 through 16.

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:POLarity NORM,(@3:5)`		*Sets the input polarity for Channels 3 through 9 to NORMal. This will generate an interrupt when the input signal on Channels 3, 4 or 5 is greater than the input offset.*
`INP:POL? 5`	`NORM`	*Returns the input polarity for Channel 5 as NORMal*
`INP:POL INV,(@6)`		*Sets input Polarity for Channel 6 to INVerted*

INPUT:RANGE

The *INPut:RANGe* command selects the input range of one or more channels. The input range may be either set for ±1.0 V, ±2.5 V, ±10.0 V, ±25.0 V, ±100 V or ±250 V.

INPut:RANGe <range>,<channel_list>

***Where <range> is can be* ±1.0, ±2.5 V, ±10.0, ±25.0 V, ±100.0 *or* ±250.0 V.**

Where <channel_list> is the standard channel list format supporting Channels 1 through 16.

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:RANGe 100,(@1:16)`		*Sets the input range for Channels 1 through 16 to 100 V*
`INP:RANG 10,(@4:6)`		*Sets the input range for Channels 5 through 6 to 10 V*
`INP:RANG? 5`	`10`	*Returns the input range for Channel 5 as 10 V*

OUTPUT:POLARITY:EXTERNAL:INTERRUPT

The *OUTPut:POLarity:EXTernal:INTerrupt* command sets the polarity of the front panel interrupt output. When the polarity is set to normal, the output will be low when there is an interrupt event. When the polarity is set to invert, the output will be high when there is an interrupt event.

OUTPut:POLarity:EXTernal:INTerrupt <polarity> ***Where <polarity> is either NORMal or INVerted.***

EXAMPLES

Command/Query	Response	Description/Explanation
`OUTP:POL:EXT:INT NORM`		*Sets the external interrupt output polarity to a low pulse (NORMal) when an interrupt occurs*
`OUTPut:POLarity:EXTernal:INTerrupt?`	`NORM`	*Returns the external interrupt output polarity as NORMal*
`OUTP:POL:EXT:INT INV`		*Sets the external interrupt output polarity to INVerted*
`OUTP:POL:EXT:INT?`	`INV`	*Returns the polarity of the external interrupt output as INVerted*

OUTPUT:POLARITY:EXTERNAL:LATCHED

The *OUTPut:POLarity:EXTernal:LATCHed* command sets the polarity of the front panel latched interrupt output. When the polarity is set to normal, the output will be low when there is an interrupt event. When set to invert, the output will be high when there is an interrupt event.

OUTPut:POLarity:EXTernal:LATCHed <polarity> ***Where <polarity> is either NORMal or INVerted.***

EXAMPLES

Command/Query	Response	Description/Explanation
`OUTPut:POLarity:EXTernal:LATCHed NORM`		*Sets the external latched output polarity to low when an interrupt occurs*
`OUTP:POL:EXT:LATC?`	`NORM`	*Returns the external latched output polarity as NORMal*
`OUTP:POL:EXT:LATC INV`		*Sets the external latched output polarity to high when an interrupt occurs*
`OUTP:POL:EXT:LATC?`	`INV`	*Returns the external latched output polarity as INVerted*

OUTPUT:TTLTRG

The *OUTPut:TTLTrg* command selects the TTL trigger line that will be activated when the threshold is crossed.

OUTPut:TTLTrg <line> ***Where <line> is a numeric value between and including 0 - 7.***

EXAMPLES

Command/Query	Response	Description/Explanation
`OUTP:TTLT3`		*(Selects TTLT3 trigger line to be activated when the threshold is crossed)*
`OUTP:TTLT?`	`3`	*(Indicates that TTLT3 is selected to be activated when the threshold is crossed)*

OUTPUT:TTLTrg:STATE

The *OUTPut:TTLTrg:STATe* command enables/disables the TTL trigger line output.

OUTPut:TTLTreg:STATe <boolean> ***Where <boolean> is either 0 | 1 | OFF |ON.***

EXAMPLES

Command/Query	Response	Description/Explanation
`OUTP:TTLT:STAT 1`		*(Enables the use of TTL trigger lines as outputs)*
`OUTP:TTLT:STAT?`	`1`	*(Indicates that the TTL trigger lines are enabled as outputs)*

SENSe:CONDition:LEVel

This command sets the threshold in volts. Unlike the INPut:RANGe command, the SENSe:CONDition:LEVel command is not normalized to the ±10 V range and the value entered for the voltage is the actual voltage that will be used as the threshold.

SENSe:CONDition:LEVel <voltage_level>,<channel_list>

Where <volgate_level> is ±1.0 | ±2.5 | ±10.0 | ±25.0 | ±100 | ±250.

Where <channel_list> is the standard channel list format supporting Channels 1 through 16.

EXAMPLES

Command/Query	Response	Description/Explanation
`INPut:RANGe 100,(@1:16)`		*Sets the input range for Channels 1 through 16 to 100 V*
`SENS:COND:LEV 25,(@4:6)`		*Sets the threshold for Channels 4 through 6 to 25 V*
`INP:RANG? 5`	`25`	*Returns the threshold for Channel 5*

APPLICATION EXAMPLES

This section contains examples of using SCPI command strings for programming the VM4018 module. The code is functional and will contain a brief description about the operation.

Example 1
In this example, the VM4018 sets the output interrupt polarity on the front panel and the debounce time for the digital debounce circuitry.

OUTPut:POLarity:EXTernal:INTerrupt NORM	*Sets the external interrupt output polarity to high pulse when an interrupt occurs.*
INPut:DEBounce 25e-6	*Sets input debounce time for all channels to 25 µs . This will not allow a signal to generate an interrupt unless the channel is active for greater than 25 µs.*

Example 2
In this example, the VM4018 enables or disables the specified channels for interrupt generation along with setting the offset, polarity and voltage range. It returns the value of the First Latched register that records the first input channel to cross its threshold and queries the current state of inputs.

`INP:MASK 1,(@1,2)`	*Enables Channels 1 and 2 for interrupt generation.*
`INP:MASK 0,(@3:16)`	*Disables Channels 3 through 16 from generating an interrupt.*
`INP:RANG 10,(@1,2)`	*Selects ±10 V as the input range for Channel 1 and 2.*
`INP:OFFS +5.25,(@1,2)`	*Selects +5.25 V as the offset voltage for Channels 1 and 2.*
`INP:POL NORM,(@1,2)`	*Selects both Channel 1 and 2 to generate an interrupt when Channels 1 and 2 are greater than the offset voltage.*
`FETC:LATC?`	*Returns the active signal in the First Latched Register.*
`FETC:RAW?` `65535`	*Returns the State of unconditioned (unmasked and non-inverted) inputs.*
`FETC:COND?` `3`	*Returns the state of masked and inverted inputs.*

SINGLE CHANNEL OPERATION

This example is for controlling a device that can tolerate a maximum input voltage level at +35 VDC for a maximum time of 250 ms before damage will occur. The input power to this device is provided from a remote source that can be disabled. A low signal applied to the power source remote inhibit will disable its output. The controller will then be notified that an out-of-tolerance condition has occurred and the device was shut down.

The following code is for monitoring a single input for voltage level that exceeds +35 VDC for longer than 250 ms. A low latched output is required to be generated upon detection of the interrupt that is used to inhibit the remote power source.

COMMANDS	DESCRIPTION
`INP:RANG 100,(@1)`	*Selects ±100 V as the input range for Channel 1.*
`INP:DEB .25`	*Sets the debounce time limit to 250 ms.*
`INP:MASK 1,(@1)`	*Enables Channel 1 to generate an interrupt.*
`INP:MASK 0,(@2:16)`	*Disables Channels 2 through 16 from generating an interrupt.*
`INP:POL NORM,(@1)`	*Selects Channel 1 to generate an interrupt when Channel 1 is greater than the offset voltage.*
`SENS:COND:LEV +35,(@1)`	*Selects +35 V as the offset (reference) voltage.*
`OUTP:POL:EXT:LATC INV`	*Sets the external latched output to be active low.*

Note: The INPut:OFFSet command can be used in place of the SENSe:CONDition:LEVel command. The INPUT:OFFSet command is normalized to the 10 V range, so, in order to have the same +35 V offset, the command must be entered as:

```
INPut:OFFSet +3.5,(@1)
```

BRACKETING A VOLTAGE

In this example, an input voltage level will be bracketed for an over- or under-voltage error condition. The input voltage of 5.0 V will be monitored for an over-voltage of 5.25 V and an under-voltage of 4.75 V. The error condition must be true for greater than 750 µs. An interrupt will be generated if either of these conditions occur. Channel 1 will be used for an over-voltage and Channel 2 for an under-voltage. The positive inputs of Channels 1 and 2 are tied together externally, along with their negative inputs. The output interrupt will not be latched, but will be pulsed.

COMMANDS	DESCRIPTION
`INP:RANG 10,(@1,2)`	*Selects ±10 V as the input range for Channel 1 & 2.*
`INP:DEB 75e-5`	*Sets the debounce time limit to 750 µs.*
`INP:MASK 1,(@1,2)`	*Enables Channel 1 or 2 to generate an interrupt.*
`INP:MASK 0,(@3:16)`	*Disables Channels 3 through 16 from generating an interrupt.*
`INP:POL NORM,(@1)`	*Selects Channel 1 to generate an interrupt when Channel 1 is greater than the offset voltage.*
`INP:POL INV,(@2)`	*Selects Channel 2 to generate an interrupt when Channel 2 is less than the offset voltage.*
`SENS:COND:LEV +5.25,(@1)`	*Selects +5.25 V as the offset (reference) voltage for Channel 1.*
`SENS:COND:LEV +4.75,(@2)`	*Selects +4.75 V as the offset (reference) voltage for Channel 2.*
`OUTP:POL:EXT:INT NORM`	*Sets the external interrupt output to be active high.*

Note: The INPut:OFFSet command can be used in place of the SENSe:CONDition:LEVel command. In the 10 V range, the numbers used are identical to the SENSe:CONDition:LEVel commands:

```
INPut:OFFSet +5.25,(@1)
INPut:OFFSet +4.75,(@2)
```

REGISTER ACCESS EXAMPLES

TABLE 3-3 REGISTER MAP

3E	
3C	
3A	
38	Interrupt enable (Write Only, Pseudo Only)
36	
34	
32	
30	First latched (Read Only)
2E	
2C	
2A	
28	Conditioned (Read Only)
26	
24	
22	
20	Raw (Read Only)
1E	
1C	
1A	
18	
16	
14	
12	
10	
E	
C	
A	
8	
6	
4	
2	
0	

The VM4018 module supports direct register access for high-speed data retrieval. The register map is as specified in Table 3-3.

In order to access the raw data using register access, the register at offset 0x20 must be read. Each bit in this register corresponds to the state of the 16 channel inputs (unmasked and non-inverted). Bit 1 corresponds to Channel 1, Bit 2 corresponds to Channel 2 and so on. This information can also be accessed using the Word Serial **FETC:RAW?** query.

In order to access the conditioned data using register access, the register at offset 0x28 must be read. Each bit in this register corresponds to the state of the 16 channel inputs (masked and inverted). Bit 1 corresponds to Channel 1, Bit 2 corresponds to Channel 2 and so on. This information can also be accessed using the Word Serial **FETC:COND?** query.

In order to access the first latched information using register access, the register at offset 0x30 must be read. Each bit in this register corresponds to the state of the 16 channel inputs. Bit 1 corresponds to Channel 1, Bit 2 corresponds to Channel 2 and so on. This information can also be accessed using the Word Serial **FETC:LATC?** query.

For example

a) If a value of 0x8000 is read from the first latched register, then it means that Channel 16's input has caused a latching.

b) If a value of 0xF000 is read from the first latched register, then it means that Channels 13 through 16 have caused a latching.

The Interrupt Enable register is a write-only register on which write operations take effect only in the Pseudo mode. In order to enable backplane interrupting, a non-zero value must be written to this register at offset 0x38. Writing a zero to this register will disable any backplane interrupting. It must be noted that in Non-pseudo mode, any writes to this register will not take effect. Backplane interrupting can also be enabled/disabled using the Word Serial **INHOUSE:REG_ENABLE** command.

PSEUDO REGISTER ACCESS

The VM4018 can be operated upon using (a) Word Serial Commands or (b) Register Access.

The VM4018 allows two types of register accesses (a) Direct Register Access using Hardware registers and (b) Pseudo Register Access. This can be configured using the **INHOUSE:PSEUDO** command.

Direct Register Access is considerably faster than Pseudo Register Access. However, the former does not provide certain features provided by the latter. Using Pseudo Register Access (a) a register read of **FIRST LATCHED** data allows another **FIRST LATCHED** event to occur, (b) allows for clearing of the first latched register upon register access rather than a Word Serial **FETC:LATC**? and (c) allows configuration of the type of backplane interrupting.

The module can be enabled for backplane interrupts using the **INHOUSE:REG_ENABLE** command. It can also be done by writing a non-zero value to the Interrupt Enable Register at offset 0x38 provided the module has been configured for Pseudo register access. The module can be instructed to clear the first latched register on register access/WS read using the **INHOUSE:CLEAR_LATCH** command. When the VXIbus backplane interrupting is enabled, the module will generate interrupts whenever latching of the first latched register takes place. If a Pseudo register access of the first latched register at offset 0x30 is performed or a Word Serial read (using **FETC:LATC?**) is performed, the latch register gets cleared allowing further latching to occur provided the module has been instructed to clear the first latched register. If the clearing of the first latched register is disabled, after the first latching takes place, the module cannot generate backplane interrupts.

Using the Direct Register Access, backplane interrupts are generated when the latching takes place for the first time. For further interrupting to occur, the Word Serial **FETC:LATC?** query must be performed.

Two types of backplane interrupts can be generated. They are (a) the reqt/reqf (in response to an **IACK** cycle) or (b) a single backplane interrupt. This can be configured using the **INHOUSE:REGINT** command. However, it must be noted that the module can be configured for only for mode at any given point in time. The former mode provides compatibility with the VXI standards and is the default mode. The latter allows for faster processing since it cuts down servicing of interrupts by 50% (since only 1 interrupt needs to be serviced for each latch event).

VXIPLUG&PLAY DRIVER EXAMPLES

```
/*****************************************************************************
      Function:   vtVM4018_setup_and_read_data
      Formal Parameters
            ViSession   instr_hndl
            - A valid sessionandle to the instrument.

            ViInt16 channel_list[]
            - This parameter specifies the channels which are to be setup.
            Only the specified channels will be enabled, the rest will be
            disabled.

            Each channel number in the array has the range :
                  vtVM4018_MIN_CHANNEL_NO (1) to
                  vtVM4018_MAX_CHANNEL_NO (16)

            ViInt16 num_of_channels
            - This parameter specifies the number of channels in the channel
            list.

            Valid Range:
                  vtVM4018_MIN_CHANNEL_NO (1) to
                  vtVM4018_MAX_CHANNEL_NO (16)

            ViReal32 offset[],
            - This parameter specifies the offset voltage to be configured for
            the input channels.

            Valid Range:
                   vtVM4018_MIN_VOLTAGE_LEVEL (-10.00 V) to
                   vtVM4018_MAX_VOLTAGE_LEVEL (9.96 V)

            ViInt16 polarity[]
            - This parameter specifies the polarity to be configured for the
            specified channels.

            Valid Range:
                  vtVM4018_INVERTED_POLARITY (0) or
                  vtVM4018_NORMAL_POLARITY   (1)

            ViInt16 voltage_range[]
            - This parameter specifies the voltage range to be configured for
            the specified channels.

            Valid Range:
                  vtVM4018_10VOLTS_RANGE  (0) or
                  vtVM4018_100VOLTS_RANGE (1)
```

```
            ViPInt16 first_latched_reg
            - This parameter returns the first input channel which crosses the
            programmed threshold voltage.

            ViPInt16 raw_data
            - This returns the 16 bit value that represents the current
            unconditioned [raw] state of the inputs.

            ViPInt16 conditioned_data
            - This returns the 16 bit value that represents the current
            conditioned state of the inputs.

      Return Values:    Returns VI_SUCCESS if successful, else returns error
                        value.

      Description       This is an application function that shows how the
                        user can use core functions to enable/disable the
                        specified channels for interrupt generation and
                        configure the specified channels' various parameters
                        such as offset, polarity and voltage-range.  It
                        returns the value of the First Latched register which
                        records the first input channel to cross its threshold
                        and queries the current state of the inputs.  Note
                        that this function resets the module to its default
                        state.

*****************************************************************************/
ViStatus_VI_FUNC vtVM4018_setup_and_read_data(ViSession instr_hndl,
                  ViInt16 channel_list[],ViInt16 num_of_channels,ViReal32
                  offset[],ViInt16 polarity[],ViInt16 voltage_range[],
                  ViPInt16 first_latched_reg,ViPInt16 raw_data,
                  ViPInt16 conditioned_data)

{

/* Variable used to store return status of the function */
      ViStatus status = VI_NULL;

/* Reset to the default state */
      status = vtVM4018_reset(instr_hndl);
            if (status < VI_SUCCESS)
      return status;

/* Function to enable the selected channels to cause interrupt */
      status = vtVM4018_enable_disable_channels (instr_hndl,

      vtVM4018_ENABLE_CHANNEL, channel_list, num_of_channels);
            if (status < VI_SUCCESS)
      return vtVM4018_ERROR_MASK_OR_UNMASK_CHANNELS;
```

```
/* Function to set the offset, polarity and voltage range to the channels */

      status = vtVM4018_config_channels ( instr_hndl, channel_list,
                        num_of_channels, offset, polarity, voltage_range);
            if (status < VI_SUCCESS)
      return vtVM4018_ERROR_SETTING_CHANNELS;

/* Function to query the first latched register */
      status = vtVM4018_query_latched_reg(instr_hndl, first_latched_reg);
            if (status < VI_SUCCESS)
      return vtVM4018_ERROR_QUERYING_LATCHED_REG;

/* Function to query the Raw data */
      status = vtVM4018_read_data (instr_hndl,
                        vtVM4018_READ_RAW_DATA, raw_data);
            if (status < VI_SUCCESS)
      return vtVM4018_ERROR_READING_RAW_DATA;

/* Function to query the Conditioned data */
      status = vtVM4018_read_data (instr_hndl,
                        vtVM4018_READ_CONDITIONED_DATA, conditioned_data);
            if (status < VI_SUCCESS)
      return vtVM4018_ERROR_READING_CONDITIONED_DATA;

      return VI_SUCCESS;
```

SECTION 4

COMMAND DICTIONARY

INTRODUCTION

This section presents the instrument command set. It begins with an alphabetical list of all the commands supported by the VM4018 divided into three sections: IEEE 488.2 commands, the instrument specific SCPI commands and the required SCPI commands. With each command is a brief description of its function, whether the command's value is affected by the *RST command and its *RST value.

The remainder of this section is devoted to describing each command, one per page, in detail. Every command entry describes the exact command and query syntax, the use and range of parameters and a complete description of the command's purpose.

ALPHABETICAL COMMAND LISTING

The following tables provide an alphabetical listing of each command supported by the VM4018 along with a brief description. If an X is found in the column titled *RST, then the value or setting controlled by this command is possibly changed by the execution of the *RST command. If no X is found, then *RST has no effect. The default column gives the value of each command's setting when the unit is powered up or when a *RST command is executed. Note that calibration values revert to the values stored in non-volatile memory upon reset. Using the CALibration:DEFault command will return calibration values back to known, factory preset values.

In order for CALibration commands/queries to be executed, calibration security must be turned off. If security is not turned off, a "-203, Command Protected" error will be returned. See the *CALibration:SECure:CODE* command for information on calibration security.

TABLE 4-1 IEEE 488.2 COMMON COMMANDS

Command	Description	*RST	*RST Value
*CLS	Clears the Status Register	X	
*ESE	Sets the Event Status Enable Register	X	
*ESR?	Query the Standard Event Status Register		N/A
*IDN?	Query the module identification string		N/A
*OPC	Set the OPC bit in the Event Status Register		
*RST	Resets the module to a known state		N/A
*SRE	Set the service request enable register		
*STB?	Query the Status Byte Register		
*TRG	Causes a trigger event to occur		
*TST?	Starts and reports a self-test procedure		N/A
*WAI	Halts execution and queries	X	

TABLE 4-2 INSTRUMENT SPECIFIC SCPI COMMANDS

Command	Description	*RST	*RST Value
CALibration:DAC	Sets the DAC values directly.		Assumes the value last stored in non-volatile memory
CALibration:DEFault	Sets calibration values to known state.		
CALibration:INTerrupt	Enables or disables the interrupt to the VMIP CPU.		
CALibration:OFFSet	Calibrates one of two DAC factor.		
CALibration:MODE	This command reduces noise during calibration		
CALibration:SECure:CODE	Sets the code required to disable calibration security.		
CALibration:SECure:STATe	Indicates if calibration security is enabled or disabled.		
CALibration:SLOPe	Calibrates one of two DAC factor.		
CALibration:STORe	Stores changes made to calibration data.		
FETCh:CONDitioned?	Reads back the current conditioned (masked and inverted) output state of the comparators in the group.		
FETCh:LATCh ed?	Read back the value that was latched when the first input(s) in the group caused an active edge.		
FETCh:RAW?	Reads back the value that represents the current unconditioned (no masking or inversion) output state of the comparators in the group.		
INHOUSE:CLEAR_LATCH	Controls clearing of first latched information.	X	0
INHOUSE:PSEUDO	Sets the type of register interface used.	X	1
INHOUSE:REGINT	Controls the interrupt response type.	X	0
INHOUSE:REG_ENABLE	Interrupt masking.	X	0
INPut:DEBounce	This sets the debounce timing on a group of the analog comparators.	X	19.2 μs
INPut:MASK	Sets the masking for a group of channels.	X	0
INPut:MASK:INTerrupt	Enable or disable interrupt generation when changing MASKs	X	0
INPut:OFFSet	Sets the comparator threshold for a group of channels.	X	0 V
INPut:POLarity	Sets the polarity for a group of channels.	X	NORMal
INPut:RANGe	Sets the range for a group of channels.	X	250 V
OUTPut:POLarity:EXTernal:INTerrupt	Sets the polarity for the interrupt output on the front panel for one of the three groups.	X	NORMal
OUTPut:POLarity:EXTernal:LATCh ed	Sets the polarity for latched interrupt output on the front panel for one of the three groups.	X	NORMal
OUTPut:TTLTrg	Selects the TTL trigger line that will be used as an output.	X	0
OUTPut:TTLTrg:STATe	Enables/disables the ability to use TTL trigger lines as outputs.	X	0
SENSe:CONDition:LEVel	Sets the threshold for a group of channels.	X	0 V

TABLE 4-3 SCPI REQUIRED COMMANDS

Command	Description	*RST	*RST Value
STATus:OPERation:CONDition?	Queries the Operation Status Condition Register.	X	
STATus:OPERation:ENABle	Sets the Operation Status Enable Register.	X	
STATus:OPERation[:EVENt]?	Queries the Operation Status Event Register.	X	
STATus:PRESet	Presets the Status Register.	X	
STATus:QUEStionable:CONDition?	Queries the Questionable Status Condition Register.	X	
STATus:QUEStionable:ENABle	Sets the Questionable Status Enable Register.	X	
STATus:QUEStionable[:EVENt]?	Queries the Questionable Status Event Register.	X	
SYSTem:ERRor?	Queries the Error Queue.	X	Clears queue
SYSTem:VERsion?	Queries which version of the SCPI standard to which the module complies.		N/A

COMMAND DICTIONARY

The remainder of this section is devoted to the actual command dictionary. Each command is fully described on its own page. In defining how each command is used, the following items are described:

Purpose	Describes the purpose of the command.
Type	Describes the type of command such as an event or setting.
Command Syntax	Details the exact command format.
Command Parameters	Describes the parameters sent with the command and their legal range.
Reset Value	Describes the values assumed when the *RST command is sent.
Query Syntax	Details the exact query form of the command.
Query Parameters	Describes the parameters sent with the command and their legal range. The default parameter values are assumed the same as in the command form unless described otherwise.
Query Response	Describes the format of the query response and the valid range of output.
Description	Describes in detail what the command does and refers to additional sources.
Examples	Present the proper use of each command and its query (when available).
Related Commands	Lists commands that affect the use of this command or commands that are affected by this command.

IEEE 488.2 COMMON COMMANDS

*CLS

Purpose	Clears the Status Register	
Type	IEEE 488.2 Common Command	
Command Syntax	*CLS	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	This command clears all event registers, clears the OPC flag and clears all queues (except the output queue).	
Examples	**Command / Query**	**Response *(Description)***
	`*CLS`	*(Clears all status and event registers)*
Related Commands	None	

*ESE

Purpose	Sets the bits of the Event Status Enable Register
Type	IEEE 488.2 Common Command
Command Syntax	*ESE <mask>
Command Parameters	<mask> = numeric ASCII value from 0 to 255
***RST Value**	N/A
Query Syntax	*ESE?
Query Parameters	None
Query Response	Numeric ASCII value from 0 to 255
Description	The Event Status Enable command is used to set the bits of the Event Status Enable Register. See ANSI/IEEE 488.2-1987 section 11.5.1 for a complete description of the ESE register. A value of 1 in a bit position of the ESE register enables generation of the ESB (Event Status Bit) in the Status Byte by the corresponding bit in the ESR. If the ESB is set in the SRE register then an interrupt will be generated. See the *ESR? command for details regarding the individual bits. The ESE register layout is: Bit 0 - Operation Complete Bit 1 - Request Control (not used in the VM4018) Bit 2 - Query Error Bit 3 - Device Dependent Error (not used in the VM4018) Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request (not used in the VM4018) Bit 7 - Power On The Event Status Enable query reports the current contents of the Event Status Enable Register.

Examples	**Command / Query**	**Response *(Description)***
	`*ESE 36` `*ESE?`	36 *(Returns the value of the event status enable register)*

Related Commands	*ESR?

*ESR?

Purpose	Queries and clears the Standard Event Status Register	
Type	IEEE 488.2 Common Command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	ESR?	
Query Parameters	None	
Query Response	Numeric ASCII value from 0 to 255	
Description	The Event Status Register query - queries and clears the contents of the Standard Event Status Register. This register is used in conjunction with the ESE register to generate the ESB (Event Status Bit) in the Status Byte. The layout of the ESR is: Bit 0 - Operation Complete Bit 1 - Request Control (not used in the VM4018, always 0) Bit 2 - Query Error Bit 3 - Device Dependent Error (not used in the VM4018, always 0) Bit 4 - Execution Error Bit 5 - Command Error Bit 6 - User Request (not used in the VM4018, always 0) Bit 7 - Power On The Operation Complete bit is set by the VM4018 when it receives an *OPC command. The Query Error bit is set when data is over-written in the output queue. This could occur if one query is followed by another without reading the data from the first query. The Execution Error bit is set when an execution error is detected. See the section in the manual covering Error Messages for a list of execution error. Errors that range from -200 to -299 are execution errors. The Command Error bit is set when a command error is detected. See the section in this manual covering Error Messages for a list of command errors. Errors that range from -100 to -199 are command errors. The Power On bit is set when the module is first powered on or after it receives a reset via the VXI Control Register. Once the bit is cleared (by executing the *ESR? command) it will remain cleared.	
Examples	**Command / Query**	**Response *(Description)***
	`*ESR?`	4
Related Commands	*ESE	

*IDN?

Purpose	Queries the module for its identification string	
Type	IEEE 488.2 Common Command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*IDN?	
Query Parameters	None	
Query Response	ASCII character string	
Description	The Identification query returns the identification string of the VM4018 module. The response is divided into four fields separated by commas. The first field is the manufacturer's name, the second field is the model number, the third field is an optional serial number and the fourth field is the firmware revision number. If a serial number is not supplied, the third field is set to 0 (zero).	
Examples	**Command / Query**	**Response *(Description)***
	`*IDN?`	VXI Technology, Inc.,VM4018,0,1.0 *(The revision listed here is for reference only; the response will always be the current revision of the instrument.)*
Related Commands	None	

*OPC

Purpose	Sets the OPC bit in the Event Status Register
Type	IEEE 488.2 Common Command
Command Syntax	*OPC
Command Parameters	None
***RST Value**	N/A
Query Syntax	*OPC?
Query Parameters	None
Query Response	1
Description	The Operation Complete command sets the OPC bit in the Event Status Register when all pending operations have completed. The Operation Complete query will return a 1 to the output queue when all pending operations have completed.

Examples	**Command / Query**	**Response *(Description)***
	`*OPC`	*(Sets the OPC bit in the Event Status Register)*
	`*OPC?`	1 *(Returns the value of the Event Status Register)*
Related Commands	*WAI	

*RST

Purpose	Resets the module's hardware and software to a known state	
Type	IEEE 488.2 Common Command	
Command Syntax	*RST	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Reset command resets the module's hardware and software to a known state. See the command index at the beginning of this chapter for the default parameter values used with this command.	
Examples	**Command / Query**	**Response *(Description)***
	`*RST`	*(Resets the module)*
Related Commands	None	

*SRE

Purpose	Set the service request enable register
Type	IEEE 488.2 Common Command
Command Syntax	*SRE <mask>
Command Parameters	<mask> =numeric ASCII value from 0 to 255
***RST Value**	None – Required parameter
Query Syntax	*SRE?
Query Parameters	None
Query Response	Numeric ASCII value from 0 to 255
Description	The service request enable mask is used to control which bits in the status byte generate back plane interrupts. If a bit is set in the mask that newly enables a bit set in the status byte and interrupts are enabled, the module will generate a REQUEST TRUE event via an interrupt. See the *STB? Command for the layout of bits. **Note**: Bit 6 is always internally cleared to zero as required by IEEE 488.2 section 11.3.2.3. Bit 0 - Latch Event Bit 1 - Unused Bit 2 - Error Queue Has Data Bit 3 - Questionable Status Summary (not used) Bit 4 - Message Available Bit 5 - Event Status Summary Bit 6 - Master Summary Status Bit 7 - Operation Status Summary

Examples	**Command / Query**	**Response *(Description)***
	`*SRE 4`	*(Sets the service request enable register)*
	`*SRE?`	4 *(Returns the value of the SRE register)*

Related Commands	*STB

*STB?

Purpose	Queries the Status Byte Register	
Type	IEEE 488.2 Common Command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	*STB?	
Query Parameters	None	
Query Response	Numeric ASCII value from 0 to 255	
Description	The Read Status Byte query fetches the current contents of the Status Byte Register. See the IEEE 488.2 specification for additional information regarding the Status byte Register and its use. The layout of the Status Register is: Bit 0 - Latched Event Bit 1 - Unused Bit 2 - Error Queue Has Data Bit 3 - Questionable Status Summary (not used) Bit 4 - Message Available Bit 5 - Event Status Summary Bit 6 - Master Summary Status Bit 7 - Operation Status Summary	
Examples	**Command / Query**	**Response *(Description)***
	`*STB?`	16 (*Queries the Status Byte Register)*
Related Commands	None	

*TRG

Purpose	Causes a trigger event to occur	
Type	IEEE 488.2 Common Command	
Command Syntax	*TRG	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Trigger command causes a trigger event to occur. This command is not used for this module and is provided for SCPI compliance only.	
Examples	**Command / Query**	**Response *(Description)***
	`*TRG`	*(Triggers an event)*
Related Commands	None	

*TST?

Purpose	Causes a self-test procedure to occur and queries the result
Type	IEEE 488.2 Common Command
Command Syntax	None – query only
Command Parameters	N/A
***RST Value**	N/A
Query Syntax	*TST?
Query Parameters	None
Query Response	Numeric ASCII value
Description	The Self-Test query causes the VM4018 to run its self-test procedures and report on the results. A value of 0 indicates that the test passed.

Examples	**Command / Query**	**Response *(Description)***
	`*TST?`	0 (*Begins the self-test procedure returns the result)*

Related Commands	None

*WAI

Purpose	Halts execution of commands and queries until the No Operation Pending message is true	
Type	IEEE 488.2 Common Command	
Command Syntax	*WAI	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None	
Query Parameters	N/A	
Query Response	N/A	
Description	The Wait to Continue command halts the execution of commands and queries until the No Operation Pending message is true. This command makes sure that all previous commands have been executed before processing. It provides a way of synchronizing the module with its master.	
Examples	**Command / Query**	**Response *(Description)***
	`*WAI`	*(Pauses the execution of additional commands until the No Operation Pending message is true.)*
Related Commands	*OPC	

INSTRUMENT SPECIFIC SCPI COMMANDS

CALibration:DAC

Purpose	This command sets the DAC values directly
Type	Instrument specific command
Command Syntax	CALibration:DAC <dac_value>,<channel_list>
Command Parameters	<dac_value> = 0 - 16383 <channel_list> = standard channel list syntax supporting channels 1 to 16
***RST Value**	N/A
Query Syntax	CALibration:DAC? <channel>
Query Parameters	<channel> = 1 to 16
Query Response	Returns the set value for the <dac_value> parameter for the specified channel
Description	This command sets the DAC values directly. The <dac_value> factory default is 8192. **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:DAC 8192,(@1:8)`	*(Sets the DAC for Channels 1 through 8 to 8192)*
	`CAL:DAC? 1`	8192 *(Returns the set DAC value for Channel 1)*

Related Commands	CALibration:SECure:CODE CALibration:SECure:STATe

CALibration:DEFault

Purpose	Initializes calibration values to known values
Type	Instrument specific command
Command Syntax	CALibration:DEFault
Command Parameters	N/A
***RST Value**	N/A
Query Syntax	N/A
Query Parameters	N/A
Query Response	N/A
Description	Initializes calibration values to known values. The following values are set: Security Code = VM4018 Serial Number = 0 Product = 4018 Offset = 8192 (all channels) Slope = ±1.0 V Range, 122.07 µV/count ±2.5 V Range, 305.19 µV/count ±10.0 V Range, 1.2207 mV/count ±25.0 V Range, 3.0519 mV/count ±100.0 V Range, 12.207 mV/count ±250.0 V Range, 30.519 mV/count Executing this command changes all previously set calibration values.  **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:DEF`	*(Initializes calibration values to known values)*

Related Commands	All CALibration commands

CALibration:INTerrupt

Purpose	This command enables or disables the interrupt to the VMIP CPU
Type	Instrument specific command
Command Syntax	CALibration:INTerrupt <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON
***RST Value**	N/A
Query Syntax	CALibration:INTerrupt?
Query Parameters	N/A
Query Response	Returns the set value for the <boolean> parameter
Description	This command enables or disables the interrupt to the VMIP CPU. The <boolean> factory default value is 1. **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:INT 1`	*(Enables CPU interrupt)*
	`CAL:INT?`	1 *(Indicates that CPU interrupts are enabled)*

Related Commands	CALibration:SECure:CODE CALibration:SECure:STATe

CALibration:MODE

Purpose	This command reduces noise during calibration
Type	Instrument specific command
Command Syntax	CALibration:MODE <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON
***RST Value**	N/A
Query Syntax	CALibration:MODE?
Query Parameters	N/A
Query Response	Returns the set value for the <boolean> parameter
Description	Enabling this command during calibration reduces noise from the input circuitry. The <boolean> factory default value is 0.  **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:MODE 1` `CAL:MODE?`	*(Enables noise reduction during calibration)* 1 *(Indicates that noise reduction is enabled)*

Related Commands	CALibration:SECure:CODE CALibration:SECure:STATe

CALibration:OFFSet

Purpose	This command calibrates one of two factors used by the DAC
Type	Instrument specific command
Command Syntax	CALibration:OFFSet <integer>,<channel_list>
Command Parameters	<integer> = 7192 – 9192 <channel_list> = standard channel list syntax supporting channels 1 to 16
***RST Value**	N/A
Query Syntax	CALibration:OFFSet? <channel>
Query Parameters	<channel> = 1 to 16
Query Response	Returns the set value for the <integer> parameter for the specified channel
Description	This command calibrates one of two factors used by the DAC. Prior to setting this calibration command, the INPut:RANGE command should be set, as the calibration values are range dependent. One offset value may be set per channel per range. The stored data is also specific to the polarity set by the INPut:POLarity command. The VM4018 can store a different value for NORMal and INVerted operation. The <integer> factory default value is 8192.  **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`INP:RANG 100,(@1:8)`	*(Sets the input range to 100 V)*
	`INP:POL NORM`	*(Sets the input trigger to send an interrupt when above threshold)*
	`CAL:OFFSET 8192,(@1:8)`	*(Sets the offset to 8192 for Channels 1 - 8 for normal operations in the 100 V range)*
	`CAL:OFFSET? 3`	8192 *(Returns the set offset value)*

Related Commands	CALibration:SECure:CODE CALibration:SECure:STATe INPut:POLarity INPut:RANGe

CALibration:SECure:CODE

Purpose	Sets the code required to disable calibration security
Type	Instrument specific command
Command Syntax	CALibration:SECure:CODE <string>
Command Parameters	<string> = the code string can be from 1 to 12 ASCII characters in length entered in IEEE 488.2 definite or indefinite length arbitrary block format
***RST Value**	N/A
Query Syntax	CALibration:SECure:CODE?
Query Parameters	N/A
Query Response	Returns the security code
Description	The Calibration Security Code command sets the code required to disable calibration security. Calibration security must first be disabled before the code can be changed. Before shipping the instrument, the factory code setting is VM4018. **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:SEC:CODE #16VM4018`	*(Sets the security code setting for the VM4018)*
Related Commands	CALibration:SECure:STATe CALibration:STORe	

CALibration:SECure[:STATe]

Purpose	Enable or disable calibration security
Type	Instrument specific command
Command Syntax	CALibration:SECure[:STATe] <boolean>,<string>
Command Parameters	<boolean> = 0 | 1 | OFF | ON <string> = the code string can be from 1 to 12 ASCII characters in length entered in IEEE 488.2 definite or indefinite length arbitrary block format
***RST Value**	<boolean> = ON
Query Syntax	CALibration:SECure:STATe?
Query Parameters	N/A
Query Response	Returns the set <boolean> value
Description	The Calibration Security State enables or disables the calibration security. While security is on, no stores to the non-volatile memory are allowed. In order to disable the security state, the security code must be supplied. To enable the security, the code does not need to be supplied. The security state is enabled by default. Security can also be enabled without entering the <string> parameter by sending the CALibration:SECure:STATe 1 | ON command. Security cannot, however, be disabled using a CALibration:SECure:STATe 0 | OFF command. If this command is sent, a "Missing Parameter" error will be returned.  **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:SEC:STAT OFF,#16VM4018`	*(Disables calibration security)*
	`CAL:SEC:STAT 1`	*(Turns the calibration security back on)*
	`CAL:SEC:STAT?`	1 *(Indicates that calibration security is enabled)*

Related Commands	CALibration:SECure:CODE CALibration:STORe

CALibration:SLOPe

<table>
<tr><td>Purpose</td><td>This command calibrates one of two factors used by the DAC</td></tr>
<tr><td>Type</td><td>Instrument specific command</td></tr>
<tr><td>Command Syntax</td><td>CALibration:SLOPe <float_value>,<channel_list></td></tr>
<tr><td>Command Parameters</td><td><float_value> = any value approximately ±5% of the default slope (see below)
<channel_list> = standard channel list syntax supporting channels 1 to 16</td></tr>
<tr><td>*RST Value</td><td>N/A</td></tr>
<tr><td>Query Syntax</td><td>CALibration:SLOPe? <channel></td></tr>
<tr><td>Query Parameters</td><td><channel> = 1 to 16</td></tr>
<tr><td>Query Response</td><td>Returns the set <float_value> parameter of the specified channel.</td></tr>
<tr><td>Description</td><td>This command calibrates one of two factors used by the DAC. Prior to setting this calibration command, the INPut:RANGe command should be set, as the calibration values are range dependent. One <float_value> can be set per channel per range.

The default values for this command are as follows:

<table>
<tr><th>Range</th><th>Default Slope</th></tr>
<tr><td>±1.0</td><td>122.07 µV/count</td></tr>
<tr><td>±2.5</td><td>305.19 µV/count</td></tr>
<tr><td>±10.0</td><td>1.2207 mV/count</td></tr>
<tr><td>±25.0</td><td>3.0519 mV/count</td></tr>
<tr><td>±100.0</td><td>12.207 mV/count</td></tr>
<tr><td>±250.0</td><td>30.519 mV/count</td></tr>
</table>

Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly</td></tr>
<tr><td>Examples</td><td>
<table>
<tr><th>Command / Query</th><th>Response (Description)</th></tr>
<tr><td>INP:RANG 100,(@1:4)</td><td>(Sets the input range to 100 V for Channels 1-4)</td></tr>
<tr><td>CAL:SLOP -12.207,(@1:4)</td><td>(Sets the slope value to -12.207 mV/count for Channels 1 -4)</td></tr>
<tr><td>CAL:SLOP? 3</td><td>-12.207 (Returns the set slope value)</td></tr>
</table>
</td></tr>
<tr><td>Related Commands</td><td>CALibration:SECure:CODE
CALibration:SECure:STATe</td></tr>
</table>

CALibration:STORe

Purpose	Stores calibration data into non-volatile memory
Type	Instrument specific command
Command Syntax	CALibration:STORe
Command Parameters	N/A
***RST Value**	N/A
Query Syntax	N/A
Query Parameters	N/A
Query Response	N/A
Description	The CALibration:STORe command stores correction data into non-volatile memory. The correction data is calibration data that has been downloaded via the program messages in the Calibration Data subsystem. The CALibration:STORe command should only be performed after all the correction data has been finalized. **Note**: Security must be turned off in order to use this command.  **Calibration commands should only be executed by qualified personnel. Changing these values incorrectly can cause the instrument to perform improperly**

Examples	**Command / Query**	**Response *(Description)***
	`CAL:SEC OFF,#16VM4018`	*(Disables security)*
	`CAL:STOR`	*(Stores correction data into non-volatile memory)*
Related Commands	CALibration:SECure:CODE CALibration:SECure:STATe	

FETCh:CONDitioned?

Purpose	Returns the state of the conditioned (masked and inverted) inputs	
Type	Query	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	FETCh:CONDitioned?	
Query Parameters	None	
Query Response	Numeric ASCII in the range of 0 to 65535	
Description	The FETCh:CONDitioned query reports the 16-bit value that represents the current conditioned (masked and inverted) state of the inputs. This information is also available at the VXIbus register level at offset 0x28.	
Examples	**Command / Query**	**Response *(Description)***
	`FETC:COND?`	0 *(Returns the current conditioned state of the inputs)*
Related Commands	FETch:RAW?	

FETCh:LATCh?

Purpose	Reports the active signals in the First Latched register
Type	Query
Command Syntax	N/A
Command Parameters	N/A
***RST Value**	N/A
Query Syntax	FETCh:LATCh?
Query Parameters	N/A
Query Response	Numeric ASCII in the range 0 to 65535
Description	The FETCh:LATCh query reports the active signals in the First Latched register. The First Latched register records the active signals when the first new input channel(s) crosses its threshold. This information is also available at the VXIbus register level at offset 0x30.

Examples	**Command / Query**	**Response *(Description)***
	`FETC:LATC?`	1 *(Returns the active signal in the First Latched register)*
Related Commands	INHOUSE:CLEAR_LATCH	

FETCh:RAW?

Purpose	Returns the state of the unconditioned (unmasked and non-inverted) inputs	
Type	Query	
Command Syntax	N/A	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	FETCh:RAW?	
Query Parameters	N/A	
Query Response	Numeric ASCII in the range 0 to 65535	
Description	The FETCh:RAW query reports the 16-bit value that represents the current unconditioned (unmasked and non-inverted) state of the inputs. This information is also available at the register level at offset 0x20.	
Examples	**Command / Query**	**Response *(Description)***
	`FETC:RAW?`	1 *(Returns the current unconditioned state of the inputs)*
Related Commands	FETch:CONDitioned?	

INHOUSE:CLEAR_LATCH

Purpose	Controls whether the first latched information will be cleared when read by word serial or pseudo register access of the first latched register	
Type	Setting	
Command Syntax	INHOUSE:CLEAR_LATCH <boolean>	
Command Parameters	<boolean> = 0 | 1 | OFF | ON	
***RST Value**	0	
Query Syntax	INHOUSE:CLEAR_LATCH?	
Query Parameters	N/A	
Query Response	ASCII numeric 0 or 1	
Description	CLEAR_LATCH determines whether the first latched information will be cleared when the information is read. For some, this provides confidence that another interrupt has not occurred. The information is cleared with the word serial FETCh:LATCh ed? command. It is also cleared if PSEUDO is set and a register read of the first latched information occurs. The information is not cleared if a hardware register read is used. When the information is cleared, all following reads will return a value of 0 until a new first latched event occurs. **Note**: All letters of the command are required; there is no short form of the command.	
Examples	**Command / Query**	**Response *(Description)***
	`INHOUSE:CLEAR_LATCH 1` `INHOUSE:CLEAR_LATCH?`	 1
Related Commands	INHOUSE:PSEUDO FETCh:LATCh ed?	

INHOUSE:PSEUDO

Purpose	Controls the use of the register interface
Type	Setting
Command Syntax	INHOUSE:PSEUDO <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON Factory Default = 1
***RST Value**	N/A
Query Syntax	INHOUSE:PSEUDO?
Query Parameters	N/A
Query Response	ASCII numeric 0 or 1
Description	If INHOUSE:PSEUDO is set true (1 or ON), the instrument uses the pseudo register interface. If false (0 or OFF), the instrument uses the hardware register interface. The value set is implemented upon the next power cycle. This command does not take effect immediately. The pseudo register interface allows use of the REG_ENABLE capability as well as the CLEAR_LATCH capability from the registers. These capabilities are not available with the hardware register interface. The hardware register interface is much faster than the pseudo register interface (speeds are controller dependent but, as an example, with one controller a hardware register access takes about 0.5 µs while a pseudo register access takes about 25 µs). The hardware register, however, interface lacks the above two features. Pseudo registers are needed if the user wants to perform a register read or a word serial FETch:LATChed? of FIRST LATCHED data in order to allow another FIRST LATCHED to occur. If pseudo is not set, then the user can read registers at hardware register speed but a word serial read FETCh:LATChed? is required to allow another FIRST LATCHED to occur. If pseudo is set, then the user can read the registers at pseudo register speed but the read of the latched data will allow a new FIRST LATCHED to occur. Pseudo also allows a register write to control the masking of interrupts for REGINT. **Note**: All letters of the command are required; there is no short form of the command.

Examples	**Command / Query**	**Response *(Description)***
	`INHOUSE:PSEUDO 1`	*(Selects the PSEUDO register)*
	`INHOUSE:PSEUDO?`	1 *(Indicates that the PSEUDO register is selected)*

Related Commands	INHOUSE:REG_ENABLE INHOUSE:CLEAR_LATCH

INHOUSE:REGINT

Purpose	Controls the module's response type to an interrupt acknowledge cycle
Type	Setting
Command Syntax	INHOUSE:REGINT <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON
***RST Value**	0
Query Syntax	INHOUSE:REGINT?
Query Parameters	N/A
Query Response	ASCII numeric 0 or 1
Description	The INHOUSE:REGINT command controls the module's response type to an interrupt acknowledge cycle. When REGINT is set to false, the module uses reqt/reqf (request true/request false), provided the latched interrupt bit is set tin the 'SRE'. A reqt (upper 8 bits are 0x7D) is generated for every latched event and a reqf (upper 8 bits are 0x7C) is generated for every reading of the latched information using either pseudo register access or word serial FETch:LATChed? Command. When REGINT is set to true, only one interrupt is generated every time a latching occurs. The upper 8 bits of the 16-bit SRE register on (0x7B). **Note**: All letters of the command are required; there is no short form of the command.

Examples	**Command / Query**	**Response *(Description)***
	`INHOUSE:REGINT 1`	*(Sets REGINT to true)*
	`INHOUSE:REGINT?`	1 *(Indicates that REGINT is set to true)*

Related Commands	INHOUSE:REG_ENABLE INHOUSE:PSEUDO

INHOUSE:REG_ENABLE

Purpose	Controls the masking of REGINT
Type	Setting
Command Syntax	INHOUSE:REG_ENABLE <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON
***RST Value**	0
Query Syntax	INHOUSE:REG_ENABLE?
Query Parameters	N/A
Query Response	ASCII numeric 0 or 1
Description	REG_ENABLE controls the masking of REGINT. If REG_ENABLE is zero, then no backplane interrupt can be generated. If REG_ENABLE is a non-zero number, then a backplane interrupt can be generated. If PSEUDO is set to on, then a write to the register at offset 0x38 also controls the masking - zero disables, a non-zero enables. This command is included for completeness. Enable/disable capabilities are provided in the pseudo register interface to allow a complete register interface. This command just provides that same capability in the word serial interface. **Note**: All letters of the command are required; there is no short form of the command.

Examples	**Command / Query**	**Response *(Description)***
	`INHOUSE:REG_ENABLE 1`	
	`INHOUSE:REG_ENABLE?`	1

Related Commands	INHOUSE:PSEUDO INHOUSE:REGINT

INPut:DEBounce

Purpose	Sets the debounce time
Type	Setting
Command Syntax	INPut:DEBounce <value>
Command Parameters	<value> = 12.8 µs to 0.4194304 s
***RST Value**	12.8 µs
Query Syntax	INPut:DEBounce?
Query Parameters	N/A
Query Response	Numeric ASCII value from 0.0000128 to 0.4194304
Description	The INPut:DEBounce command sets the time period for the digital debounce circuitry. By programming a debounce time of 1 ms, an input must exceed its threshold level for a period of 1 ms before it is recognized as a valid input. The debounce resolution is 6.4 µs. The debounce time set is applied to all channels. A signal with noise near the debounce threshold may cause the instrument to send interrupts indefinitely, making the VM4018 unresponsive. To avoid this, ensure that the offset and/or INPut:DEBounce times are properly set.

Examples	**Command / Query**	**Response *(Description)***
	`INP:DEB 12.8-6`	*(Sets a digital debounce time of 12.8 µs)*
	`INP:DEB`	0.0000128 *(Indicates that the debounce time is set to 12.8 µs)*
Related Commands	None	

INPut:MASK

Purpose	Masks unused input channels
Type	Setting
Command Syntax	INPut:MASK <state>,<channel_list>
Command Parameters	<channel_list> = standard channel list syntax supporting channels 1 to 16 <state> = ON | 1 | OFF | 0
***RST Value**	0 for all channels
Query Syntax	INPut:MASK? <channel>
Query Parameters	<channel> = 1 to 16
Query Response	Numeric ASCII value of 1 or 0
Description	The INPut:MASK command selects which channels are enabled for input voltage comparison. When a channel is programmed to be ON or 1 then it is enabled to generate interrupts. If a channel is programmed to be OFF or 0, then it cannot generate VXIbus interrupts.

Examples	**Command / Query**	**Response *(Description)***
	`INP:MASK 0,(@1:8)`	*(Makes Channels 1 – 8 incapable of generating VXIbus interrupts)*
	`INP:MASK? 3`	0 *(Indicates that Channel 3 is incapable of generating VXIbus interrupts.)*
Related Commands	None	

INPut:MASK:INTerrupt

Purpose	Enable or disable interrupt generation when changing MASKs
Type	Setting
Command Syntax	INPut:MASK:INTerrupt <boolean>
Command Parameters	<boolean> = 0 | 1 | OFF | ON
***RST Value**	0
Query Syntax	INPut:MASK:INTerrupt?
Query Parameters	N/A
Query Response	ASCII numeric 0 or 1
Description	The INPut:MASK:INTerrupt command enables or disables interrupt generation when changing MASK values. When set to 0 (the *RST state), interrupts are temporarily disabled whenever MASK values are changed. When set to 1, interrupts are generated even as MASK values are changed. When a MASK is first enabled, an interrupt is generated if a channel is beyond its threshold. To create an interrupt when this occurs, set this command to 1. Example 1: If a channel is set for NORMal polarity and the channel's input is higher than its threshold, an interrupt is generated. This interrupt will be ignored when INPut:MASK:INTerrupt is set to 0. Only when the channel's input goes below its threshold, and then goes above the threshold for a period longer than the INPut:DEBounce time, will an interrupt be generated. Example 2: If a channel is set for NORMal polarity and the channel's input is higher than its threshold, an interrupt is generated. This interrupt will be recognized when INPut:MASK:INTerrupt is set to 1. When the channel's input goes below its threshold, and then goes above the threshold for a period longer than the INPut:DEBounce time, another interrupt will be generated.
Examples	**Command / Query**: `INP:MASK:INT 1` **Response *(Description)***: *(Enables interrupt generation while changing mask values)* **Command / Query**: `INP:MASK:INT?` **Response *(Description)***: 1 *(Indicates that Input Mask Interrupt is enabled)*
Related Commands	All INPut commands

INPut:OFFSet

Purpose	Sets the input threshold for a group of channels
Type	Setting
Command Syntax	INPut:OFFSet <voltage_level>,<channel_list>
Command Parameters	<voltage_level> = -10 volts to +10 volts <channel_list> = standard channel list syntax supporting channels 1 to 16
***RST Value**	0 V for all channels
Query Syntax	INPut:OFFSet? <channel>
Query Parameters	<channel> = 1 to 16
Query Response	ASCII numeric value from -10.00 to +10.00
Description	The INPut:OFFSet command sets the input threshold for a group of channels over which the input signal must cross to cause an interrupt event. The command sets the value in the 12-bit DAC to which the input signal is compared. This command is effective only for channels operating in single-ended mode. Note that the actual input offset value is affected by the INPut:RANGe command. The input offset is normalized to the ±10 V range. The actual input offset values for the other ranges are as follows:

Range	Entered Threshold	Actual Threshold
±1.0	x	0.1x
±2.5	x	0.25x
±10.0	x	1.0x
±25.0	x	2.5x
±100.0	x	10.0x
±250.0	x	25.0x

Examples	**Command / Query**	**Response *(Description)***
	`INP:RANG 100,(@9:16)`	*(Selects an input range of ±100 V for Channels 9 - 16)*
	`INP:OFFS 2.5,(@9:16)`	*(Selects an input threshold of 25 V for Channels 9 - 16)*
	`INP:OFFS? 11`	2.500 *(Returns the set input threshold for Channel 11 of 25 V)*

Related Commands	INPut:RANGe INPut:POLarity SENSe:CONDition:LEVel

INPut:POLarity

Purpose	Sets the input polarity for one or more channels
Type	Setting
Command Syntax	INPut:POLarity <polarity>,<channel_list>
Command Parameters	<polarity> = NORMal | INVert <channel_list> = standard channel list syntax supporting channels 1 to 16
***RST Value**	NORMal for all channels
Query Syntax	INPut:POLarity? <channel>
Query Parameters	<channel> = 1 to 16
Query Response	ASCII string = NORM | INV
Description	The INPut:POLarity command selects the input polarity for one or more channels. When a channel is programmed for normal polarity, an interrupt will occur when the input voltage is greater than the programmed input offset for the channel. The invert polarity will cause an interrupt when the input voltage is less than the programmed input offset for the channel.

Examples	**Command / Query**	**Response *(Description)***
	`INP:POL INV,(@5:12)`	*(Inverts the input polarity for Channels 5 - 12)*
	`INP:POL? 6`	INV *(Indicates the polarity for Channel 6 is inverted)*

Related Commands	INPut:OFFset INPut:RANGe

INPut:RANGe

Purpose	Sets the input range for one or more channels	
Type	Setting	
Command Syntax	INPut:RANGe <range>,<channel_list>	
Command Parameters	<range> = ±1.0 | ±2.5 | ±10.0 | ±25.0 | ±100.0 | ±250.0 <channel_list> = standard channel list syntax supporting channels 1 to 16	
***RST Value**	250 V (all channels)	
Query Syntax	INPut:RANGe? <channel>	
Query Parameters	<channel> = 1 to 16	
Query Response	Numeric ASCII value = ±1.0 | ±2.5 | ±10.0 | ±25.0 | ±100.0 | ±250.0	
Description	The Input Range command selects the input range of one or more channels. The input range may be set for ±1.0 V, ±2.5 V, ±10.0 V, ±25.0 V, ±100.0 V and ±250.0 V.	
Examples	**Command / Query**	**Response *(Description)***
	`INP:RANG 100,(@1,3,5,7)`	*(Sets the input range for Channels 1, 3, 5 and 7 to ±100 V)*
	`INP:RANG? 7`	100 *(Returns the set input range for Channel 7)*
Related Commands	INPut:OFFset	

OUTPut:POLarity:EXTernal:INTerrupt

Purpose	Sets the interrupt output polarity on the front panel	
Type	Setting	
Command Syntax	OUTPut:POLarity:EXTernal:INTerrupt <polarity>	
Command Parameters	<polarity> = NORMal | INVert	
***RST Value**	NORMal	
Query Syntax	OUTPut:POLarity:EXTernal:INTerrupt?	
Query Parameters	N/A	
Query Response	ASCII string = NORM | INV	
Description	The OUTput:POLarity:EXTernal:INTerrupt command sets the polarity of the front panel interrupt output. When the polarity is set for normal, the output will be high when there is an interrupt event. When set for invert, the output will be low when there is an interrupt event.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:POL:EXT:INT NORM`	*(Sets the front panel interrupt output polarity to normal)*
	`OUTP:POL:EXT:INT?`	NORM *(Returns the set value for the front panel interrupt output polarity)*
Related Commands	None	

OUTPut:POLarity:EXTernal:LATChed

Purpose	Sets the latched interrupt output polarity on the front panel	
Type	Setting	
Command Syntax	OUTPut:POLarity:EXTernal:LATChed <polarity>	
Command Parameters	<polarity> = NORMal | INVert	
***RST Value**	NORMal	
Query Syntax	OUTPut:POLarity:EXTernal:LATChed?	
Query Parameters	N/A	
Query Response	ASCII string = NORM | INV	
Description	The OUTput:POLarity:EXTernal:LATChed command sets the polarity of the front panel latched interrupt output. When the polarity is set for normal, the output will be high when there is an interrupt event. When set for invert, the output will be low when there is an interrupt event.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:POL:EXT:LATC INV`	*(Sets the polarity of the front panel latched interrupt output to inverted)*
	`OUTP:POL:EXT:LATC?`	INV *(Returns the value for the front panel latched interrupt output)*
Related Commands	None	

OUTPut:TTLTrg

Purpose	This command selects the TTL trigger line that will be activated when the threshold is crossed	
Type	Setting	
Command Syntax	OUTPut:TTLTrg <line>	
Command Parameters	<line> = 0 – 7	
***RST Value**	0	
Query Syntax	OUTPut:TTLTrg?	
Query Parameters	N/A	
Query Response	Returns the set <line> value	
Description	This command selects the TTL trigger line that will be activated when the threshold is crossed.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:TTLT3` `OUTP:TTLT?`	*(Selects TTLT3 trigger line for output)* 3 (*Indicates that the TTLT3 trigger line is selected)*
Related Commands	OUTPut:TTLTrg:STATe	

OUTPut:TTLTrg:STATe

Purpose	This command enables/disabled the TTL trigger line outputs	
Type	Setting	
Command Syntax	OUTPut:TTLTrg:STATe <boolean>	
Command Parameters	<boolean> = 0 | 1 | OFF | ON	
***RST Value**	0	
Query Syntax	OUTPut:TTLTrg:STATe?	
Query Parameters	N/A	
Query Response	Returns the set <boolean> value	
Description	This enables/disables the TTL trigger line outputs. Setting the value to 0 or OFF disables the use of TTL trigger lines as outputs, while setting the value to 1 or ON enables it.	
Examples	**Command / Query**	**Response *(Description)***
	`OUTP:TTLT:STAT 1` `OUTP:TTLT:STAT?`	*(Enables the use of TTL trigger lines as outputs)* 1 (*Indicates that the TTL trigger lines are enabled as outputs)*
Related Commands	OUTPut:TTLTrg	

SENSe:CONDition:LEVel

Purpose	This command sets the threshold in volts	
Type	Setting	
Command Syntax	SENSe:CONDition:LEVel <voltage_level>,<channel_list>	
Command Parameters	<voltage_level > = -<range> to +<range> <channel_list> = standard channel list syntax supporting channels 1 to 16	
***RST Value**	0 V for all channels	
Query Syntax	SENSe:CONDition:LEVel? <channel>	
Query Parameters	<channel> = 1 to 16	
Query Response	Returns the set value of the <voltage_level> parameter	
Description	This command sets the threshold in volts. Unlike the INPut:OFFSet command, the SENSe:CONDition:LEVel command is not normalized to the ±10 V range and the value entered for the voltage is the actual voltage that will be used as the threshold.	
Examples	**Command / Query**	**Response *(Description)***
	`INP:RANG 100,(@1,3,5)`	*(Sets the input range to ±100 V)*
	`SENS:COND:LEV 25,(@1,3,5)`	*(Sets Channels 1, 3 and 5 to a 25 V threshold)*
	`SENS:COND:LEV? 3`	25 (*Indicates that Channel 3 is set to a 25 V threshold)*
Related Commands	INPut:OFFSet	

REQUIRED SCPI COMMANDS

STATus:OPERation:CONDition?

Purpose	Queries the Operation Status Condition Register	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation:CONDition?	
Query Parameters	None	
Query Response	0	
Description	The Operation Status Condition Register query is provided for SCPI compliance only. The VM4018 does not alter the state of any of the bits in this register and always reports a 0.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER:COND?`	0
Related Commands	None	

STATus:OPERation:ENABle

Purpose	Sets the Operation Status Enable Register	
Type	Required SCPI command	
Command Syntax	STATus:OPERation:ENABle <NRf>	
Command Parameters	<NRf> = numeric ASCII value from 0 to 32767	
***RST Value**	NRf must be specified	
Query Syntax	STATus:OPERation:ENABle?	
Query Parameters	None	
Query Response	Numeric ASCII value from 0 to 32767	
Description	The Operation Status Enable Register is included for SCPI compatibility and the VM4018 does not alter any of the bits in this register. The register layout is as follows: Bit 0 - Calibrating Bit 1 - Setting Bit 2 - Ranging Bit 3 - Sweeping Bit 4 - Measuring Bit 5 - Waiting for trigger Bit 6 - Waiting for arm Bit 7 - Correcting	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER:ENAB 0`	0
Related Commands	None	

STATus:OPERation[:EVENt]?

Purpose	Queries the Operation Status Event Register	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:OPERation [:EVENt]?	
Query Parameters	None	
Query Response	0	
Description	The Status Operation Event Register query is included for SCPI compliance. The VM4018 does not alter any of the bits in this register and always reports a 0.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:OPER?`	
Related Commands	None	

STATus:PRESet

Purpose	Presets the Status Registers	
Type	Required SCPI command	
Command Syntax	STATus:PRESet	
Command Parameters	None	
***RST Value**	N/A	
Query Syntax	None – command only	
Query Parameters	N/A	
Query Response	N/A	
Description	The Status Preset command presets the Status Registers. The Operational Status Enable Register is set to 0 and the Questionable Status Enable Register is set to 0. This command is provided for SCPI compliance only.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:PRES`	*(Presets the Status Registers)*
Related Commands	None	

STATus:QUEStionable:CONDition?

Purpose	Queries the Questionable Status Condition Register	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:QUEStionable:CONDition?	
Query Parameters	None	
Query Response	0	
Description	The Questionable Status Condition Register query is provided for SCPI compliance only. The VM4018 does not alter any of the bits in this register and a query always reports a 0.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES:COND?`	0
Related Commands	None	

STATus:QUEStionable:ENABle

Purpose	Sets the Questionable Status Enable Register	
Type	Required SCPI command	
Command Syntax	STATus:QUEStionable:ENABle <NRf>	
Command Parameters	<NRf> = numeric ASCII value from 0 to 32767	
***RST Value**	NRf must be supplied	
Query Syntax	STATus:QUEStionable:ENABle?	
Query Parameters	None	
Query Response	Numeric ASCII value from 0 to 32767	
Description	The Status Questionable Enable command sets the bits in the Questionable Status Enable Register. This command is provided only to comply with the SCPI standard. The Status Questionable Enable query reports the contents of the Questionable Status Enable Register. The VM4018 does not alter the bit settings of this register and will report the last programmed value.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES:ENAB 64` `STAT:QUES:ENAB?`	64
Related Commands	None	

STATus:QUEStionable[:EVENt]

Purpose	Queries the Questionable Status Event Register	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	STATus:QUEStionable [:EVENt]?	
Query Parameters	None	
Query Response	0	
Description	The Questionable Status Event Register is provided for SCPI compliance only. The VM4018 does not alter the bits in this register and queries always report a 0.	
Examples	**Command / Query**	**Response *(Description)***
	`STAT:QUES?`	0
Related Commands	None	

SYSTem:ERRor?

Purpose	Queries the Error Queue	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	SYSTem:ERRor?	
Query Parameters	None	
Query Response	ASCII string	
Description	The System Error query is used to retrieve error messages from the error queue. The error queue will maintain the two error messages. If additional errors occur, the queue will overflow and the second and subsequent error messages will be lost. In the case of an overflow, an overflow message will replace the second error message. See the SCPI standard Volume 2: Command Reference for details on errors and reporting them. Refer to the "Error Messages" section of this manual for specific details regarding the reported errors.	
Examples	**Command / Query**	**Response *(Description)***
	`SYST:ERR?`	-350, "Queue overflow"
Related Commands	None	

SYSTem:VERSion?

Purpose	Queries the SCPI version number to which the VM4018 complies	
Type	Required SCPI command	
Command Syntax	None – query only	
Command Parameters	N/A	
***RST Value**	N/A	
Query Syntax	SYSTem:VERSion?	
Query Parameters	None	
Query Response	Numeric ASCII value	
Description	The System Version query reports version of the SCPI standard to which the VM4018 complies.	
Examples	**Command / Query**	**Response *(Description)***
	`SYST:VERS?`	1994.0
Related Commands	None	

INDEX