User Manual

Tektronix

VX4570
Mass Storage Controller

070-9259-01

This document supports firmware version 1.00 and above.

Warning
The servicing instructions are for use by qualified personnel only. To avoid personal injury, do not perform any servicing unless you are qualified to do so. Refer to the Safety Summary prior to performing service.

Printed in the U.S.A.

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

Getting Started

Operating Basics

Syntax and Commands

Status and Events

Appendices

Glossary and Index

General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

Only qualified personnel should perform service procedures.

While using this product, you may need to access other parts of the system. Read the *General Safety Summary* in other system manuals for warnings and cautions related to operating the system.

Injury Precautions

Avoid Electric Overload

To avoid electric shock or fire hazard, do not apply a voltage to a terminal that is outside the range specified for that terminal.

Do Not Operate Without Covers

To avoid electric shock or fire hazard, do not operate this product with covers or panels removed.

Use Proper Fuse

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

Do Not Operate in Wet/Damp Conditions

To avoid electric shock, do not operate this product in wet or damp conditions.

Do Not Operate in Explosive Atmosphere

To avoid injury or fire hazard, do not operate this product in an explosive atmosphere.

Product Damage Precautions

Provide Proper Ventilation

To prevent product overheating, provide proper ventilation.

Do Not Operate With Suspected Failures

If you suspect there is damage to this product, have it inspected by qualified service personnel.

Safety Terms and Symbols

Terms in This Manual

These terms may appear in this manual:

⚠ ***WARNING.*** *Warning statements identify conditions or practices that could result in injury or loss of life.*

⚠ ***CAUTION.*** *Caution statements identify conditions or practices that could result in damage to this product or other property.*

Terms on the Product

These terms may appear on the product:

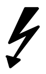
DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

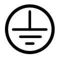
CAUTION indicates a hazard to property including the product.

Symbols on the Product

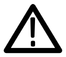
The following symbols may appear on the product:



DANGER
High Voltage



Protective Ground
(Earth) Terminal



ATTENTION
Refer to
Manual

Double
Insulated

Service Safety Summary

Only qualified personnel should perform service procedures. Read this *Service Safety Summary* and the *General Safety Summary* before performing any service procedures.

Do Not Service Alone

Do not perform internal service or adjustments of this product unless another person capable of rendering first aid and resuscitation is present.

Disconnect Power

To avoid electric shock, disconnect the main power by means of the power cord or, if provided, the power switch.

Use Care When Servicing With Power On

Dangerous voltages or currents may exist in this product. Disconnect power, remove battery (if applicable), and disconnect test leads before removing protective panels, soldering, or replacing components.

To avoid electric shock, do not touch exposed connections.

Preface

This manual assumes you are familiar with VXIbus instruments and operation and with the purpose and function of this instrument.

Please read and follow all instructions for installation and configuration. Use the Installation Checklist to ensure proper installation and to record your initial settings.

The *Operating Basics* section gives a summary of VXIbus operation and presents an overview of the operation of this instrument.

The *Syntax and Commands* section provides a summary of all the commands followed by detailed descriptions of each command, including examples.

The *Status and Events* section contains an explanation of the Status and Event Reporting System and lists the system messages.

You may also want to keep the *Reference* guide by the instrument.

Conventions

The names of all switches, controls, and indicators appear in this manual exactly as they appear on the instrument.

Specific conventions for programming are given in *Syntax and Commands*.

Getting Started

Getting Started

This section begins with a brief description of the VX4570 Mass Storage Controller Module, and explains how to configure and install the module in a VXIbus mainframe.

Product Description

The VX4750 is a single-wide C-size VXI module that provides an interface between the VXIbus and a mass storage device. The VX4570 is designed for use in a data acquisition system where VXI modules are collecting large amounts of data to be off-loaded to mass storage for later retrieval and post-processing analysis.

In such a data acquisition system the VX4570 will be a VXI commander. It moves data from its data collector servants over the VXIbus (using Fast Data Channel protocol) into shared memory on the VX4570. The VX4570 will then move the data from shared memory to the mass storage device.

Data from the data collector is written out to a data acquisition file on the mass storage device with no formatting or modification. Before storage, a header is added to each FDC buffer received from the data collector. The header includes the FDC channel number, buffer size, and other information for post processing. A setup file containing the data collector's setup parameters is associated with each data acquisition file. Information in the setup file also aids in post-processing data extraction and analysis.

Refer to the *Operating Basics* for additional information on the header and setup file.

The interface to the mass storage device is implemented on a daughter board. The daughter board is a SCSI II single-ended interface with a high density 50-pin connector.

Figure1–1 shows a functional block diagram of the module. Figure 1–2 shows the VX4570 controls and indicators, and Figure 1–3 shows the front panel.

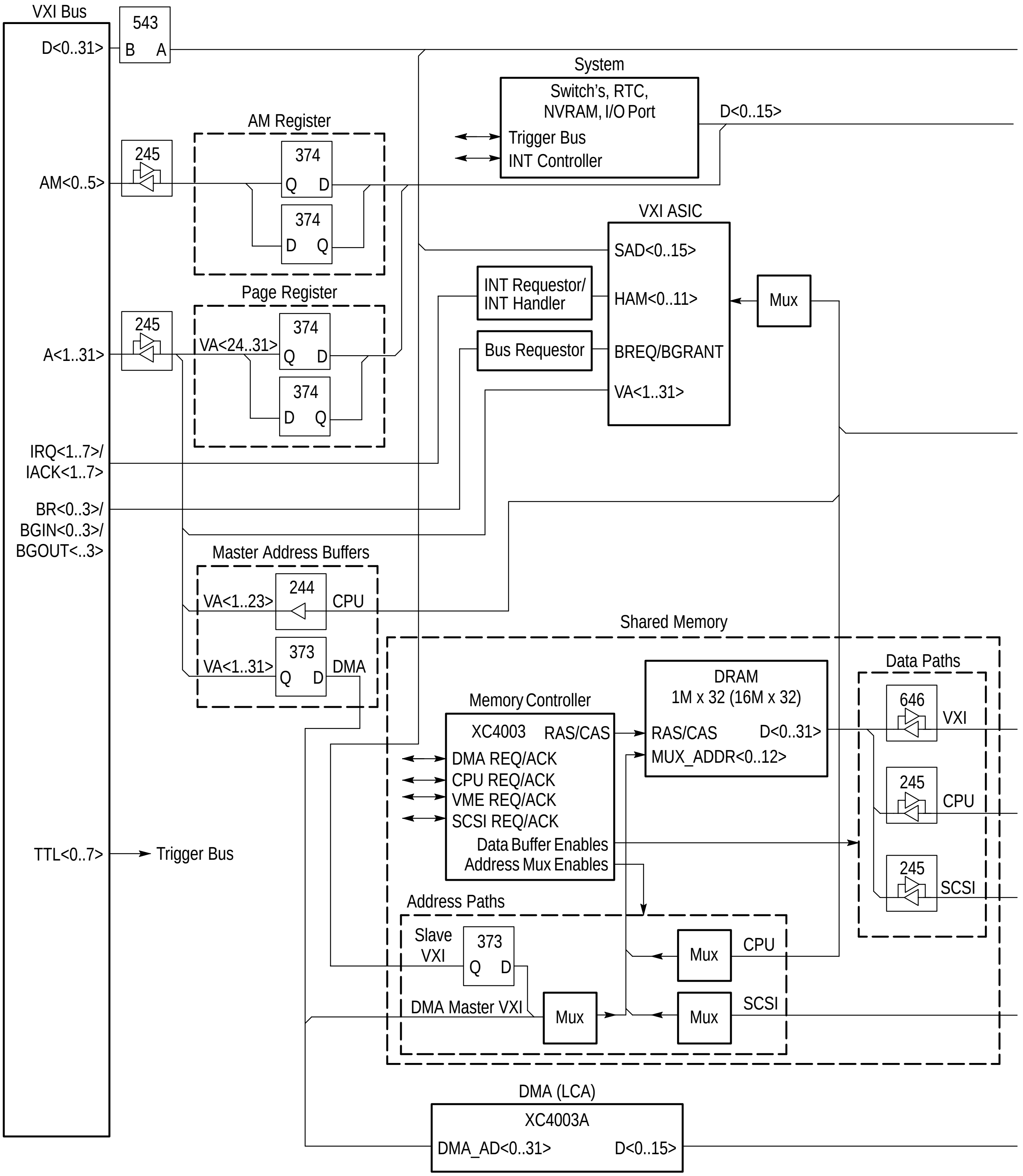


Figure 1–1: VX4570 Block Diagram

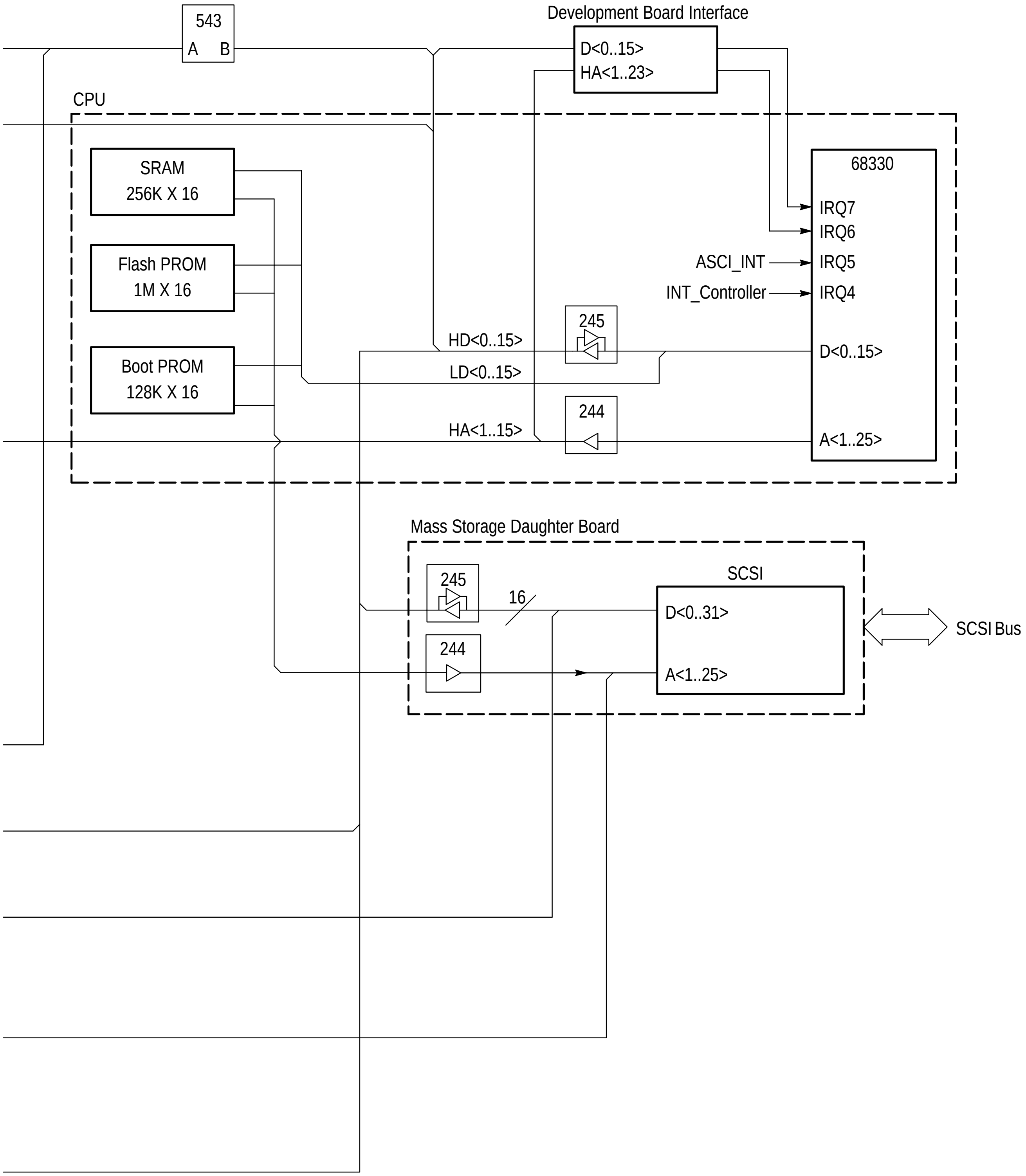


Figure 1–1 (continued): VX4570 Block Diagram

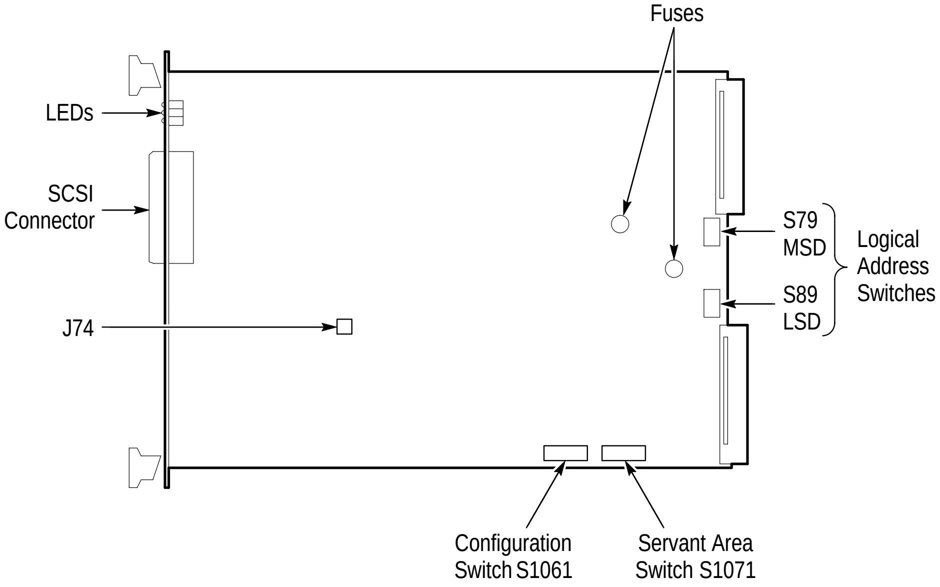


Figure 1–2: VX4570 Controls and Indicators

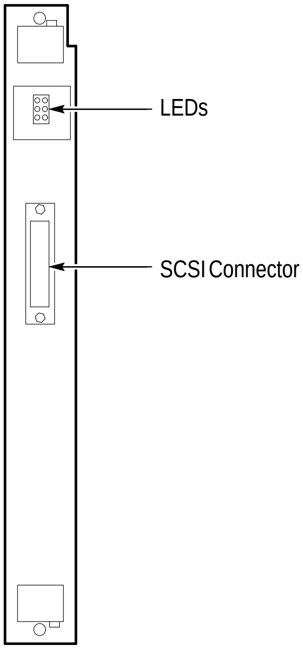


Figure 1–3: VX4570 Front Panel

Fuses

The VX4570 has a fuse that limits the amount of current that the module can draw from the VXI backplane +5 V power pins. This fuse protects the module in case of an accidental shorting of the power bus or any other situation where excessive current might be drawn.

If the fuse opens, the VX4570 will assert SYSFAIL* on the VXIbus.

If the +5 V fuse opens, the VXIbus Resource Manager will be unable to assert SYSFAIL INHIBIT to disable SYSFAIL*.

If a +5 V fuse opens, remove the fault before replacing the fuse. Replacement fuse information is given in the *Appendix A:Specifications* section of the user manual for the appropriate relay module.

BITE (Built-in Test Equipment)

Built-in Test Equipment (BITE) is provided by extensive self tests that are automatically invoked on power-on and can also be invoked on command. Circuitry tested includes the CPU, SRAM, DRAM, non-volatile RAM, real-time clock, SCSI chip registers, interrupt controller registers, address modifier register, and page register.

Accessories

Table 1–1 lists the standard accessories included with the VX4570.

Table 1–1: Standard Accessories

Accessory	Part Number
VX4570 User Manual	070-9259-XX
VX4570 Reference	070-9295-XX

Controls and Indicators

The following controls are provided to select the functions of the VX4570 operating environment.

Switches

The Logical Address switches and Configuration switches must be correctly set to insure proper operation. See *Configuration* for details on how to set the switches.

Configuration

The following switches must be correctly set to ensure proper operation. Refer to Figure 1–2 for their physical locations.

Logical Address Switches

Each functional module in a VXIbus System must be assigned a unique logical address, from 1 to decimal 255 (hexadecimal FF). The base VMEbus address of the VX4570 is set to a value between hexadecimal C0 ($C0_{16}$) and hexadecimal FF (FF_{16}) by two hexadecimal rotary switches. Align the desired switch position with the arrow on the module shield.

The physical address of the instrument is on a 64-byte boundary. If the Logical Address switch representing the most significant digit (LA–HI) of the logical address is set to position X and the switch representing the least significant digit (LA–LO) of the logical address is set to position Y, then the base physical address of the VX4570 will be $[(40_{16} \times XY_{16}) + C000_{16}]$. Table 1–2 shows two Logical Address switch examples.

NOTE*. When using the VX4570 as a master for the VX4244 Digitizer, you must set the logical address for these servants to be greater than the logical address for the VX4570. For convenience, it is recommended that the logical addresses be set in ascending order: For example, set the VX4570 at logical address 10 and set the VX4244 at logical address 11. Please read the readme.txt file that is included with the System Management and Control Software (SMCS) for further clarification.*

Table 1–2: Logical Address Switch Examples

L.A.[1]	HI	LO	Base Physical Address	
			decimal	hexadecimal
A_{16}	0_{16}	A_{16}	(64 * 10) + 49152 = 49792	$(40_{16} * A_{16}) + C000_{16} = C280_{16}$
15_{16}	1_{16}	5_{16}	(64 * 21) + 49152 = 50496	$(40_{16} * 15_{16}) + C000_{16} = C540_{16}$

[1] L.A. is the Logical Address

IEEE-488 Address

NOTE*. At the time of printing for this manual, the VX4570 can only be used with embedded controllers or MXI slot 0 modules. Please contact your local Tektronix representative before attempting to use this product with an IEEE–488 slot 0 module.*

Configuration Switch

The Configuration switch, S1061 has the following functions:

S1061 rockers 1 and 2 – VMEbus Arbitration Level. These two rockers determine which Bus Request level will be used by the VX4570 when it requires use of the VMEbus. The setting of rockers 1 and 2 for the required Bus Request levels is as follows:

Rocker 1	Rocker 2	Level
closed (down)	closed (down)	0
open (up)	closed (down)	1
closed (down)	open (up)	2
open (up)	open (up)	3

S1061 rocker 3 – Boundary Scan mode. In the closed (down) position the VX4570 is in Boundary Scan mode and will NOT operate as a VXI instrument. This rocker must be in the open (up) position for normal operation of the VX4570.

S1061 rockers 4 through 8. Unused.

Servant Area Switch

The Servant Area switch, S1071 has the following functions:

S1071 rockers 1 through 4 – Servant Area Size. These four rockers select how many data collector servants will be granted to the VX4570 when the VXI Resource Manager configures the system hierarchy. These four rockers use binary values. Rocker 1 is the LS bit and rocker 4 is the MS bit. For a servant area size of 5 the rockers would be:

Rocker	Position
1	open (up)
2	closed (down)
3	open (up)
4	closed (down)

For this switch, up = 1 and down = 0.

S1071 rockers 5 through 8. Unused.

LEDs

The following LEDs are provided on the front panel:

LED	Description
POWER (green)	When lighted, indicates all DC power for the card is available. Unlighted indicates loss of a power rail(s).
FAILED (red)	Lighted whenever SYSFAIL* is asserted, indicating a module failure. Module failure includes loss of power rail, CPU failure, or a catastrophic self-test failure.
MASTER (green)	When lighted, indicates the module is performing a master cycle on the VMEbus.
ACCESS (yellow)	When lighted, indicates the module is being addressed as a slave on the VMEbus.
FETCH (green)	When lighted, indicates the module is accessing the SCSI bus.
ERROR (red)	When lighted, indicates a programming error has been processed by the card.

Front Panel Connector

See Figure 2 for the physical location of this connector. The front panel has a 50-pin high density standard SCSI–2 connector. The connector type is “shielded alternative 1” in the SCSI–2 specification. Refer to *Appendix B* for connector pinouts.

Installation

This section describes how to install the VX4570.

Tools Required

A slotted screwdriver set is required for proper installation.

Requirements and Cautions

The VX4570 Controller is a C-size VXIbus instrument module and therefore may be installed in any C- or D-size VXIbus mainframe slot other than slot 0. To install the module in a D-size mainframe, consult the operating manual for the mainframe. Refer to *Configuration* for information on selecting and setting the Logical Address switch of the module. This switch defines the programming address of your module. To avoid confusion, it is recommended that the slot number and the logical address be the same.

NOTE*. Note that there are two printed ejector handles on the card. To avoid installing the card incorrectly, make sure the ejector marked “VX4570” is at the top.*

***NOTE**. Verify that the mainframe is able to provide adequate cooling and power with this module installed. Refer to the mainframe Operating Manual for instructions.*

If the VX4570 is used in a Tektronix mainframe, all VX4570 cooling requirements are met.

***NOTE**. If the VX4570 is inserted in a slot with any empty slots to the left of the module, the VME daisy-chain jumpers must be installed on the backplane in order for the VXI Module to operate properly. Check the manual of the mainframe being used for jumpering instructions. Jumpers are not necessary for autoconfiguring backplane designs.*

Installation Procedure

Follow these steps to install the VX4570.

⚠ ***CAUTION.** The VX4570 Controller is a piece of electronic equipment and therefore has some susceptibility to electrostatic damage (ESD). To avoid ESD, use precautions when handling the module.*

1. Record the revision levels, serial numbers (located on the label on the top shield of the host module and on a sticker on the VX4570), and switch settings on the *Installation Checklist*.
2. Verify that the switches are set to the correct values. Refer to *Configuration* for more information on setting switches.
3. Make sure that the mainframe power is off.
4. Insert the module into one of the instrument slots of the mainframe (see Figure 1–4).

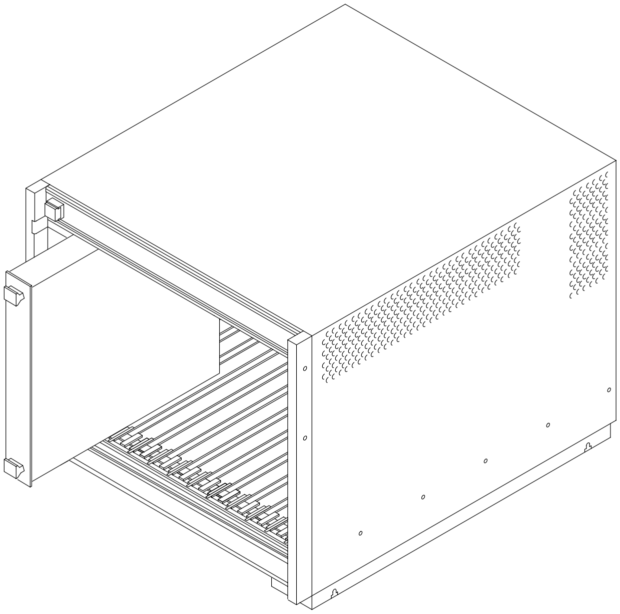


Figure 1–4: Module Installation

Installation Checklist

Installation parameters will vary depending on the mainframe being used. Be sure to consult the mainframe operating manual before installing and operating the module.

Revision Level: __________

Serial No.: __________

Mainframe Slot Number: __________

Switch Settings:

VXIbus Logical Address Switch: __________

Configuration Switches:

S1061 VMEbus Arbitration Level:

Rocker 1: __________

Rocker 2: __________

Rocker 3: __________

(Rockers 4–8 unused)

S1071 Servant Area Size:

Rocker 1: __________

Rocker 2: __________

Rocker 3: __________

Rocker 4: __________

(Rockers 5–8 unused)

Cable Installed (if any): __________

Performed by: ______________________ Date: ____________

Functional Check

The VX4570 Mass Storage Controller Module executes a self test at power-on, on direction of a VXIbus hard or soft reset condition, or on command. The power-on self test consists of an interface self test and an instrument self test. The self test requested by command performs only the instrument self test. A VXIbus hard reset occurs when another device, such as the VXIbus Resource Manager, asserts the backplane line SYSRESET*. A VXIbus soft reset occurs when another device, such as the VX4570's commander, sets the Reset bit in the VX4570's Control register.

During power-on, or a hard or soft reset, the following actions take place:

1. The SYSFAIL* (VME system failure) line is set active, indicating that the module is executing a self test, and the Failed LED is lighted. For a soft reset, SYSFAIL* is set. All Tektronix commanders will simultaneously set SYSFAIL INHIBIT to prevent the resource manager from prematurely reporting the failure of a card.
2. The instrument self test, as described in the *TST? command description is then executed. If the self test fails, the module makes an internal record of the failure(s) that occur. Use the SYSTem:ERRor? command to view the results.
3. On completion of the interface self test, SYSFAIL* is de-asserted. If the test fails, the SYSFAIL* line remains active. If the interface self test passed, the SYSFAIL* line is released, and the module enters the VXIbus PASSED state (ready for normal operation). If it failed, the module enters the VXIbus FAILED state.

After self test, the module is returned to its default state, as described in the *RST command description.

Self test can be run at any time during normal operation by using the *TST? command.

In addition to the self test, more comprehensive and additional tests can be executed using the module's DIAGnostic commands described in *Appendix E*.

Operational Check

SYSFAIL* becomes active during power-on, hard or soft reset, or self test, or if the module loses +5 V power. When the mainframe Resource Manager detects SYSFAIL* set, it will attempt to inhibit the line. This causes the VX4570 Mass Storage Controller Module to deactivate SYSFAIL* except when +5 V power is lost.

Operating Basics

Operating Basics

The VX4570 Mass Storage Controller Module is a VXIbus message-based instrument and communicates using the VXIbus Word Serial Protocol. The module is programmed by issuing ASCII characters from the system controller to the VX4570 via the module VXIbus commander and the VXIbus mainframe backplane. Refer to the manual for the VXIbus device that will be the VX4570 Controller's commander for details on the operation of that device.

Functional Overview

The VX4750 is a single-wide C-size VXI module that provides an interface between the VXIbus and a mass storage device. The VX4570 is primarily designed for use in a data acquisition system where VXI modules are collecting large amounts of data to be off-loaded to mass storage for later retrieval and post processing analysis.

In such a data acquisition system the VX4570 will be a VXI commander. It moves data from its data collector servants over the VXIbus (using Fast Data Channel protocol) into shared memory on the VX4570. The VX4570 will then move the data from shared memory to the mass storage device.

Data from the data collector is written out to a data acquisition file on the mass storage device with no formatting or modification. Before storage, a header is added to each FDC buffer received from the data collector. The header includes FDC channel number, buffer size, and other information for post processing. A setup file containing the data collector's setup parameters is associated with each data acquisition file. Information in the setup file also aids in post-processing data extraction and analysis.

The interface to the mass storage device is implemented on a daughter board. The daughter board is a SCSI II single-ended interface with a high density 50-pin connector.

Power-On

The VX4570 will complete its self test and be ready for programming five seconds after power-on. The POWER LED will be on, and all other LEDs off. The yellow ACCESS LED will blink during the power-on sequence as the VXIbus Resource Manager addresses all modules in the mainframe.

When the VX4570 receives the VXI Begin Normal Operation (BNO) command, the green FETCH LED will be lighted as initialization of the SCSI bus begins.

The green MASTR LED will also be lighted if the VX4570 has been assigned and granted servants by the VXIbus Resource Manager.

Instrument I/O: VXIbus Basics

CAUTION. *If the user's mainframe has other manufacturers' computer boards operating in the role of VXIbus foreign devices, the assertion of BERR* (as defined by the VXIbus Specification) may cause operating problems on these boards.*

A Normal Transfer Mode read of the VX4570 Controller proceeds as follows:

1. The commander reads the VX4570 Response register and checks if the Write Ready and DOR bits are true. If they are, the commander proceeds to the next step. If not, the commander continues to poll these bits until they become true.

2. The commander writes the Byte Request command (hexadecimal 0DEFF) to the Data Low register of the VX4570.

3. The commander reads the VX4570 Response register and checks if the Read Ready and DOR bits are true. If they are, the commander proceeds to the next step. If not, the commander continues to poll these bits until they become true.

4. The commander reads the VX4570 Data Low register.

A Normal Transfer Mode write to the VX4570 Controller proceeds as follows:

1. The commander reads the VX4570 Response register and checks if the Write Ready and DIR bits are true. If they are, the commander proceeds to the next step. If not, the commander continues to poll the Write Ready and DIR bits until they are true.

2. The commander writes the Byte Available command which contains the data (hexadecimal 0BCXX or 0BDXX, depending on the End bit) to the Data Low register of the VX4570.

The module has no registers beyond those defined for VXIbus message based devices. All WSP communications with the module are through the Data Low register, the Response register, or the VXIbus interrupt cycle. Any attempt by another module to read or write to any undefined location of the VX4570 address space may cause incorrect operation of the module.

As with all VXIbus devices, the VX4570 Controller has registers located within a 64 byte block in the A16 address space. The base address of the VX4570 device registers is determined by the device unique logical address and can be calculated as follows:

Base Address = $V_{16} * 40_{16} + C000_{16}$

where V is the logical address of the device as set by the Logical Address switches.

VX4570 Configuration Registers

Table 2–1 contains a list of the VX4570 Configuration registers and a complete description of each register. The offset is relative to the module base address.

Table 2–1: Register Definitions

Register	Address (hexadecimal)	Type	Value (Bits 15–0)
ID Register	0000	RO	1001 1111 1111 1101 (hexadecimal 9FFD)
Device Type	0002	RO	1001 0101 1100 0101 (hexadecimal 95C5)
Status	0004	R	Defined by the state of the interface
Control	0004	W	Defined by the state of the interface
Offset	0006	WO	Assigned by the Resource Manager
Protocol	0008	RO	0001 1111 1111 1111 (hexadecimal 1FFF)
Response	000A	RO	Defined by the state of the interface
Data High	000C		Not used
Data Low	000E	W	Not fixed; command-dependent
Data Low	000E	R	Not fixed; command-dependent

RO is Read Only

WO is Write Only

R is Read

W is Write

Syntax and Commands

Command Syntax

Command protocol and syntax for the VX4570 Controller are as follows:

- A command string consists of a string of ASCII-encoded characters terminated by a <program message terminator>. The <program message terminator> is optional white space, followed by any one of the following command terminations:

  a line feed <LF> character (hexadecimal 0A, decimal 10)

  the END bit set

  the END bit with a line feed <LF>

  The command string is buffered until the terminator is encountered, at which time the entire string is processed.

- Multiple commands in a single command string are separated by a semi-colon (;) character. In addition to terminating a command, the semi-colon character directs the SCPI command parser to interpret the next command with the assumption that all characters up to and including the last semi-colon in the previous command have just been parsed.

- White space characters can be used to make a command string more readable. These characters are ASCII-encoded bytes in the range hexadecimal 00–09 and 0B–20 (decimal 0–9 and 11–32). This range includes the ASCII control characters and the space, but excludes the line feed <LF>. White space characters are ignored when legally encountered in a command string. White space is allowed anywhere in a command string, except for the following:

  Within a program mnemonic (for example MMEM:CATA LOG?)

  Around a colon (:) mnemonic separator (for example FDC: OPEN or FDC :OPEN)

  Between a mnemonic and a (?) (for example MMEM:CATALOG?)

  Following an asterisk (*) (for example * STB?)

  Within a number (for example 12 34)

  At least one white space character is required between a command/query header and its associated arguments. For example in the command

  ```
  VXI:FDC:OPEN 5,0
  ```

the command header is the string “VXI:FDC:OPEN” The arguments associated with this command are the Logical Address “5” and the channel number “0”. At least one white space character must be sent before the first argument.

- All characters in a command can be sent in either upper or lower case form.
- Multiple data parameters passed by a command are separated by a comma (,).
- A question mark (?) following a command indicates that a response will be returned. All responses from the module are terminated with the line feed <LF> (hexadecimal 0a) character.
- In the command descriptions, the following special characters are used. Except for the colon (:), these characters are not part of the command and should not be sent. If an optional field is omitted, the default for the command is applied.

[ ]	Brackets indicate an optional field
:	A colon is used to separate command fields
< >	Field indicator

Syntax Example

Capital letters indicate the minimum required characters which need to be sent. Lower case letters are optional. For example, the syntax of the catalog query is given as:

```
MMEMory:CATalog? "sd6"
```

Each of the following is a valid form of this command:

```
mmem:catalog? "sd6"

MMEM:Catalog? "SD6"

Mmem:Catalog? "SD6"

MMEM:CAT? "SD6"

mmem:cat? "SD6"

MMEMORY:Catalog? "SD6"
```

Functional Command Groups

This section lists the VX4570 system and module commands.

System Commands

The following low-level commands are typically sent by the module's commander, and are transparent to the user of the module. (An exception is the Read STB command, which is sent whenever a Serial Poll on an IEEE-488 system is performed.) Most commanders or Slot 0 devices have specific ASCII commands that cause the commander to send one of these low-level commands to a specified instrument. Refer to the operating manual of the commander or Slot 0 device for information on these commands.

Command	Effect
Assign Handler	Assigns VME interrupt line this module will monitor.
Assign Interrupter	Assigns VME interrupt line this module will assert.
Begin Normal Operation	The module begins operation if it has not already done so.
Byte Available	Transfers module commands to this module.
Byte Request	Requests the module to return a byte of data from the output queue.
Clear	The module clears its VXIbus interface and any pending commands. Current module operations are unaffected.
Read Protocol	The module returns its protocol to its commander.
Read STB	The module returns its VXI status byte to its commander.
Assign Handler	Assigns VME interrupt line this module will monitor.
Assign Interrupter	Assigns VME interrupt line this module will assert.

Module Commands

A summary of the VX4570-specific and IEEE-488.2 Common Commands is listed below. The next section, *Command Descriptions*, includes detailed descriptions of each command.

Table 3–1: Summary of VX4570-Specific Required Common SCPI Commands

Command	Functions Controlled
STATus	Status register functions: OPERation, QUEStionable, PREset
SYSTem	System-level functions: ERRor?, VERsion?

Table 3–2: Summary of VX4570-Specific Established SCPI Commands

Command	Functions Controlled
MMEMory	Mass memory functions: MSIS, CATalog?, NAME, OPEN, CLOSE, LOAD, STORe, DELete, INITialize, PACK, COPY, MOVE, FEED
VXI	SELect, CONFigure:LADDress?, READ?

Table 3–3: Summary of VX4570-Specific Proposed SCPI Commands

Command	Functions Controlled
VXI:COMMander	Commander FDC control: SELect?, OPEN, READ, WRITe, CLOSe, IDLE, MODE?, TYPE?
VXI:SERVant	Servant FDC control: SELect, OPEN, READ, WRITe, CLOSe
MEMory	ALLOcate, FREE, SIZE?

Table 3–4: Summary of IEEE-488.2 Common Commands

Command/Syntax	Description
Clear Status *CLS	Clears the SCPI and IEEE 488.2 event registers and the SCPI error/event queue, and the output queue.
Standard Event Status Enable *ESE <NRf>	Sets the contents of the Standard Event Status Enable register.
Standard Event Status Enable Query *ESE?	Returns the current value of the Standard Event Status Enable register in <nr1> format.
Standard Event Status Register Query *ESR?	Returns the current value of the Standard Event Status register in <nr1> format, then set the contents of this register to 0.
Identification Query *IDN?	Returns an ASCII string in the output queue which identifies the board.

Table 3–4: Summary of IEEE-488.2 Common Commands (Cont.)

Command/Syntax	Description
Operation Complete *OPC	Sets bit 0 (the Operation Complete bit) of the Standard Event Status register when all pending device operations have been completed.
Operation Complete Query *OPC?	Places the ASCII character 1 in the output queue when all pending device operations have been completed.
Reset *RST	Places the module in its power-on state with some exceptions.
Service Request Enable *SRE <NRf>	Sets the contents of the Service Request Enable register.
Service Request Enable Query *SRE?	Returns the current value of the Service Request Enable register in <nr1> format.
Read Status Byte Query *STB?	Returns the current value of the Status Byte register in <nr1> format.
Self-Test Query *TST?	Executes the self test.
Wait-to-Continue *WAI	Does not execute any further commands or queries until all pending operations have been completed.

Command Descriptions

This section lists the VX4570-specific SCPI commands and queries in alphabetic order.

Definitions

The command descriptions in this section use terms and abbreviations as defined below:

string value	IEEE-488.2 <string program data>
msus	string value (example: “SD1”)
file_name	string value; maximum of 12 characters, DOS style “yyyyyyyy.xxx”
src-file	<file_name> | <file_name>,<msus>
dest_file	<file_name> | <file_name>,<msus>
log_addr	numeric value 0 to 255, truncated to integer.
chan_num	numeric value 0 to 7, truncated to nearest integer.
data_handle	enumerated string values
N/A	not applicable

MEMory

Command Syntax

```
MEMory
   :ALLOcate <buffer_size>

   :FREE
```

Query Syntax

```
MEMory
   :SIZE?
```

Query Response

This query returns the number and size of the Shared Memory buffers used in FDC data transfers.

“Buffer size = buffer_size, Buffer count = buffer_count”
buffer_size is an integer value from 32768 to 524288.
buffer_count is an integer value from 126 to 7.

*RST Value

Buffer size and buffer count both equal zero (no Shared Memory buffers have been allocated).

Limits

N/A

Related Commands

N/A

Description

The MEMory:ALLOcate <buffer_size> command allocates buffers from Shared Memory for use in FDC transfers. It is the size of the buffer into which data will be FDC transferred from its source. The buffer_size parameter specifies the number of Kbytes of buffer memory. Valid values range from 32 (representing 32768 bytes) to 512 (representing 524288 bytes). A buffer size that is smaller than the FDC transfer size will cause unpredictable system behavior. A buffer size too big will result in less than optimal system performance. Ideally, the buffer size should be equal to the FDC transfer size. This allows the maximum number of buffers in memory, minimizing memory pool overhead.

The MEMory:FREE command deallocates buffers.

Examples

Command	Response
MEM:ALLO 32	Allocate a 32 Kbyte buffer.
MEM:SIZE?	“Buffer size = 32768, Buffer count = 126”
MEM:FREE	Deallocate the buffers.

MMEMory:CATalog

Command Syntax

```
MMEMory:CATalog? [<msus>]
```

Query Response

```
<numeric_value>,<numeric_value>{,<file_entry>}
```

***RST Value** N/A

Limits N/A

Related Commands N/A

Description

This query returns information on the contents and state of mass storage media.

The command returns two numeric values followed by as many strings as there are files in the directory list. The format is as follows:

```
<numeric_value>,<numeric_value>{,<file_entry>}
```

The first numeric value indicates the amount of used media space (in bytes).

The second numeric value indicates the total storage space available on the media (in bytes).

File entries are strings indicating the name, type, and size of the file:

```
<file_name>,<file_type>,<file_size>
```

The file_name is as it appears in the directory list, and the file size is in bytes. The file_type parameter is not used at this time.

Examples

Command	Response
MMEM:CAT?	9342464, 21052416, data.dat, , 9307532

The returned string indicates that 9,342,464 bytes of media space have been used out of a total media storage space available of 21,052,416 bytes, that there is a single file with the file name data.dat, and it is 9,307,532 bytes.

MMEMory:CLOSe

Command Syntax

```
MMEMory:CLOSe
```

Query Syntax

N/A

***RST Value**

N/A

Limits

N/A

Related Commands

N/A

Description

The CLOSe command closes the currently NAMEd file. Once closed, a file must be NAMEd and OPENed before it can be accessed.

Examples

Command	Response
MMEM:CLOSE	closes the currently NAMEd file

MMEMory:COPY

Command Syntax

```
MMEMory:COPY <src_file>,<dest_file>
MMEMory:COPY <src_file>,[<src_msus>],<dest_file>,[<dest_msus>]
```

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands MMEMory:MOVE

Description This command copies an existing file to a new file.

Two forms of the command are allowed. The first form has two parameters. In this form the first parameter, <src_file>, specifies the file to be copied, and the second parameter, <dest_file>, specifies the new file name. The copy operation is performed on the default mass storage device.

The second form has four parameters. The first two parameters, <src_file> and <src_msus>, specify the file name and storage device of the source file. The second two parameters, <dest_file> and <dest_msus>, specify the new file name and new storage device for the destination file.

If the <src_file> does not exist, the instrument generates error –256 “Mass Storage Error: File Name Not Found”. If the <dest_file> already exists, the instrument generates error –257, “Mass Storage File Name Error: File Name already exists”.

Examples

Command	Response
MMEM:COPY ”file1.dat”, ”file2.dat”	Copies the contents of file1.dat to a new file named file2.dat. Both files are on the default mass storage device.

MMEMory:DELete

Command Syntax

```
MMEMory:DELete <file_name>[,<msus>]
```

Query Syntax

N/A

***RST Value**

N/A

Limits

N/A

Related Commands

MMEMory:NAME

Description

The DELete command removes a file from the specified mass storage. The <file_name> parameter specifies the file name to be removed. An attempt to DELete a file that does not exist generates error –256 “Mass Storage Error: File Name Not Found”.

Examples

Command	Response
MMEM:DEL “file1.dat”	Removes the file “file1.dat” from the specified mass storage.

MMEMory:FEED

Command Syntax

```
MMEMory:FEED<"WSP"|"FDC">
```

Query Syntax

N/A

*RST Value

“FDC”

Limits

Two enumerated quoted string values: “FDC”, “WSP”

Related Commands

N/A

Description

The FEED command specifies the I/O mechanism that the Mass Storage Unit uses to return extracted data back to the user. The FEED command only affects the movement of previously collected data OUT of theVX4570 to the user’s host computer. The command allows the user to choose between retrieval of data using Word Serial Protocol (WSP) or Fast Data Channel protocol (FDC). WSP is slower, generic, and will work with any VXIbus controller. FDC is faster and more efficient, but requires Commander-side FDC drivers on the host computer.

The command parameter is a quoted string and has two values: “WSP” selects Word Serial Protocol and “FDC” selects Fast Data Channel protocol. The default is FDC.

Examples

Command	Explanation
MMEM:FEED “WSP”	Assigns Word Serial Protocol as the I/O mechanism to return extracted data.

MMEMory:INITialize

Command Syntax

```
MMEMory:INITialize [<msus>]
```

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands MMEMory:MSIS

Description The INITialize command initializes (formats) the specified mass storage media. If the <msus> parameter is not specified, the default is used. The default is the <msus> selected at *RST, not the currently selected <msus>.

NOTE. *Any previous data on the media is destroyed when the media is initialized.*

Examples

Command	Response
MMEM:INIT	This command initializes (formats) the specified mass storage media.

MMEMory:LOAD

Command Syntax

```
MMEMory:LOAD
   :TRACe <label>
```

Query Syntax

N/A

*RST Value

N/A

Limits

N/A

Related Commands

MMEM:STORe
MMEM:NAME
MMEM:OPEN

Description

The LOAD command returns the contents of the currently selected file.

Reception of this command causes the module to read the file referenced in the last NAME and OPEN commands and transfer its contents to the user. The entire contents of the file is returned and the End Of File is indicated by the detection of the communication protocol End bit.

After sending the LOAD command, you must perform readback until the End bit has been received.

An attempt to access a file that is not open generates an error –383 “Mass Storage Error: No file opened”.

The TRACe command allows you to extract selected information from a file of test results. The file data to be retrieved is defined in the dif_expression referenced by the <label> parameter. The data file that contains the test results is the file currently NAMED and OPENED.

Reception of this command causes the module to read the test results file and extract the requested trace data. The extracted data is buffered and returned to the user.

See *Appendix F: Post Processing* for more information and examples.

Examples

Command	Response
MMEM:LOAD	None
MMEM:LOAD:TRAC TRC1	None

MMEMory:MOVE

Command Syntax

```
MMEMory:MOVE <src_file>,<dest_file>
MMEMory:MOVE <src_file>,<src_msus>,<dest_file>,<dest_msus>
```

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands MMEMory:COPY

Description The MOVE command moves or renames an existing file to another file name.

Two forms of the command are allowed. The first form has two parameters. In this form the first parameter, <src_file>, specifies the file to be renamed, and the second parameter, <dest_file>, specifies the new name of the file. The move operation is performed on the default mass storage device.

The second form has four parameters. The first two parameters, <src_file> and <src_msus>, specify the file name and storage device of the source file. The second two parameters, <dest_file> and <dest_msus>, specify the new file name and new storage device for the source file.

If the <src file> does not exist, the instrument generates error –256 "Mass Storage File Name Error: File Name Not Found". If the <dest_file> already exists, the instrument generates error –257, "Mass Storage File Name Error: File Name already exists".

Examples

Command	Response
MMEM:MOVE "file1.dat", "file2.dat"	This command renames file1.dat. The new file name is now file2.dat.

MMEMory:MSIS

Command Syntax

```
MMEMory:MSIS [<msus>]
```

Query Syntax

N/A

***RST Value**

Highest SCSI device found.

Limits

N/A

Related Commands

MMEMory:INITiate

Description

The “Mass Storage IS” command selects a default mass storage device which is used by all MMEMory commands except INITialize. If the parameter is omitted, the device-dependent setting for default mass storage device is selected.

Examples

Command	Response
MMEM:MSIS	None

MMEMory:NAMe

Command Syntax

```
MMEMory:NAMe<file_name> [,<msus>]
```

Query Syntax

N/A

*RST Value

N/A

Limits

N/A

Related Commands

MMEMory:OPEN

Description

The Name command identifies the name of the file specification used by the OPEN, LOAD, STORe and CLOSe commands.

An attempt to name another file without closing the previous named file with the CLOSe or MSIS commands will generates error –251, “Mass Storage File Name Error: A File Is Already Named”.

Examples

Command	Response
MMEM:NAME “test0.dat”	None
MMEM:OPEN	None

MMEMory:OPEN

Command Syntax `MMEMory:OPEN`

Query Syntax `MMEMory:OPEN?`

Query Response open file status value (NR1)

0 = no files are currently open

1 = a file is currently open

***RST Value** N/A

Limits N/A

Related Commands MMEMory:NAME

Description The OPEN command opens the currently NAMEd file for access. An attempt to open a file that is already open generates error –253, “Mass Storage File Open Error: A File Is Already Opened”. An attempt to open a file before naming a file will generate error –259, “Mass Storage File Open Error: No File Named”.

The Mass Storage Unit’s file structure allows only one file to be opened at a time. The query returns information indicating whether or not a file is currently open.

Examples

Command	Response
MMEM:OPEN	None
MMEM:OPEN?	1

MMEMory:PACK

Command Syntax `MMEMory:PACK [<msus>]`

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands N/A

Description This command packs the mass storage device, so that unused memory between files is recovered and packed together.

Examples

Command	Response
MMEM:PACK	None

MMEMory:STORe

Command Syntax

```
MMEMory:STORe

   :DINTerchange<label>
```

Query Syntax N/A

***RST Value** N/A

Limits N/A

Related Commands

MMEMory:LOAD
MMEMory:NAME
MMEMory:OPEN

Description

The STORe command transfers data from memory to the storage device. Reception of this command causes the module to read data in from the backplane and to store it in the currently NAMED and OPENED file. Storage continues until detection of the communication protocol End bit.

The DINTerchange command allows the user to define dif_expression for use in processing stored file data. The <label> parameter specifies a name to reference the dif_expression. It is a short, unquoted character string. There are typically two kinds of dif_expressions: expressions that define an instrument's state settings and expressions defining trace data extraction parameters. Examples of settings include date and time of a test run, instrument setup values, and notes and observations. Examples of trace parameters include channel selection, count of data values, and time stamp filters.

Reception of this command causes the VX4570 to attempt to read in an ASCII dif_expression. The expression is stored in memory until overwritten by the next DINTerchange command, or the system is reset.

An attempt to access a file that is neither NAMED not OPENED results in an error –383, "Mass Storage Error: No File Opened".

See *Appendix F: Post Processing* for more information and examples.

VXI:COMMander

Command Syntax

```
VXI

:COMMander
   :FDC
      :SELect <log_addr>,<chan_num>
      :MODE <DMA | CPU>
      :OPEN [<log_addr>,<chan_num>]
      :READ [<log_addr>,<chan_num>]
      :WRITe [<log_addr>,<chan_num>]
      :CLOSe [<log_addr>,<chan_num>]
      :IDLE [<log_addr>,<chan_num>]
      :TYPE [<log_addr>,<chan_num>,<"STReam"|"NORMal" >]
```

Query Syntax

```
VXI

:COMMander
   :FDC
      :SELect?
      :MODE?
      :TYPE?
```

Query Response

The query response for the SELECT? parameter is as follows:

“LA = log_addr, Chan Num = chan_num”

log_addr	(Logical Address) integer value (NR1) 0 to 255, or –1 if no value is assigned.
chan_num	integer value (NR1) 0 to 7, or –1 if no value is assigned.

The query response for the MODE? parameter is as follows:

mode value (NR1)

0 = DMA Control Hardware

1 = CPU Control Hardware

The query response for the TYPE? parameter is as follows:

type value (NR1)

4 = STREAM

8 = NORMAL

***RST Value**

Logical Address equals –1.

channel number equals –1.

Default Mode value is set to DMA control hardware.

Default Type is set to Stream transfers.

Limits

N/A

Related Commands

VXI:SERVANT:FDC

Description

This command assigns values for the following parameters for all subsequent VXI:COMMander:FDC commands requiring these parameters.

SELect. Selects the Logical Address and FDC Channel Number for all subsequent VXI:COMMander:FDC commands requiring these parameters.

MODE. Assigns the FDC data transfer mechanism.

Mode = "DMA" selects DMA control hardware to transfer the FDC buffer data across the VXIbus. DMA Control Hardware transfers the FDC data at 2 Mbytes per second.

Mode = "CPU" uses the CPU to read the data from the FDC buffer and write it out the backplane. CPU Control Hardware transfers FDC data at 133 Kbytes per second.

TYPE. Assigns the FDC Transfer type for the selected FDC channel with the selected VXI servant device.

> TYPE = "STReam" selects the FDC Stream protocol.
>
> TYPE = "NORMal" selects FDC Normal Protocol.

The default type is Stream. The servant's Logical Address, the FDC channel number, and transfer type are optional parameters that may be passed to the TYPE function. If they are omitted, then the currently selected values are assumed. All parameters must be either passed or omitted. Combinations will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated. The selected TYPE value will take effect the next time the channel is read or written. See FDC Transfer Protocol, v2.07.

READ. Reads data from the selected VXI Servant device. Data is transferred from the servant via FDC Protocol using the currently selected FDC channel. The servant's Logical Address and the FDC channel number are optional parameters that may be passed to the READ function. If they are omitted then the currently selected values are assumed. Both Logical Address and channel number must be either passed or omitted. Either one by itself will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated.

WRITe. Writes data to the selected VXI Servant device. Data is transferred from the servant via FDC Protocol using the currently selected FDC channel. The servant's Logical Address and the FDC channel number are optional parameters that may be passed to the WRITE function. If they are omitted then the currently selected values are assumed. Both Logical Address and channel number must be either passed or omitted. Either one by itself will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated.

OPEN. Opens the selected FDC channel with the selected VXI servant device. The servant's Logical Address and the FDC channel number are optional parameters that may be passed to the OPEN function. If they are omitted then the currently selected values are assumed. Both Logical Address and channel number must be either passed or omitted. Either one by itself will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated. Once a channel is opened, data can be transferred with READ or WRITE commands. See FDC Transfer Protocol, v2.07.

CLOSe. Closes the selected FDC channel with the selected VXI servant device. The servant's Logical Address and the FDC channel number are optional parameters that may be passed to the CLOSE function. If they are omitted then the currently selected values are assumed. Both Logical Address and channel number must be either passed or omitted. Either one by itself will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated. Once a channel is closed, data cannot be transferred until another READ or WRITE command is issued. See FDC Transfer Protocol, v2.07.

IDLE. Idles the selected FDC channel with the selected VXI servant device. The servant's Logical Address and the FDC channel number are optional parameters that may be passed to the IDLE function. If they are omitted then the currently selected values are assumed. Both Logical Address and channel number must be either passed or omitted. Either one by itself will generate a –109 "Missing Parameter" error. If a valid Logical Address or channel has not been selected (i.e. values = –1), a –221 "Settings Conflict" error is generated. The IDLE function issues the FDC GOTO IDLE command. See FDC Transfer Protocol, v2.07.

Examples

1. `VXI:COMM:FDC:SEL 5 0`

   ```
   OPEN

   READ

   CLOSE
   ```

2.
   ```
   VXI:COMM:FDC:OPEN 5 1

   VXI:COMM:FDC:READ 5 1

   VXI:COMM:FDC:OPEN 5 2

   VXI:COMM:FDC:READ 5 2

   VXI:COMM:FDC:CLOSE 5 2

   VXI:COMM:FDC:CLOSE 5 1
   ```

VXI (VXI subsystem)

Command Syntax

```
VXI

:SELect?     <log_addr>
:CONFigure
   :LADDress?

   :MODEl?

READ?
```

Query Syntax

```
VXI:SELect?
VXI:CONFigure:LADDress?
VXI:CONFigure:MODEl?
VXI:READ?
```

Query Response

`VXI:SELect?`

This query returns the currently selected VXI device that the VX4570 is communicating with. The return value will be the device's logical address, and will be either the VX4750 itself or one of its servant devices. The value is a <NR1> in the range of 1 to 255.

`VXI:CONFigure:LADDress?`

This query returns a comma separated list of logical addresses accessible to the VX4750. The first address will always be that of the VX4750 itself. The list will be <NR1> values in the range of 1 to 255.

`VXI:CONFigure:MODEl?`

This query returns a comma separated list of servant device model numbers in ASCII HEX. The first model number will always be that of the VX4570 itself.

`VXI:READ?`

This query executes an unsolicited read from the currently addressed servant device. This command is designed to tell the VX4570 to do a read back from a servant device and is needed when the servant device implements query commands that are not terminated with question marks. Question-mark terminated query commands signal the VX4570 to perform an automatic read back of servant responses to query commands. This command is only valid when addressed to servant devices of the VX4570. If issued to the VX4570, the command will generate an “Undefined header” error message.

*RST Value

The power-on default value for <log_addr> is the logical address of the VX4570.

Limits

N/A

Related Commands

N/A

Description

This command specifies the logical address which is to be used by all subsequent commands to the VXI subsystem. The <log_addr> value is a <NR1> in the range of 1 to 255. The power-on default value for <log_addr> is the logical address of the VX4570. Logical addresses that are neither the VX4750's nor one of its direct servants will generate error –329, "LA Not Recognized in Servant List".

Examples

Command	Response
VXI:SEL 5	Specifies a logical address of 5 for all subsequent commands.
VXI:SEL?	5
VXI:CONF:LADD?	5, 6, 7

VXI:SERVant

Command Syntax

```
VXI

:SERVant
   :FDC
      :SELect <chan_num>
      :OPEN [<chan_num>]
      :READ [<chan_num>]
      :WRITe [<chan_num>]
      :CLOSe [<chan_num>]
```

Query Syntax

```
VXI
:SERVant
   :FDC
      :SELect?
```

Query Response

"Chan Num = chan_num."

chan_num = integer value (NR1) 0 to 7, or –1 if no value assigned.

***RST Value**

Channel number equals –1.

Limits

N/A

Related Commands

VXI:COMMANDER:FDC

Description

This command assigns values for the following parameters for all subsequent VXI:SERVant:FDC commands requiring these parameters.

SELect. Selects the FDC Channel Number for all subsequent VXI:SERVant:FDC commands requiring that parameter.

READ. Reads data from the selected servant-side FDC channel. The FDC channel number is an optional parameter that may be passed to the READ function. If passed, the parameter defines the channel to read; if omitted, then the currently selected (default) Channel Number is used. If a valid channel has not been selected (i.e. value = –1), a –221 "Settings Conflict" error is generated. Data read from the channel is passed on to the SCPI parser task for processing.

WRITe. Writes test data to the selected servant-side FDC channel. The FDC channel number is an optional parameter that may be passed to the WRITE function. If passed, the parameter defines the channel to write. If omitted, then the currently selected (default) Channel Number is used. If a valid channel has not been selected (i.e. value = –1), a –221 "Settings Conflict" error is generated. This command exists for symmetry and has limited functionality. Currently, it sends an ASCII character test message to the selected FDC channel.

OPEN. Opens the selected servant-side FDC channel. The FDC channel number is an optional parameter that may be passed to the OPEN function.

If passed, the parameter defines the channel to open. If omitted, then the currently selected (default) Channel Number is used. If a valid channel has not been selected (i.e. value = –1), a –221 "Settings Conflict" error is generated. Once a channel is opened, data can be transferred with READ or WRITE commands. See FDC Transfer Protocol, v2.08.

CLOSe. Closes the selected FDC channel. The FDC channel number is an optional parameter that may be passed to the CLOSe function.

If passed, the parameter defines the channel to close. If omitted, then the currently selected (default) Channel Number is used. If a valid channel has not been selected (i.e. value = –1), a –221 "Settings Conflict" error is generated. Once a channel is closed data cannot be transferred until another OPEN and READ or WRITE command is issued. See FDC Transfer Protocol, v2.08.

Examples

1.
```
VXI:SERV:FDC:SEL 1

OPEN

READ

CLOSE
```

2.
```
VXI:SERV:FDC:OPEN 2

VXI:SERV:FDC:READ 2

VXI:SERV:FDC:OPEN 3

VXI:SERV:FDC:READ 3

VXI:SERV:FDC:CLOSE 2

VXI:SERV:FDC:CLOSE 3
```

IEEE-488.2 Common Commands

This section lists the IEEE-488.2 common commands and queries recognized by the VX4570.

*CLS

Clear Status. This commands clears the following:

Event Status register (ESR)
any pending Service Requests (SRQs)
Error queue
Status Byte register

*ESE <NRf>

Event Status Enable (ESE) command. This command defines the mask for setting the Event Status Summary bit (bit 5) in the Status Byte register (*STB?). The mask is logically ANDed with the Event Status register (*ESR?) to determine whether or not to set the Event Status Summary bit. The mask can be any numeric value from 0 to 255, corresponding to the encoded bits of the ESR register. A "1" in a bit position enables reporting of the function. A "0" disables it. The *ESE register is cleared at power-on, or by writing an *ESE 0 command only. If <mask> is not specified, it is unchanged. Bits 1 and 6 are unused, and are always interpreted as zero.

For example, the command *ESE 37 (hexadecimal 25, binary 00100101) enables setting the Event Status Summary bit whenever an operation is complete, a query error is detected, or a command error is detected.

*ESE?

Event Status Enable (ESE) query. This command returns the value of the Event Status Enable register as a numeric value from 0 to 255. For example, a value of 32 (hexadecimal 20, binary 00100000) indicates that command error reporting is enabled.

*ESR?

Event Status Register (ESR) query. This command returns the value of the Event Status register. The *ESR command is destructively read (that is, read and cleared). The Event Status Summary bit in the Status Byte (*STB?) is also cleared by a read of the ESR. The ESR is set to 128 on power-on (bit 7) set. It is cleared by an *ESR? or *CLS command only. When converted to a binary number, the bits of the ESR correspond to:

bit	0 (LSB)	Operation Complete
	1	Request Control (unused, always 0)
	2	Query error
	3	Device Dependent error
	4	Execution error
	5	Command error
	6	User Request (unused, always 0)
	7 (MSB)	Power On

The Error bits are set whenever the module detects an error. The error values from –100 to –199 are Command errors. Error values from –200 to –299 are Execution errors. Error values from –300 to –399 are Device Dependent errors. Error values from –400 to –499 are Query errors. (See *Status and Events* for a listing of all the error codes.)

The Request Control and User Request bits are unused, and are always reported as zeroes.

The Operation Complete bit is set in response to an *OPC command. A 1 indicates that the module has completed all pending commands and queries.

*IDN?

Identification query. This returns a 4-field response. Field 1 is the manufacturer, field 2 the model, field 3 the serial number, and field 4 contains both the SCPI and the firmware version levels. The response syntax is:

```
TEKTRONIX,VX4570,xx000000,SCPI:94.0 FV1.0<LF>
```

*OPC

Operation Complete. This command causes the module to set the Operation Complete bit in the Event Status register (ESR) when all pending commands and queries are complete.

*OPC?

Operation Complete query. This command causes the module to place a “1” in the Output queue when all pending commands and queries are complete. All commands following *OPC are suspended until the pending operations are complete. The *OPC? command does not affect the OPC bit in the Event Status register.

*RST

Reset. This command resets the VX4570 to its power-on state. The condition of the VX4570 after the completion of the power-on self test or receipt of a *RST command is as follows:

Fast Data Channels	disabled
Error Queue	cleared
Buffer size, buffer count	zero (not allocated)
MSIS	highest SCSI device found
Logical Address	–1
channel number	–1
Default Mode	DMA control hardware
Default Type	Stream transfers

*SRE <mask>

Service Request Enable (SRE) register. This command defines a mask for generating VXI Request True interrupts. The mask can be any number from 0 to 255, corresponding to the encoded bits defined below. Bits 0, 1, 2, 3, 6, and 7 of the SRE register are not used, and are ignored if received. A 1 in a bit position enables the corresponding service request. A 0 disables it. The *SRE register is cleared at power-on or by writing an *SRE 0 command only. If <mask> is not specified, it defaults to 0. When converted to a binary number, the bits of the *SRE correspond to:

Bit No.	Meaning
bit0 (LSB)	Not used
1	Not used
2	Not used
3	Not used
4	Message available
5	Event Status Summary
6	Not used
7	Not used

*SRE?

Service Request Enable (SRE) query. This command returns the value of the Service Request Enable register as a numeric value from 0 to 255.

*STB?

Status Byte Query. This query returns the value of the Status Byte register as a numeric value between 0 and 255. The Status Byte register contents are also returned in response to a VXI Read STB command. The Status Byte register is encoded as follows:

Bit No.	Meaning
bit 0 (lsb)	Not used
bit 1	Not used
bit 2	Not used
bit 3	Not used
bit 4	Message available
bit 5	Event Status Summary bit
bit 6	Service Request Pending Summary bit
bit 7	Not used

Bit 4 indicates a message is available in the Output buffer.

For bit 5, a 1 indicates an event status condition is active. This bit reflects the logical AND of the Event Status Enable register and the current Event Status register. If any bits are set after the ANDing, then the Event Status Summary bit is set.

For bit 6, a 1 indicates a VXI Request True interrupt has been generated. Like bit1 5, it reflects the logical AND of the Service Request Enable register, and the currently active service request conditions. If any bits are set after the ANDing, then the Service Request Pending Summary bit is set. This bit is destructively read. That is, it is cleared when the *STB? command is executed.

*TRG

Equivalent to a VXI Trigger command.

*TST?

Execute the self test. The Error LED will be lit while the self test is being executed. The self test takes approximately five seconds to complete. After executing the *TST? command, self test errors (if any) will be queued.

The Operation Complete bit of *ESR? or the *SRE interrupt can be used to determine when the test has been completed. The *TST? query gives a summary of the results. The SYSTem:ERRor? command gives the failure results (if any). See *Appendix C* for a listing of self test failures.

For the self test query, the state of the self test routine is returned, as defined in the responses below.

–1<LF>	self test failed
0<LF>	self test successful

*WAI

Wait to Continue. This command causes the module to wait until all pending commands and queries are complete. All commands following *WAI are suspended until the pending operations are complete.

Status and Events

Status and Events

The Status and Event Reporting System reports asynchronous events and errors that occur in the VX4570 Mass Storage Controller Module. This system consists of four 8-bit registers and two queues that you access through the command language. You can use these registers and queues to query the instrument status and control the interrupts that report events.

In general, after an interrupt occurs, first conduct a serial poll, query the registers to see why the interrupt occurred, and then send the SYSTem:ERROr? query to see a descriptive error message.

This section describes the four registers and two queues of the Status and Event Reporting system. For each register, you are given a description, a table describing all of the bits, and an example of how to use the register.

The Status and Event Reporting process, synchronizing programming commands, and the system messages are also described in this section.

Status and Event Reporting System

The Status and Event Reporting system monitors and reports such events as an error occurring or the availability of a response to a query. This system includes descriptions of the following registers and queues:

- Status Byte Register
- Service Request Enable Register
- Standard Event Status Register
- Event Status Enable Register
- Output queue
- System Error queue

Status Byte Register

The Status Byte Register, shown in Table 4–1, summarizes information from other registers. Use a serial poll or a *STB? query to read the contents of the Status Byte Register. The response is the sum of the decimal values for all bits set. When you use a serial poll, bit 6 shows Request Service information. When you use the *STB? query, bit 6, the Master Status Summary bit, indicates that bits 4 or 5 may be set.

Table 4–1: The Status Byte Register

Bit	Decimal Value	Function
0–3	–	Not used.
4	16	Message Available shows that output is available in the Output queue.
5	32	Event Status Bit indicates that one or more events have occurred and the corresponding bits in the Standard Event Status Register have been set.
6	64	Request Service (obtained from a serial poll) shows that the VX4570 has requested service from the GPIB controller. Master Status Summary (obtained from *STB? query) summarizes the Event Status bit, Message Available bits, and Error/Event queue Not Empty bits in the Status Byte Register.
7	–	Not used.

A common example of using the Status Byte register is to enable the Message Available bit. This is done by sending an *SRE 16 command to the VX4570. If the *STB? query returns a value of 80, bit 4 (decimal value of 16) and bit 6 (decimal value of 64) have been set (giving a decimal sum of 80). Bit 4 indicates that a message is available in the output queue. Bit 6 indicates that a bit in the Status Byte register that has been enabled by setting the corresponding bit in the Service Request Enable register (in this case bit 4) has been set.

Service Request Enable Register

The Service Request Enable Register, shown in Table 4–2, controls which bits in the Status Byte Register will generate a service request. Use the *SRE command to set bits in the Service Request Enable Register. Use the *SRE? query to see which bits in this register are enabled. The response from this query is the sum of the decimal values for all bits set.

Table 4–2: The Service Request Enable Register

Bit	Decimal Value	Function
0–3	–	Not used.
4	16	Message Available Bit indicates that a service request will be generated when a message is placed in the Output queue.
5	32	Event Status Bit indicates that events summarized in bit 5 of the Status Byte Register will generate a service request.
6–7	–	Not used

If, for example, the *SRE? query returns a value of 48, bits 4 and 5 are set in the Service Request Enable Register. Any event that causes the Message Available bit (bit 4) or Event Status bit (bit 5) to be set in the Status Byte Register now generates an interrupt. If you want an interrupt to be generated only when the Event Status bit (bit 5) is set, use the *SRE 32 command.

Standard Event Status Register

The Standard Event Status Register, shown in Table 4–3, records many types of events that can occur in the VX4570. Use the *ESR? query to read the contents of this register. The response is the sum of the decimal values for all bits set. Reading this register clears all bits so the register can accumulate information about new events.

Table 4–3: The Standard Event Status Register

Bit	Decimal Value	Function
0	1	Operation Complete shows that the operation is complete. This bit is set when all pending operations complete following a *OPC command.
1	–	Not used.
2	4	Query Error shows that data in the Output queue was lost.
3	8	Device Dependent Error indicates that device operation could not proceed properly because of some internal condition.
4	16	Execution Error shows that an error occurred while the VX4570 was executing a command or query.
5	32	Command Error shows that an error occurred while the VX4570 was parsing a command or query.
6	–	Not used
7	128	Power On shows that the VX4570 was powered on.

The following example assumes that all bits have been enabled using the Event Status Enable Register (see the next section for information about this register). If a *ESR? query returns a value of 128, bit 7 (decimal value of 128) is set indicating that the instrument is in the initial power-on state.

Event Status Enable Register

The Event Status Enable Register, shown in Table 4–4, controls which events are summarized in the Event Status bit (bit 5) of the Status Byte Register. Use the *ESE command to set bits in the Event Status Enable Register. Use the *ESE? query to see what bits in the Event Status Enable Register are set. The response from this query is the sum of the decimal values for all bits summarized in the event status bit of the Status Byte Register.

Table 4–4: The Event Status Enable Register

Bit	Decimal Value	Function
0	1	Set bit 5 of the Status Byte register when bit 1 (the Operation Complete bit) of the Standard Event Status register is set.
1	2	Not used.
2	4	Set bit 5 of the Status Byte register when bit 2 (the Query Error bit) of the Standard Event Status register is set.
3	8	Set bit 5 of the Status Byte register when bit 3 (the Device Dependent Error bit) of the Standard Event Status register is set.
4	16	Set bit 5 of the Status Byte register when bit 4 (the Execution Error bit) of the Standard Event Status register is set.
5	32	Set bit 5 of the Status Byte register when bit 5 (the Command Error bit) of the Standard Event Status register is set.
6	64	Not used.
7	128	Set bit 5 of the Status Byte register when bit 7 (the Power On bit) of the Standard Event Status register is set.

If, for example, the *ESE? query returns a value of 255, all bits are set, indicating that all events will set the Event Status bit (bit 5) of the Status Byte Register.

The Output Queue

The VX4570 stores query responses in the Output queue.

The System Error Queue

The VX4570 error messages are stored in the System Error queue. Use the SYSTem:ERRor? query to get the error number and a text description of the error. Reading an error removes it from the queue.

The following error messages are returned by the SYSTem:ERRor? query:

–100 Command error:

–103 Invalid separator:

–109 Missing parameter:

–111 Header separator error:

–112 Program mnemonic too long:

–113 Undefined header:

–151 Invalid string data:

–200 Execution error

–221 Settings Conflict; No Assigned LA:

–224 Illegal parameter value

–250 Mass Storage Error.

–251 Mass Storage File Name Error: A File Is Already Named.

–252 Mass Storage Error: Missing media

–253 Mass Storage File Open Error: A File Is Already Opened.

–254 Mass Storage Error: Media Full.

–255 Mass Storage Error: Directory Full

–256 Mass Storage Error: File Name Not Found

–257 Mass Storage File Name Error: File Name already exists

–259 Mass Storage File Open Error: No File Named

–300 Device specific error

–310 System error

–311 Memory error

–319 FDC error: invalid FdcMbx

–320 FDC channel not created

–321 FDC channel open failure

–322 Invalid FDC channel

–323 FDC Read Ready Bit timeout:

–324 FDC DMA Done timeout:

–325 FDC Write Ready Bit timeout:

–326 FDC Error Bit Asserted:

–327 FDC Channel Already Open

–328 FDC Channel Transfer Cancelled:

–329 LA Not Recognized in Servant List

–330 Self–test failed:

–331 FDC error: tCmdrfdcCtrl spawn failed

–332 Cmd Pass Through Error: Servant Response Timeout.

–333 FDC buffer write failure.

–334 Memory Pool Free Error.

–335 FDC error: No Buffer Memory Allocated

–336 Memory Pool Free Error: FDC Channels Active

–337 Memory Already Allocated

–338 Memory Pool Allocation Error.

–339 Memory Pool Free Error: RcvMemMbx failure.

–340 Memory Pool Free Error: Task delete failure.

–341 Memory Pool Free Error: Mbx delete failure.

–350 Queue overflow

–364 Cmdr WSP Drivers Detected Bus Error

–365 Cmdr FDC DMA Drivers Detected Bus Error

–370 Memory Allocation Error: Rcv Input

–371 Sh Memory Pool Allocation Error: mmgr

–372 DIF Memory Allocation Error: difDB

–373 DIF Extraction Memory Allocation Error: difDB

–374 DIF Return Buffer Allocation Error: difDB

–375 DIF Trace Specification Memory Allocation Error: difDB

–376 DIF Setting Specification Memory Allocation Error

–377 Memory error: ScpiParser allocation failure

–380 Invalid DIF Expression: Using Defaults

–381 Invalid DIF Expression

–382 Mass Storage Error: File Appending Is Not Allowed

–383 Mass Storage Error: No File Opened

–384 FDC Warning:returning with data cnt = 0, but no errors:

–385 Mass Storage Error: ioctl Seek Failure

Appendices

Appendix A: Specifications

Table A–1: VXI Instrument Characteristics

Characteristics	Description
VXI General Characteristics	The instrument provides a VXI interface that complies with Revision 1.4. The VXI interface is defined by the VXI Consortium, Inc.
Interface Type	Message Based (1.4)
Protocols	Word Serial (WSP)
TTL Outputs	VXI TTLTRG* Lines TTLTRG0* through TTLTRG7* under program control.

Table A–2: Environmental/Reliability Characteristics

Characteristics	Description
Temperature	Operating: Meets or exceeds MIL-T-28800E for Type III, 0 to 50° C external ambient, when operated in a mainframe providing Class 3 equipment. Non-operating: –40° C to +71° C Airflow of at least 0.5 liters/sec at 0.01 mm H_2O air pressure, –10° C/55° C for 10° C (or less) temperature rise of internal air, as measured at the cooling air exit points, and with no heat transfer either to or from any adjacent VXI modules.
Humidity	Relative Humidity: Non-oper: Up to 95%, at up to 50° C. Operating: Up to 95% at up to 30° C, and up to 45%, at up to 50° C.
Altitude (1)	Operating: 6,000 ft. altitude.
Altitude (2)	Meets or exceeds MIL-T-28800E for Type III, (operating to 10,000 ft., non-operating to 15,000 ft.).

Table A–3: VX4570-Specific Characteristics

Characteristics	Description
VXI Compliance	Instrument complies with revision 1.4 of the VXI specification.
VXI Device Classification	Message based device.
VME Interrupter Level	Switch selectable to a level between 1 and 7.
VXI Logical Address	Switch selectable to a value between 0 and 254.
Contents of device/manufacturer dependent VXI registers.	ID Register: 9FFD hexadecimal. Device Type: 95C5 hexadecimal.

Table A–3: VX4570-Specific Characteristics (Cont.)

Characteristics	Description
VXI TTL Trigger Outputs	One or more of the VXI TTLTRG* signals may be driven. All TTLTRG* outputs may be disabled.
VXI TTL Trigger Inputs	One of the VXI TTLTRG* signals may be selected to be polled or to act as an interrupt source to the module's microprocessor.
CPU	
Local CPU memory	SRAM 512 Kbytes (four 128 K × 8 parts)
Boot/Monitor ROM	128 Kbytes (64 K × 16 parts)
Flash ROM	2 Mbytes (four 512 K × 8 parts)
Shared Memory	8 Mbytes (two 1-Meg × 32 SIMMs)

Table A–4: Certifications and Compliances

EC Declaration of Conformity	Meets intent of Directive 89/336/EEC for Electromagnetic Compatibility. Compliance was demonstrated to the following specifications as listed in the Official Journal of the European Communities: EN 55011 Class A Radiated and Conducted Emissions EN 50081-1 Emissions: EN 60555-2 AC Power Line Harmonic Emissions EN 50082-1 Immunity: IEC 801-2 Electrostatic Discharge Immunity IEC 801-3 RF Electromagnetic Field Immunity IEC 801-4 Electrical Fast Transient/Burst Immunity IEC 801-5 Power Line Surge Immunity To ensure compliance with EMC requirements this module must be installed in a mainframe which has backplane shields installed which comply with Rule B.7.45 of the VXIbus Specification. Only high quality shielded cables having a reliable, continuous outer shield (braid and foil) which has low impedance connections to shielded connector housings at both ends should be connected to this product.

Appendix B: Input/Output Connections

The front-panel SCSI connector conforms to the SCSI–2 standard.

Appendix C: Self Test Failures

For the fields below, underscores (_) represent a field filled in by the program.

If an error occurs during self test, the SYSTem:ERRor? query will return the message “–330 Self-test failed: <failure>”, where <failure> will be one of the following messages:

```
SRAM - addr 0x_______ data = 0x____, should be 0x____
```

The Static RAM (SRAM) could not be written or read properly with a pattern of 0x5555 or 0xaaaa. The SRAMs are 128K × 8-bit devices interfacing to a 16-bit bus. They are configured in pairs with U161 (msb) and U261 (lsb) for addresses 0x2000000 through 0x203fffe, and U361 (msb) and U461 (lsb) for addresses 0x2040000 through 0x207fffe.

```
DRAM in J___ - addr 0x________ data = 0x____, should be 0x____
```

The Dynamic RAM could not be written or read properly with a pattern of 0x5555 or 0xaaaa. The DRAMs are 1M × 32-bit SIMMs with J831 for addresses 0x10000000 through 0x103ffffe, and J931 for addresses 0x11000000 through 0x113ffffe.

Only the first 128K of each 16M page is tested during the power-on self test or the *TST? command. The DIAG:DRAM command will perform the memory test on all memory, which takes approximatly four minutes.

```
68901 ____ register - should be 0x__, is 0x__
```

A particular register in the MC68901 Multi-Function Chip (U76) could not be written or read properly with a pattern of 0x55 or 0xaa.

```
Serial NVRAM
```

The Serial Non-Volitale RAM (U662) could not be written or read properly.

```
Real Time Clock
```

The RTC (U852) could not be programmed.

```
SCSI chip-scratch reg lo word - should be 0x____, is 0x____
```

The lower word (D0–D15) of the SCSI chip (U31 of daughter board) Scratch register could not be written or read with a pattern of 0x5555 or 0xaaaa.

```
SCSI chip-scratch reg hi word - should be 0x____, is 0x____
```

The upper word (D16–31) of the SCSI chip (U31 of daughter board) Scratch register could not be written or read with a pattern of 0x5555 or 0xaaaa.

```
AM/Page register – should be 0x____, is 0x____
```

The Address Modifier register or the Page register could not be written or read properly with a pattern of 0x5555 or 0xaaaa. The Page register (Uxxx) is the upper byte and the AM register (Uxxx) is the lower byte.

Appendix D: User Service

This appendix contains service-related information for the VX4570 that covers the following topics:

- Performance verification
- Preventive maintenance
- Troubleshooting
- User-replaceable parts

Performance Verification

You may use the Functional Check procedure listed in the *Getting Started* section of this manual to verify that the module is operating correctly. The instrument has been fully tested and calibrated before leaving the factory.

If the self test indicates a failure, contact your Tektronix field office or representative for assistance.

Preventive Maintenance

You should perform inspection and cleaning as preventive maintenance. Preventive maintenance, when done regularly, may prevent VX4570 malfunction and enhance reliability. Inspect and clean the VX4570 as often as conditions require by following these steps:

1. Turn off power and remove the VX4570 from the VXIbus mainframe.
2. Remove loose dust on the outside of the instrument with a lint-free cloth.
3. Remove any remaining dirt with a lint-free cloth dampened with water or a 75% isopropyl alcohol solution. Do not use abrasive cleaners.

Troubleshooting

If you suspect a malfunction, first double check connections to and from the VX4570. If the trouble persists, perform a self test.

If the self test indicates a failure, contact your Tektronix field office or representative for assistance.

User-Replaceable Parts

Refer to *Appendix G: Replaceable Parts List* for parts information.

Appendix E: Diagnostic Commands

The following diagnostic commands can be used to test the functions of the instrument. Refer to *Syntax and Commands* for information on formatting the commands. All of these are arguments for the DIAGnostic command.

Command DIAGnostic	Response
`:NVRAm`	Test the Nonvolatile RAM.
`:RTC`	Test the Real Time Clock chip.
`:SCSIchip`	Test the SCSI chip.
`:PAGE`	Test the Address Modifier and Page register. The Page register drives A24 through A32 during VME master cycles.
`:MFP`	Test the Multifunction Peripheral chip (MC68901). The MFP chip is the interrupt controller for the board and also supplies the Bus Error signal to the CPU in the event of a malfunctioning onboard peripheral.
`:DRAM`	Performs a memory test on the DRAM. With 8 Megabytes installed, the test takes approximately 2.5 minutes. During self test the first 128 Kbytes of each 4 Megabyte page is tested.
`:SRAM`	Performs a memory test on the CPU RAM.
`:DMA <numeric_value>`	Test the DMA controller by transferring data between the onboard DRAM and some device on the VMEbus at the A24 address specified by the numeric_value. The device must have at least 64 Kbytes (65,535) of A24/D16 memory. Example: DIAG:DMA 2097152
`:TRIGgers <numeric_value>`	Test the VXI TTL trigger line hardware. This test will generate a VXI trigger TTL trigger line specified by numeric_value. The trigger hardware/software will verify that the TTL trigger line was asserted.
`:VMERead? <numeric_value>`	Test the Master hardware. This test will do a VME A16/D16 read cycle at the address specified by the numeric_value. Returns the D16 data value or the message “BERR at VME address numeric_value” if no device is present at that address.
`:MEDIa? <numeric_value>`	Returns information about the SCSI device at BUSID numeric_value. Valid numeric_values are 0–6. The string returned will contain the vendor ID, product ID, revision, type, number of blocks, and block size.

Command DIAGnostic	Response
`:SECRead? <numeric_value>`	Returns the data in the sector specified by numeric_value. The amount of data returned is 512 bytes. ***NOTE:*** *The data may be binary, so the readback should terminate only on the VXI End Bit and not a control character, such as linefeed.*
`:SECWrite <numeric_value>`	Writes 512 bytes of data to sector numeric_value. ***NOTE:*** *The data is not part of the command. Two writes are performed, one to send the command and the second to send 512 bytes of data.* Example: first write: DIAG:SECW 6<terminator>second write: <1–512 bytes of data><terminator>

Appendix F: Post Processing

The VX4570 Mass Storage Unit (MSU) is designed to quickly store data for extended periods of time. This application creates large files of collected data that require post processing for analysis. Post processing information that has been stored on the VX4570 MSU requires retrieving data that has been streamed onto the storage unit via a collection mechanism that is optimized for real time operation. Consequently, the format of the stored data depends on the type of device generating the data, how the device was set up, and the order in which the data was received from the device.

Data Interchange Process

A variety of instruments can provide data for VX4570 storage. A generic and flexible method of data retrieval is necessary to support the numerous types of data that can be collected. The Standard Commands for Programmable Instruments specification (SCPI) provides a mechanism that allows data to be shared between software packages and instruments. Called Data Interchange Format (DIF), the protocol is flexible, extensible, and can accommodate many data formats and structures.

Using the DIF protocol, data can be specified and retrieved from the MSU. Files of collected data can be referenced by name, individual devices can be selected, channels can be referenced individually or in groups, data can be specified with starting and ending indexes or time stamp, and collection sizes can be specified by byte or sample count. Test environment data can also be referenced, such as date, time, name of the test engineer, and a test description.

An application program (typically a Graphical User Interface [GUI]) allows the user to define the data to be retrieved. The request is sent to the VX4570 as an ASCII DIF expression. A DIF expression that specifies the data to be retrieved is called a ‘trace’ specification. Another type of DIF expression, a ‘settings’ specification, specifies the device’s setup values. The DIF expression is parsed by the MSU, the requested data is extracted from the file of test data, buffered up and returned to the user. A user application program can then analyze the extracted data.

DIF Expressions

The DIF expressions used by the VX4570 are based on the Data Interchange Format described in the SCPI 1994 specification. The basic command set is used where appropriate, but extensions to the DIF command set have also been used to describe setup parameters for complex instruments not completely addressed in the formal DIF specification.

DIF expressions are block structured and hierarchical. Expressions typically consist of several blocks. Each block contains a keyword that defines the type of block. Within each block, keywords with values describe the setup parameters. DIF expressions can specify both how an instrument is set up as well as what collected data to retrieve.

Settings Specifications

A DIF expression that specifies a device's setup values is called a 'settings' specification. A settings specification includes a name to reference it by. Knowledge of an instrument's setup values is sometimes required to correctly return the requested data.

For example, the SCPI command 'MMEM:STOR:DINT SET1' would define a settings specification named 'SET1'. The actual definition of SET1 is an ASCII DIF expression.

The following block definitions are used by the MSU to record an instrument's setup:

SETting — The Setting block describes an instrument setup. Sub-blocks within the setting block describe the individual setup parameters. The Setting block is required in a DIF setting expression. A Setting block has an associated <Label> that uniquely identifies the block.

DIMension — The Dimension block identifies a group of related setup parameters in an instrument. This sub-block may occur several time in a Settings definition. An example of a settings dimension would be a group of digitized channels, all armed, sampled, and triggered identically.

LADDress — The Logical Address sub-block identifies the logical address of the servant device that the setting specification refers to. The logical address value is an <NR1>.

LABel — A required sub-block, Label identifies the dimension. The identifier is of type <Label>. Dimension Labels can not be repeated within a settings statement.

CINDex — CINDex defines the total number of enabled channels in the group. It is an integer value of 0 to 4.

CSELect — CSELect identifies the specific channels that are enabled in a group. It is a string value, composed of up to four comma-separated integer values in the range of 1 to 4, or the value 99 if no channels are enabled in the group. For example, "1,3" indicates channels 1 and 3 are enabled in the group.

FREQuency — Frequency identifies the sampling frequency of the setup dimension.

SIZE — Size specifies the record setup size for the group.

Trace Specifications

A DIF expression that specifies retrieval data is called a ‘trace’ specification. A trace specification includes a name to reference it by.

For example, the SCPI command ‘MMEM:STOR:DINT TRC1’ defines a trace specification named ‘TRC1’. The actual definition of TRC1 is an ASCII DIF expression. The SCPI command ‘MMEM:LOAD:TRAC TRC1’ prompts the MSU to retrieve the data defined in the TRC1 trace specification.

The following block definitions are used by the MSU in the ASCII DIF expression to specify data to be extracted from a data collection file:

Trace — The Trace block describes an entire waveform trace specification. Sub-blocks within the trace block describe the individual details of the data set. The Trace block is required in a DIF trace expression. A Trace block has an associated <Label> that uniquely identifies the trace block. A <Label> is a short, unquoted string (<=5 characters).

Name — The Name block is an arbitrary name or description of the trace. It is a quoted human readable string value.

Note — The Note block is arbitrary quoted string data. Both Name and Note block character strings should not exceed 32 characters.

Setting — The Setting block identifies an instrument setup with a <Label> that uniquely identifies the setup.

Independent — The Independent block identifies one independent dimension of the trace. This sub-block may occur several time in a Trace definition. An example of an independent dimension of a trace may be a single channel of digitized data.

LADDress — Defines the logical address of the servant device. This is necessary when multiple devices are sending data to the MSU.

Label — An Independent sub-block, Label identifies an Independent dimension. Only Labels defined in a Dimension block of a Setting’s DIF expression are allowed. The identifier is of type <Label>.

CSELect — CSELect identifies the channel whose data is to be collected. The value is an <+NR1> in the range of 0 to 15. If CSELect is omitted, error “–381, Invalid DIF Expression” is generated.

Start — Start identifies the starting index (inclusive) of data extraction. It is an <+NR1>. If Start is omitted in the in the Independent block, or is an invalid value, a default value of 0 is used.

Stop — Stop specifies the ending index value (inclusive). It is an <+NR1>. If stop is invalid, error “–381: Invalid DIF Expression” is generated.

Count — Count specifies the number of byte (8 bit) values to be extracted. It is an <+NR1> value. If Count is invalid, error "–381: Invalid DIF Expression" is generated.

Sample — Sample specifies the number of samples of data to be extracted. The actual number of data bytes collected depends on the type of collected data (8 bit, 16 bit, or 32 bit).

Step — Step specifies the increment value between data values. A Step value of 1 would select every value; a Step value of 2 would select every other data value; a Step value of 3 would select every third byte, etc. Step is an <+NR1>. The default value for Step is 1.

TSStart — Time stamp start. Start extracting data TSStart time after trigger. <+NR3>. *Note: not implemented in initial release.*

Examples of the Data Extraction Process

The VX4570 allows data to be collected for long periods of time, generating extremely large test result data files. Up loading an entire results file from the VX4570 to the system host computer for analysis could involve moving a very large file across the VXIbus backplane. Additionally, data stored by the MSU during collection is placed on the media in a fashion that is run-time efficient, but not in any standard file format (The overhead of placing the data in a standard file format (e.g. DOS) cannot be tolerated). Post processing, therefore, requires the user be allowed to select a subset of collected data, and drivers that extract that data for presentation to the user.

During test execution, test data is stored in a file by the MSU. In order to extract data for analysis, the test file must be opened, and the user must define the data of interest. A SCPI DIF Trace expression is used to describe the data. The test file is opened, the requested data extracted and then returned to the user via the MMEM:LOAD:TRACE SCPI command.

Example 1

A typical test results retrieval sequence is as follows:

1. A test is run, with the results being stored on the MSU in a file called 'Test0.dat'.

2. A user application program running on the host computer system (such as a Plug&Play Soft Front Panel) assists the user in defining the data to be extracted. The application program creates a SCPI DIF Trace expression that is sent to the VX4570. The DIF expression is sent as two separate statements; one that defines the statement as a DIF expression, followed by the actual ASCII DIF expression itself. The two statements must be separated by a valid SCPI statement terminator. (Because ASCII DIF expressions can be quite verbose, a block transfer mode is used to enhance the expression's

transfer from the host to the MSU. The separating terminator allows the VX4570’s I/O to switch in and out of this mode).

3. The test results file is opened, and the SCPI LOAD command that actually commands the card to return the requested data is sent to the MSU.

4. After sending the LOAD command, the host computer does read back from the MSU to acquire the selected data.

The following command sequence illustrates this process.

1.

```
MMEM:STOR:DINT TRC1
(DIF (
   VERSion 1993.0)
   TRACe = TRC1 (
      NAME "test 1 chan 0 & 1 results"
      NOTE "chan 0 in bytes, chan 1 in samples"
      SETting SET1
      INDependent (
         LADDress 5
         LABel G0
         CSELect 0
         STARt 0
         COUNt 16384)
      INDependent (
         LADDress 5
         LABel G0
         CSELect 1
         STARt 100
         SAMPle 100)))
```

Statement 1 defines a SCPI DIF trace expression. ‘MMEM:STOR:DINT TRC1 is the SCPI command that indicates a DIF Trace expression is to follow. The DIF expression is referenced by the name ‘TRC1’. The actual ASCII DIF expression is enclosed in parentheses. The DIF expression has a Trace block labeled ‘TRC1’. (Trace labels are used for internal bookkeeping).

The Trace block contains Name and Note sub-blocks containing string information. The Setting sub-block references a DIF Setting expression (referenced by SET1) that contains setup information about the instrument. The Independent sub-block defines one individual dimension of collected data, identifying the instrument group, specific channel, the beginning and ending point of the collected data.

2.

```
MMEM:NAME "test0.dat"
MMEM:OPEN
```

Statement 2 identifies a test results file and opens it for access.

3.

```
MMEM:LOAD:TRACe TRC1
```

Statement 3 demonstrates the SCPI LOAD command that actually returns data defined in the DIF trace expression. The returned data would be 16k bytes from channel 0 followed by 100 samples (starting at the 100th sample) of channel 1 data.

4.

```
MMEM:STOR:DINT TRC2
(DIF (
   VERSion 1993.0)
   TRACe = TRC2 (
      NAME "test 2 chan 5 results"
      NOTE "100 samples"
      SETting SET1
      INDependent (
         LABel G1
         CSELect 5
         STart 0
         SAMPle 100)))
```

5.

```
MMEM:STOR:DINT TRC3
(DIF (
   VERSion 1993.0)
   TRACe = TRC3 (
      NAME "test 3 chan 0 and 1 results"
      NOTE "chan 0 in bytes, chan 1 in samples"
      SETting SET3
      INDependent (
         Label G0
         CSELect 0
         STARt 0
         COUNt 16384)
      INDependent (
         LABel G0
         CSELect 1
         STARt 10000
         SAMPle 100)))
```

Statements 4 and 5 define two more Trace expressions.

6. `MMEM:LOAD:TRACe TRC2`

Statement 6 returns the first 100 samples of channel 5 data.

7. `MMEM:LOAD:TRACe TRC3`

Statement 7 returns 16k bytes of channel 0 data, followed by 100 samples of channel 1, offset 10000 samples from the start of the buffer.

Example 2

Frequently, data cannot be properly extracted from the test results file without knowing how the instrument was programmed prior to data collection. Instrument setup parameters that are pertinent to data extraction are contained in a settings DIF Expression.

A user application program running on the host computer system (such as a Plug&Play Soft Front Panel) assists the user in setting up an instrument for data collection. The application program creates a SCPI DIF Settings expression, based on the instrument's setup parameters, that is sent to the VX4570. The DIF expression is sent as two separate statements; one that defines the statement as a DIF expression, followed by the actual ASCII DIF expression itself. The two statements must be separated by a valid SCPI statement terminator. (Because ASCII DIF expressions can be quite verbose, a block transfer mode is used to enhance the expression's transfer from the host to the MSU. The separating terminator allows the VX4570's I/O to switch in and out of this mode).

When the DIF Trace expression arrives, the MSU utilizes the setup information in the Settings DIF expression to properly index into the test results file.

This example defines a Settings DIF expression with four dimensions. Each dimension has a servant logical address, a group label, a list of channels enabled, the sampling frequency for the group, and the data collection buffer size.

```
MMEM:STOR:DINT SET
(DIF (
   VERSion 1994.0)
   IDENtify (
      DATE 1993,9,29
      TIME 16,4,14.23)
   SET SET1
DIMension (
   LADDress 5
   LABel G0
   CINDex 4
   CSELect 1,2,3,4
   FREQuency 20000
   SIZE 16384)
DIMension (
   LADDress 5
   LABel G1
   CINDex 2
   CSELect 2,3
   FREQuency 50000
   SIZE 65536)
DIMension (
   LADDress 5
   LABel G2
   CINDex 1
   CSELect 4
   FREQuency 100000
   SIZE 262144)
DIMension (
   LADDress 5
   LABel G3
   CINDex 0
   CSELect 99))
```

Appendix G: Replaceable Parts List

This section contains a list of the replaceable modules for the VX4570. Use this list to identify and order replacement parts.

Parts Ordering Information

Replacement parts are available through your local Tektronix field office or representative.

Changes to Tektronix products are sometimes made to accommodate improved components as they become available and to give you the benefit of the latest improvements. Therefore, when ordering parts, it is important to include the following information in your order.

- Part number
- Instrument type or model number
- Instrument serial number
- Instrument modification number, if applicable

If you order a part that has been replaced with a different or improved part, your local Tektronix field office or representative will contact you concerning any change in part number.

Change information, if any, is located at the rear of this manual.

Module Servicing

Modules can be serviced by selecting one of the following three options. Contact your local Tektronix service center or representative for repair assistance.

Module Exchange. In some cases you may exchange your module for a remanufactured module. These modules cost significantly less than new modules and meet the same factory specifications. For more information about the module exchange program, call 1-800-TEK-WIDE, extension 6630.

Module Repair and Return. You may ship your module to us for repair, after which we will return it to you.

New Modules. You may purchase replacement modules in the same way as other replacement parts.

Using the Replaceable Parts List

This section contains a list of the mechanical and/or electrical components that are replaceable for the VX4570. Use this list to identify and order replacement parts. The following table describes each column in the parts list.

Parts List Column Descriptions

Column	Column Name	Description
1	Figure & Index Number	Items in this section are referenced by figure and index numbers to the exploded view illustrations that follow.
2	Tektronix Part Number	Use this part number when ordering replacement parts from Tektronix.
3 and 4	Serial Number	Column three indicates the serial number at which the part was first effective. Column four indicates the serial number at which the part was discontinued. No entries indicates the part is good for all serial numbers.
5	Qty	This indicates the quantity of parts used.
6	Name & Description	An item name is separated from the description by a colon (:). Because of space limitations, an item name may sometimes appear as incomplete. Use the U.S. Federal Catalog handbook H6-1 for further item name identification.
7	Mfr. Code	This indicates the code of the actual manufacturer of the part.
8	Mfr. Part Number	This indicates the actual manufacturer’s or vendor’s part number.

Abbreviations

Abbreviations conform to American National Standard ANSI Y1.1–1972.

Mfr. Code to Manufacturer Cross Index

The table titled Manufacturers Cross Index shows codes, names, and addresses of manufacturers or vendors of components listed in the parts list.

Manufacturers Cross Index

Mfr. Code	Manufacturer	Address	City, State, Zip Code
00779	AMP INC.	CUSTOMER SERVICE DEPT PO BOX 3608	HARRISBURG, PA 17105–3608
06540	NEW HAVEN MFG CORPORATION	AMATOM ELECTRONIC HARDWARE DIV 446 BLAKE ST	NEW HAVEN, CT 06515–1238
0JR04	TOSHIBA AMERICA INC	9775 TOLEDO WAY	IRVINE, CA 92718
0KB01	STAUFFER SUPPLY CO	810 SE SHERMAN	PORTLAND, OR 97214–4657
61857	SAN–O INDUSTRIAL CORP	91–3 COLIN DRIVE	HOLBROOK, NY 11741
62559	SCHROFF INC	170 COMMERCE DRIVE	WARWICK, RI 02886–2430
71400	BUSSMANN	DIVISION COOPER INDUSTRIES INC PO BOX 14460	ST LOUIS, MO 63178
80009	TEKTRONIX INC	14150 SW KARL BRAUN DR PO BOX 500	BEAVERTON, OR 97077–0001
0KM03	INSTRUMENT SPECIALTIES CO INC.	505 PORTER WAY	PLACENTIA, CA 92670
30817	INSTRUMENT SPECIALTIES CO INC	EXIT 53, RT 80 BOX A	DELAWARE WATER GAP, PA 18327
TK2647	INSTRUMENT SPECIALTIES CO INC.	C/O TEMCO NW 1336 SE 51ST STREET	HILLSBORO, OR 97123

Replaceable Parts List

Fig. & Index Number	Tektronix Part Number	Serial No. Effective	Serial No. Discont'd	Qty	Name & Description	Mfr. Code	Mfr. Part Number
	070–9259–XX			1	MANUAL,TECH:USER	80009	070–9259–XX
1	950–4091–00			4	SCREW M 2.5 X 5 PANHEAD PHIL	0KB01	950–4091–00
2	163–0478–XX			1	IC,DIGITAL:CMOS,PRGM (U111, REF DESIG PRINTED ON CKT BRD)	80009	163–0478–XX
3	950–1678–00			1	SHIELD:BOTTOM	80009	950–1678–00
4	342–1017–00			1	INSULATOR MYLAR BACKSHIELD	80009	342–1017–00
5	211–0867–00			16	SCREW PHIL M2.5X 4,FLHD SS	0KB01	211–0867–00
6	348–1434–00			4	GASKET,EMI:2.912 L,CLIP ON	30817	97–613–17–029
7	950–8663–00			8	REAR SHIELD STANDOFF	80009	950–8663–00
8	950–5688–00			4	STANDOFF HEX M2.5 THRU X .538L	80009	950–5688–00
9	163–0475–XX			1	IC,DIGITAL:CMOS,PRGM (U183, REF DESIG PRINTED ON CKT BRD)	80009	163–0475–XX
10	163–0513–XX			1	IC,DIGITAL:CMOS,PRGM (U282, REF DESIG PRINTED ON CKT BRD)	80009	163–0513–XX
11	159–0116–00			1	FUSE,CARTRIDGE:1A,125V (F48, REF DESIG PRINTED ON CKT BRD)	61857	SM4–1A
12	131–0993–00			1	CONN,BOX:SHUNT,FEMALE,STR,1 X 2,0.1 CTR,0.385 H,30 BLACK	00779	530153–2
13	159–0207–00			1	FUSE,PLUG–IN:5A,125V,FAST BLOW (F59, REF DESIG PRINTED ON CKT BRD)	61857	SM1–5
14	160–9691–01			1	IC, MEMORY:CMOS,EPROM (U24, REF DESIG PRINTED ON CKT BRD)	80009	160–9691–01
15	163–0476–00			1	IC,DIGITAL:CMOS (U77, REF DESIG PRINTED ON CKT BRD)	80009	163–0476–00
16	163–0477–XX			1	IC,DIGITAL:CMOS,PRGM (U962, REF DESIG PRINTED ON CKT BRD)	80009	163–0477–XX
17	337–4085–00			1	SHIELD,ELEC:FRONT	80009	337–4085–00
18	337–4094–00			2	SHIELD:DIVIDER EDGE SHIELD	80009	337–4094–00
19	950–4051–00			4	STANDOFF HEX M2 9 X 19MM	80009	950–4051–00
20	156–4382–00			1	IC,MEMORY:CMOS,DRAM (J931, REF DESIG PRINTED ON CKT BRD)	0JR04	THM321000AS–70
21	156–4382–00			1	IC,MEMORY:CMOS,DRAM (J831, REF DESIG PRINTED ON CKT BRD)	0JR04	THM321000AS–70
22	950–2837–00			1	HEX NUT M2.5 ZINC	62559	950–2837–00
23	950–1001–00			1	PCB FRONT PANEL HOLDER	62559	950–1001–00
24	950–3794–00			2	92505–25005:WASHER WAVY 2.7MM	80009	950–3794–00
25	950–4448–00			2	SCREW M2.5X10 CHEESEHEAD	80009	950–4448–00
26	348–1498–00			1	GASKET EMI:ESD GASKET	0KM03	0493008900
27	950–1008–00			1	SCREW M 2.5 X 8 SELF TAP	62559	950–1008–00
28	367–0410–00			1	HANDLE,EJECTOR:BOTTOM,SINGLE WIDE MODULE	62559	20817–327

Replaceable Parts List (Cont.)

Fig. & Index Number	Tektronix Part Number	Serial No. Effective	Serial No. Discont'd	Qty	Name & Description	Mfr. Code	Mfr. Part Number
29	950–4827–00			3	SCREW PHIL M 2.5 X 8 CSK	0KB01	950–4827–00
30	334–9015–00			1	MARKER,IDENT:LOWER LABEL,MASS STORAGE	80009	334–9015–00
31	211–0408–00			2	SCR,ASSEM WSHR:4–40 X 0.250,T–10 TORX DR	0KB01	211–0408–00
32	367–0411–00			1	HANDLE,EJECTOR:TOP,SINGLE WIDE	62559	20817–328
33	334–9016–00			1	MARKER,IDENT:UPPER LABEL	80009	334–9016–00
34	333–4178–00			1	PANEL,FRONT	80009	333–4178–00
35	129–1451–00			2	STANDOFF, 11 MM L,2.5MM X 0.45 THD,ALUM	06540	19058–A–2545–0
36	348–1365–01			1	SHLD GSKT,ELEC:SLOTTED FINGER,SNAP–IN	TK2647	348–1365–01
37	159–0245–00			1	FUSE,WIRE LEAD:1A,125V,FAST (F20, REF DESIG PRINTED ON CKT BRD)	71400	TR/MCR–1
					STANDARD ACCESSORIES		
	063–1940–00			1	SOFTWARE PKG: SYSTEM MANAGEMENT AND CONTROL,V1.0,3.5 INCH DISKETTE	80009	063–1940–00
	063–1943–00			1	SOFTWARE PKG:PLUG AND PLAY VER. 1.0 FOR SOFT FRONT PANEL 3.5 INCH DISKETTE	80009	063–1943–00
	070–9259–XX			1	MANUAL,TECH:USER	80009	070–9259–XX

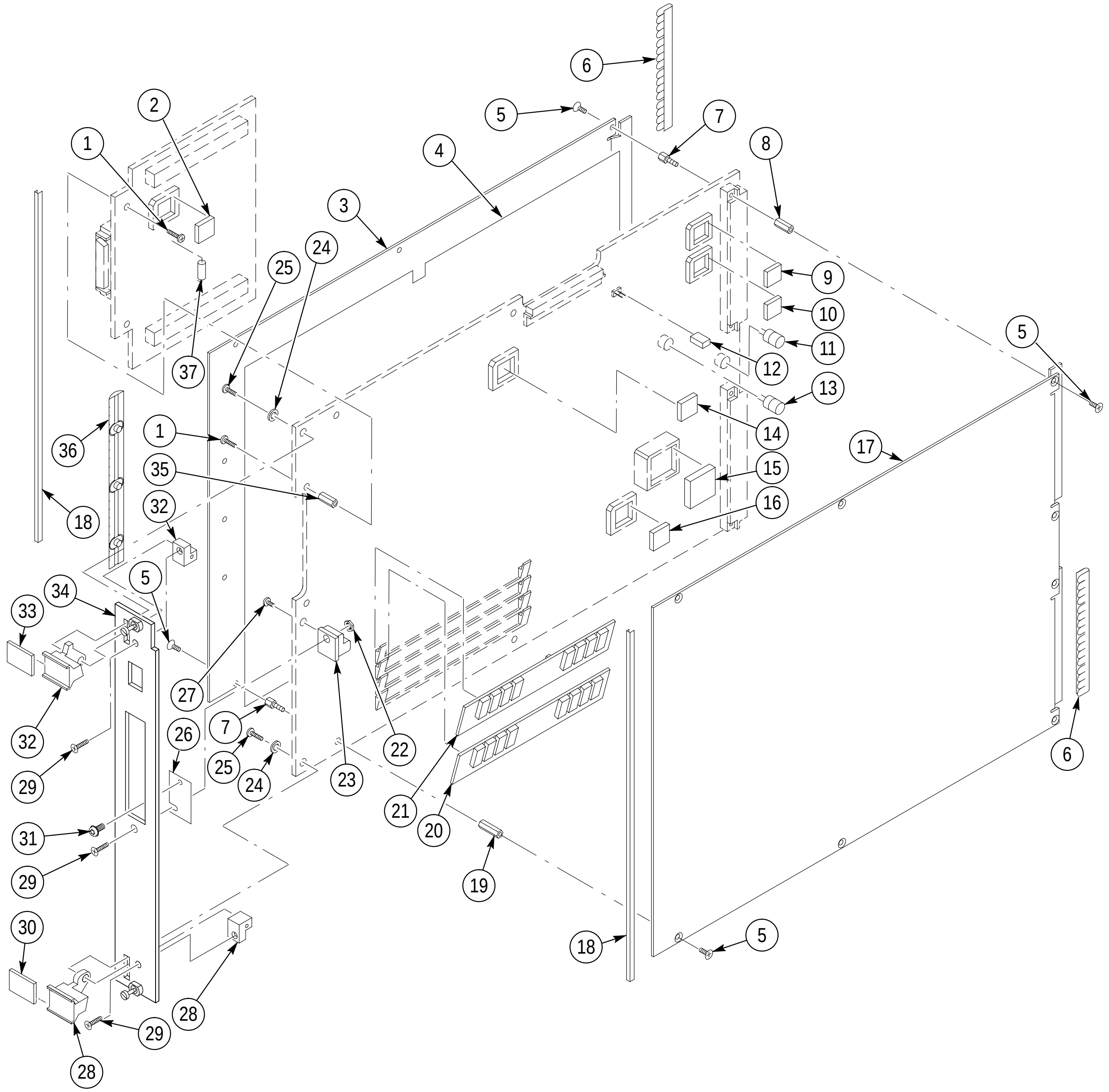


Figure G–1: VX4570 Exploded View

Glossary and Index

Glossary

The terms in this glossary are defined as used in the VXIbus System. Although some of these terms may have different meanings in other systems, it is important to use these definitions in VXIbus applications. Terms which apply only to a particular instrument module are noted. Not all terms appear in every manual.

Accessed Indicator
An amber LED indicator that lights when the module identity is selected by the Resource Manager module, and flashes during any I/O operation for the module.

ACFAIL*
A VMEbus backplane line that is asserted under these conditions: 1) by the mainframe Power Supply when a power failure has occurred (either AC line source or power supply malfunction), or 2) by the front panel ON/STANDBY switch when switched to STANDBY.

A-Size Card
A VXIbus instrument module that is 100.0 × 160 mm × 20.32 mm (3.9 × 6.3 in × 0.8 in), the same size as a VMEbus single-height short module.

Asynchronous Communication
Communications that occur outside the normal "command-response" cycle. Such communications have higher priority than synchronous communication.

Backplane
The printed circuit board that is mounted in a VXIbus mainframe to provide the interface between VXIbus modules and between those modules and the external system.

B-Size Card
A VXIbus instrument module that is 233.4 × 160 mm × 20.32 mm (9.2 × 6.3 in × 0.8 in), the same size as a VMEbus double-height short module.

Bus Arbitration
In the VMEbus interface, a system for resolving contention for service among VMEbus Master devices on the VMEbus.

Bus Timer
A functional module that measures the duration of each data transfer on the Data Transfer Bus (DTB) and terminates the DTB cycle if the duration is excessive. Without the termination capability of this module, a Bus Master attempt to transfer data to or from a non-existent Slave location could result in an infinitely long wait for the Slave response.

Client

In shared memory protocol (SMP), that half of an SMP channel that does not control the shared memory buffers.

CLK10

A 10 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1–12 on P2. It is distributed to each module slot as a single source, single destination signal with a matched delay of under 8 ns.

CLK100

A 100 MHz, ±100 ppm, individually buffered (to each module slot), differential ECL system clock that is sourced from Slot 0 and distributed to Slots 1–12 on P3. It is distributed to each module slot in synchronous with CLK10 as a single source, single destination signal with a maximum system timing skew of 2 ns, and a maximum total delay of 8 ns.

Commander

In the VXIbus interface, a device that controls another device (a servant). A commander may be a servant of another commander.

Command

A directive to a device. There are three types of commands:

In Word Serial Protocol, a 16-bit imperative to a servant from its commander.

In Shared Memory Protocol, a 16-bit imperative from a client to a server, or vice versa.

In a Message, an ASCII-coded, multi-byte directive to any receiving device.

Communication Registers

In word serial protocol, a set of device registers that are accessible to the commander of the device. Such registers are used for inter-device communications, and are required on all VXIbus message-based devices.

Configuration Registers

A set of registers that allow the system to identify a (module) device type, model, manufacturer, address space, and memory requirements. In order to support automatic system and memory configuration, the VXIbus standard specifies that all VXIbus devices have a set of such registers, all accessible from P1 on the VMEbus.

C-Size Card

A VXIbus instrument module that is 340.0 mm × 233.4 mm × 30.48 mm (13.4 in. × 9.2 in × 1.2 in).

Custom Device
A special-purpose VXIbus device that has configuration registers so as to be identified by the system and to allow for definition of future device types to support further levels of compatibility.

Data Transfer Bus
One of four buses on the VMEbus backplane. The Data Transfer Bus allows Bus Masters to direct the transfer of binary data between Masters and Slaves.

DC SUPPLIES Indicator
A red LED indicator that illuminates when a DC power fault is detected on the backplane.

Device Specific Protocol
A protocol for communication with a device that is not defined in the VXIbus specification.

D-Size Card
A VXIbus instrument module that is 340.0 × 366.7 mm × 30.48 mm (13.4 × 14.4 in × 1.2 in).

DTB
See Data Transfer Bus.

DTB Arbiter
A functional module that accepts bus requests from Requester modules and grants control of the DTB to one Requester at a time.

DUT
Device Under Test.

ECLTRG
Six single-ended ECL trigger lines (two on P2 and four on P3) that function as inter-module timing resources, and that are bussed across the VXIbus subsystem backplane. Any module, including the Slot 0 module, may drive and receive information from these lines. These lines have an impedance of 50 Ω; the asserted state is logical High.

Embedded Address
An address in a communications protocol in which the destination of the message is included in the message.

ESTST
Extended STart/STop protocol; used to synchronize VXIbus modules.

Extended Self Test
Any self test or diagnostic power-on routine that executes after the initial kernel self test program.

External System Controller

The host computer or other external controller that exerts overall control over VXIbus operations.

FAILED Indicator

A red LED indicator that lights when a device on the VXIbus has detected an internal fault. This might result in the assertion of the SYSFAIL* line.

IACK Daisy Chain Driver

The circuit that drives the VMEbus Interrupt Acknowledge daisy chain line that runs continuously through all installed modules or through jumpers across the backplane.

ID-ROM

An NVRAM storage area that provides for non-volatile storage of diagnostic data.

Instrument Module

A plug-in printed circuit board, with associated components and shields, that may be installed in a VXIbus mainframe. An instrument module may contain more than one device. Also, one device may require more than one instrument module.

Interface Device

A VXIbus device that provides one or more interfaces to external equipment.

Interrupt Handler

A functional module that detects interrupt requests generated by Interrupters and responds to those requests by requesting status and identity information.

Interrupter

A device capable of asserting VMEbus interrupts and performing the interrupt acknowledge sequence.

IRQ

The Interrupt ReQuest signal, which is the VMEbus interrupt line that is asserted by an Interrupter to signify to the controller that a device on the bus requires service by the controller.

Local Bus

A daisy-chained bus that connects adjacent VXIbus slots.

Local Controller

The instrument module that performs system control and external interface functions for the instrument modules in a VXIbus mainframe or several mainframes. See Resource Manager.

Local Processor

The processor on an instrument module.

Logical Address

The smallest functional unit recognized by a VXIbus system. It is often used to identify a particular module.

Mainframe

Card Cage. For example, the Tektronix VX1400 Mainframe, an operable housing that includes 13 C-size VXIbus instrument module slots.

Memory Device

A storage element (such as bubble memory, RAM, and ROM) that has configuration registers and memory attributes (such as type and access time).

Message

A series of data bytes that are treated as a single communication, with a well defined terminator and message body.

Message Based Device

A VXIbus device that supports VXI configuration and communication registers. Such devices support the word serial protocol, and possibly other message-based protocols.

MODID Lines

Module/system identity lines.

Physical Address

The address assigned to a backplane slot during an access.

Power Monitor

A device that monitors backplane power and reports fault conditions.

P1

The top-most backplane connector for a given module slot in a vertical mainframe such as the Tektronix VX1400. The left-most backplane connector for a given slot in a horizontal mainframe.

P2

The bottom backplane connector for a given module slot in a vertical C-size mainframe such as the VX1400; or the middle backplane connector for a given module slot in a vertical D-size mainframe such as the VX1500.

P3

The bottom backplane connector for a given module slot in a vertical D-size mainframe such as the Tektronix VX1500.

Query

A form of command that allows for inquiry to obtain status or data.

READY Indicator

A green LED indicator that lights when the power-on diagnostic routines have been completed successfully. An internal failure or failure of +5 V power will extinguish this indicator.

Register Based Device
A VXIbus device that supports VXI register maps, but not high level VXIbus communication protocols; includes devices that are register-based servant elements.

Requester
A functional module that resides on the same module as a Master or Interrupt Handler and requests use of the DTB whenever its Master or Interrupt Handler requires it.

Resource Manager
A VXIbus device that provides configuration management services such as address map configuration, determining system hierarchy, allocating shared system resources, performing system self test diagnostics, and initializing system commanders.

Self Calibration
A routine that verifies the basic calibration of the instrument module circuits, and adjusts this calibration to compensate for short- and long-term variables.

Self Test
A set of routines that determine if the instrument module circuits will perform according to a given set of standards. A self test routine is performed upon power-on.

Servant
A VXIbus message-based device that is controlled by a commander.

Server
A shared memory device that controls the shared memory buffers used in a given Shared Memory Protocol channel.

Shared Memory Protocol
A communications protocol that uses a block of memory that is accessible to both client and server. The memory block operates as a message buffer for communications.

Slot 0 Controller
See Slot 0 Module. Also see Resource Manager.

Slot 0 Module
A VXIbus device that provides the minimum VXIbus slot 0 services to slots 1 through 12 (CLK10 and the module identity lines), but that may provide other services such as CLK100, SYNC100, STARBUS, and trigger control.

SMP
See Shared Memory Protocol.

STARX

Two (2) bi-directional, 50 Ω, differential ECL lines that provide for inter-module asynchronous communication. These pairs of timed and matched delay lines connect slot 0 and each of slots 1 through 12 in a mainframe. The delay between slots is less than 5 ns, and the lines are well matched for timing skew.

STARY

Two (2) bi-directional, 50 Ω, differential ECL lines that provide for inter-module asynchronous communication. These pairs of timed and matched delay lines connect slot 0 and each of slots 1 through 12 in a mainframe. The delay between slots is less than 5 ns, and the lines are well matched for timing skew.

STST

STart/STop protocol; used to synchronize modules.

SYNC100

A Slot 0 signal that is used to synchronize multiple devices with respect to a given rising edge of CLK100. These signals are individually buffered and matched to less than 2 ns of skew.

Synchronous Communications

A communications system that follows the “command-response” cycle model. In this model, a device issues a command to another device; the second device executes the command; then returns a response. Synchronous commands are executed in the order received.

SYSFAIL*

A signal line on the VMEbus that is used to indicate a failure by a device. The device that fails asserts this line.

System Clock Driver

A functional module that provides a 16 MHz timing signal on the Utility Bus.

System Hierarchy

The tree structure of the commander/servant relationships of all devices in the system at a given time. In the VXIbus structure, each servant has a commander. A commander may also have a commander.

Test Monitor

An executive routine that is responsible for executing the self tests, storing any errors in the ID-ROM, and reporting such errors to the Resource Manager.

Test Program

A program, executed on the system controller, that controls the execution of tests within the test system.

Test System
A collection of hardware and software modules that operate in concert to test a target DUT.

TTLTRG
Open collector TTL lines used for inter-module timing and communication.

VXIbus Subsystem
One mainframe with modules installed. The installed modules include one module that performs slot 0 functions and a given complement of instrument modules. The subsystem may also include a Resource Manager.

Word Serial Protocol
A VXIbus word oriented, bi-directional, serial protocol for communications between message-based devices (that is, devices that include communication registers in addition to configuration registers).

Word Serial Communications
Inter-device communications using the Word Serial Protocol.

WSP
See Word Serial Protocol.

10-MHz Clock
A 10 MHz, ±100 ppm timing reference. Also see CLK10.

100-MHz Clock
A 100 MHz, ±100 ppm clock synchronized with CLK10. Also see CLK100.

488-To-VXIbus Interface
A message based device that provides for communication between the IEEE-488 bus and VXIbus instrument modules.

Index