Errata

Title & Document Type: E1750A Broadband Distribution Amplifier User’s Guide

Manual Part Number: E1750-90001

Revision Date: July 1993

User’s Guide

Enclosed is the User’s Guide for the HP E1750A Broadband Distribution Amplifier. Insert this guide, plus any other VXIbus guides (or manuals) that you have, into the binder that came with your Hewlett-Packard mainframe.

HP 75000 SERIES C

HP E1750A Broadband Distribution Amplifier

barcode goes here

User’s Guide

This guide describes how to use the HP E1750A Broadband Distribution Amplifier. The information in this guide applies to instruments having the number prefix listed below, unless accompanied by a "Manual Updating Changes" package indicating otherwise.

SERIAL PREFIX NUMBER: 3325 and above

HP 75000 SERIES C

HP E1750A Broadband Distribution Amplifier

Contents

1

Using the HP E1750A

HP E1750A Faceplate Features

Figure 1-1. HP E1750A Faceplate

HP E1750A Faceplate Features

These are the descriptions of the numbered features in Figure 1-1.

1 Input LED. Lights when a 100 kHz to 10 MHz sinusoidal signal is applied. Input power of +7 to +19 dBm is recommended.

2 INPUT connector. BNC (female)

3 Output LEDs. Lights when a 100 kHz to 10 MHz sinusoidal output signal is present.

4 Output connectors. BNC (female)

Figure 1-2. HP E1750A Block Diagram

How the HP E1750A Works

Refer to Figure 1-2 when reading the following explanation.

A sinusoidal signal at a frequency of 100 kHz to 10 MHz with a power level of +7 to +19 dBm is applied to the input. A 50 ohm resistor across the input establishes a nominal 50 ohm input impedance.

The input signal is processed by an Automatic Gain Control (AGC) loop to produce a constant amplitude signal ($V_{leveled}$) at the output of the wideband amplifier. The AGC loop consists of a resistive input attenuator, a wideband amplifier, a precision level detector, a voltage reference (which sets the $V_{leveled}$ level), an integrating amplifier, and a LED/photoconductor unit. The photoconductor section forms part of the input attenuator. To maintain $V_{leveled}$ constant as the input power varies, the AGC loop forces the photoconductor resistance to change inversely to the input voltage level. The result is that larger input signal are attenuated more, and smaller input signals are attenuated less.

The faceplate LEDs indicate the presence of the input signal and each of the output signals. The fault logic circuit, in response to the peak detector circuits indicating a loss on the input and one or more of the outputs, sends a fault flag to the VXI interface circuit. This, in turn issues an interrupt request to the VXI bus. The status of the input and output signals can be read over the VXI bus at the discretion of the VXI bus controller or in response to a fault interrupt request of the HP E1750A.

Warnings and Cautions

WARNING

SHOCK HAZARD. ONLY SERVICE-TRAINED PERSONNEL WHO ARE AWARE OF THE HAZARDS INVOLVED SHOULD INSTALL, REMOVE, OR CONFIGURE THE HP E1750A. BEFORE YOU REMOVE ANY INSTALLED MODULE, DISCONNECT AC POWER FROM THE MAINFRAME AND FROM OTHER MODULES THAT MAY BE CONNECTED TO THE HP E1750A.

CAUTION

MAXIMUM INPUTS. The maximum voltage that can be applied to the faceplate INPUT terminal is 5 Vrms (+27 dBm) ± 40 Vdc.

Always check that the mainframe power is OFF before installing or removing any board or instruments into the backplane connectors.

STATIC ELECTRICITY. Static electricity is a major cause of component failure. To prevent damage to the electrical components in the HP E1750A, observe anti-static techniques whenever removing a module from the mainframe or whenever working on a module.

Setting the Address Switch

The HP E1750A logical address switch (LADDR) factory setting is 8. You may have changed the setting during module installation. Valid address values are from 1 to 255. If the HP E1750A is used with an HP E1406A Command Module in a C-Size Mainframe, refer to the "*HP E1406A Command Module User's Guide*" for addressing information. Otherwise, use Figure 1-3 to set the logical address switch.

NOTE

The address switch selected value must be a multiple of 8 if the module is used with a VXIbus Command Module, and being instructed by SCPI commands.

Figure 1-3. Logical Address Switch

Selecting the Interrupt Priority

The HP E1750A module generates an interrupt when the status register transitions from a valid signal condition on both the input and six outputs to an invalid signal condition. These interrupts are sent to, and acknowledgments are received from, the HP E1406A Command Module via the VXIbus backplane interrupt lines.

For most applications where the HP E1750A module is installed in an HP 75000 Series C mainframe, the interrupt priority jumper does not have to be moved. This is because the VXIbus interrupt lines have the same priority, and interrupt priority is established by installing modules in slots numerically closest to the HP E1406A Command Module. Thus, Slot 1 has a higher priority than Slot 2; Slot 2 has a higher priority than Slot 3, etc.

Refer to Figure 1-4 to change the interrupt priority. You can select seven different interrupt priority levels. Level 1 is the lowest priority and Level 7 is the highest priority. The Module's factory setting is Level 1. To change, remove both 2-pin jumpers from the old priority locations and reinstall in the new priority locations The two jumper locations must have the same interrupt priority level selected.

NOTE

The interrupt priority jumper should be installed in position 1 when using the HP E1406A Command Module. Changing the priority level jumper is not recommended. Do not change unless specifically instructed to do so.

Figure 1-4. Interrupt Jumpers

Installing and Removing the HP E1750A

The HP E1750A can be installed in any "C" mainframe Slot except Slot 0. The HP E1406A Command Module should be installed in
Slot 0.

To Install the Module

CAUTION

Always check that the mainframe power is OFF before installing or removing any board or instruments into the backplane connectors.

1 Inspect the pins on the back of the module for straightness before inserting into the mainframe.

2 Install the module by lining up the module with the Slot guides and then pressing the module firmly into the backplane connectors. See Figure 1-5.

NOTE

The screws located at the top and bottom of the module can be used to secure the installed module to the mainframe.

To Remove the Module

CAUTION

Always check that the mainframe power is OFF before installing or removing any board or instruments into the backplane connectors.

Remove the captive screws. Remove the module by using the faceplate thumb tabs as shown in Figure 1-5. This extracts the module from the backplane connectors, allowing the module to be removed.

Figure 1-5. Installing/Removing the HP E1750A

Operation Check

The purpose of these procedures is to check for functionality of the HP E1750A. To verify the specifications given in Chapter 5, see "Performance Verification" in this chapter.

The following procedures assume you have followed the instructions for selecting the address and priority interrupts in the first part of this chapter and have installed the HP E1750A into an appropriate "C" mainframe with the HP E1406A Command Module installed into Slot 0.

Figure 1-6. Operation Check Setup

Equipment

- Signal Generator: HP 8663A
- Oscilloscope: HP 54503A
- BNC Short

CAUTION

Input amplitude at the HP E1750A INPUT must be below 5Vrms (+27 dBm) or damage to the Amplifier will occur.

Procedure

1 Apply power to the mainframe.

2 Set the Signal Generator to the following conditions:
Frequency: 10 MHz
Wave type: Sinusoidal
Amplitude: +10 dBm

3 Connect the equipment as shown in Figure 1-6 with the Signal Generator connected to the HP E1750A input.

NOTE Once your HP E1750A Broadband Distribution Amplifier is installed and the VXI mainframe power is ON, all that is required for operation is a signal to the HP E1750A input.

4 Verify that the input and all six output LEDs are lit.

5 Set the oscilloscope to the following:
Channel 1: 0.5 V/div
Timebase: 50 ns/div
Input Impedance: 50 ohms

6 Connect the oscilloscope to each of the output ports to verify that the output signal is a 1 Vrms (2.8 Vp-p) sinewave at 10 MHz.

7 Disconnect the signal from the HP E1750A INPUT.

8 Verify all the LEDs are off. This indicates loss of input.

Status Register Check

1 Reconnect the signal input to the HP E1750A.

NOTE A programming example to read the id, device and status registers is located with other programming examples in Chapter 2. Refer to the command module manual for interface instructions.

2 Use the controller/interface to read the status register. The command string that will read the status register is:

```
VXI:READ? 8,4;
Where "8" is the logical address
and "4" is the status register
```

3 The response from the status register to indicate all ports are active should be:
20460

4 Remove the signal from the HP E1750A input. Front panel LEDs should be off.

5 Query the status register, again.

6 The response from the status register to indicate all ports are inactive should be:
16396

Short Outputs Test

1 Replace the input signal to the HP E1750A.

2 Place a BNC short on the first output. The LED should be off. Query the status register.

3 Repeat step two for all six Outputs.

4 If all the LEDs function properly, the output signals, and the status register function as expected, the HP E1750A is operational.

Performance Verification

The procedures detailed here test the specified electrical performance of the HP E1750A Broadband Distribution Amplifier. These tests do not require access to the interior of the instrument. These tests do not test typical or supplementary characteristics.

If the performance tests are to be considered valid, the following conditions must be met.

- Module being tested is plugged into a VXI mainframe, and the mainframe turned on.
- Test conducted in the order presented in this section.
- Tests conducted under normal operating conditions as stated in Chapter 5, Specifications.
- Test equipment selected as listed in each test according to the Recommended Test Equipment Table.

Phase Noise Test

Detailed procedures for the phase noise portion of the performance tests are not in this guide. This test requires the HP 3048A Phase Noise Measurement system, a highly specialized test system. Instruction for performing phase noise tests can be found in the HP 3048A system documentation.

Test Record

The results of the performance tests may be recorded in the test records located at the end of this chapter.

Table 1-1. Recommended Test Equipment

Instrument	Required Characteristics	Use	Recommended Model
Low Pass Filter	Bandwidth:10 MHz Attenuation: ≥24 dB @ ≥20 MHz	Harmonic Output Test	Mini-Circuits™ Model BLP 10.7
Signal Generator	Frequency Range: 100 kHz to 10 MHz Output Level: +7 to +16 dBm Harmonic Distortion:<-30 dBc Spurious Distortion:<-80 dBc	Input Signal Source	HP 8663A
RMS Voltmeter	Accuracy : ±2% at 3V Full Scale, 1 MHz	Output Signal Level Test	HP 3400B
Spectrum Analyzer	Frequency Range Span: 0 Hz to 50 MHz Measurement Range @ 50Ω Impedance: +15 to -75 dBm Frequency Resolution Bandwidth: 10 Hz to 1 kHz	Output Signal Level and Harmonics	HP 8568A/B
50 ohm BNC Feedthrough		Output Level Test	HP 10100C
VXI "C" Mainframe		All Tests	HP E1400B

Input Frequency Response Test

The output signal at each of the six ports must be from
+13 dBm ± 1 dB (0.89 Vrms through 1.12 Vrms) into 50 ohms
for a varying input frequency from 100 kHz to 10 MHz. A +9 dBm signal from a Signal Generator is applied to the HP E1750A Input. The frequency of the input signal is changed from 100 kHz to 10 MHz while the six HP E1750A Outputs ports are verified for 0.89 to
1.12 Vrms.

Specifications Tested

HP E1750A Input Ranges

Minimum Frequency	100 KHz
Maximum Frequency	10 MHz

HP E1750A Outputs, 50Ω load

Level	0.89 Vrms to 1.12 Vrms

Equipment

- HP 8663A Signal Generator
- HP 3400B Analog True RMS Voltmeter
- HP E1400B VXI Mainframe
- HP 10100C 50Ω BNC Feedthrough

Figure 1-7. Input Frequency Response Test Setup

Procedure

1 Install the HP E1750A into a VXI "C" Mainframe and apply power.

2 Set the RF Signal Generator to:
FREQUENCY: 100 kHz
LEVEL: +9 dBm

3 Connect the RF Signal Generator to the HP E1750A Input.

4 Connect the RMS Voltmeter via the 50 ohm load to the HP E1750A Output 1.

5 Set the RMS voltmeter to 3V Full Scale.

6 Verify the level of this signal is between 0.89 to 1.12 Vrms at all six of the HP E1750A Outputs. Record this in the test record for this frequency.

7 Adjust the Signal Generator to:
FREQUENCY: 10 MHz

8 Verify the level of this signal is between 0.89 to 1.12 Vrms at all six of the HP E1750A Outputs. Record this in the test record for this frequency.

Input Level Response Test

A 1 MHz signal is input to the HP E1750A from a signal generator. The level of this signal is adjusted from +7 dBm to +16 dBm while the six Outputs are verified for a 0.89 to 1.12 Vrms signal.

Specifications Tested

HP E1750A Input Ranges

Minimum Level	+7 dBm
Maximum Level	+19 dBm

HP E1750A Outputs, 50Ω load

Level	+13 dBm ± 1 dB

Equipment

- HP 8663A Signal Generator
- HP 3400B RMS Voltmeter
- HP E1400B VXI Mainframe
- HP 10100C 50Ω BNC Feedthrough

Figure 1-8. Input Level Response Test Setup

Procedure

1 Install the HP E1750A into a VXI mainframe and apply power.

2 Set the signal generator to:
FREQUENCY: 1 MHz
LEVEL: +7 dBm

3 Connect the signal generator to the HP E1750A Input.

4 Connect the RMS voltmeter via the 50 ohm load to the HP E1750A Output 1.

5 Set the RMS voltmeter to 3V Full Scale Range.

6 Verify that the RMS voltage on each of the six Outputs is from 0.89 to 1.12 Vrms. Record this in the Test Record for this frequency.

7 Adjust the signal generator to:
LEVEL: +16 dBm

8 Verify that the RMS voltage on each of the six Outputs is from 0.89 to 1.12 Vrms. Record this in the Test Record for this frequency.

Harmonic Distortion and Spurious Signals Tests

HP E1750A introduced signal harmonics on the output must be less than 45 dBc below the fundamental. Spurious (Non-Harmonic) signals must be less than 80 dBc below the fundamental. To perform this verification, a spectrum analyzer is set to display a fundamental frequency and an amplitude reference is established. The output frequency spectrum is then examined on all six outputs to determine harmonic and spurious amplitude relationships of the fundamental.

Specification Tested

HP E1750A Outputs, 50Ω load

Harmonic Distortion	< –45 dBc 100 kHz to 5 MHz < –40 dBc 5.01 MHz to 10 MHz
Spurious Phase Modulation	< –80 dBc discrete sidebands, 20 Hz to 50 kHz

Test Equipment

- HP 8663A Signal Generator
- HP 8568A/B Spectrum Analyzer
- HP E1400B VXI Mainframe
- Mini-Circuit™ BLP 10.7 Low Pass Filter

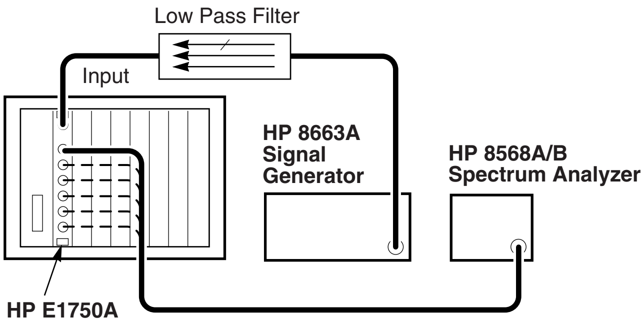


Figure 1-9. Harmonic Distortion and Spurious Test Setup

Procedure

1 Install the HP E1750A into a VXI Mainframe and apply power.

2 Set the signal generator to:
CENTER FREQUENCY: 5 MHz
LEVEL: +9 dBm

3 Connect the signal generator via low pass filter to the HP E1750A Input.

4 Connect the spectrum analyzer to the HP E1750A Output 1.

5 On the spectrum analyzer establish a reference level for the 5 MHz fundamental frequency. Make sure that there is no distortion due overdriving the spectrum analyzer input.

6 Set the spectrum analyzer to display the 2nd, 3rd, 4th and 5th harmonics of the 5 MHz fundamental frequency.

7 Analyze the 10, 15, 20, and 25 MHz harmonics at each of the six HP E1750A Outputs. Verify that all are at least 45 dB below the fundamental level. Record this in the Test Record for harmonics.

8 Set the signal generator to:
FREQUENCY: 10 MHz
LEVEL: +9 dBm

9 On the spectrum analyzer establish a reference level for the 10 MHz fundamental frequency. Make sure that there is no distortion due overdriving the spectrum analyzer input.

10 Analyze the 20, 30, 40 and 50 MHz harmonics of the 10 MHz signal at each of the six HP E1750A Outputs. Verify that all are at least 40 dB below the 10 MHz fundamental level. Record this in the Test Record for harmonics.

11 Set the spectrum analyzer to:

START FREQ: 9.95 MHz
STOP FREQ: 10.05 MHz
REFERENCE LEVEL: 15 dBm
VBW: 10 Hz
RBW: 10 Hz

12 Set a reference level at the peak of the 10 MHz signal. Find the peak of the highest spurious signal within 20 Hz to 50 kHz on both sides of the carrier. Verify that all signals are at least 80 dBc below the 10 MHz fundamental level. Record this in the Test Record for spurious signals.

HP E1750A Performance Test Record (Page 1 of 4)

Hewlett-Packard Model E1750A
Broadband Distribution Amplifier

Serial Number: Repair/Work Order No.

Test Performed By: Temperature:

Date: Relative Humidity:

Notes:

Input Frequency Response Test

+9 dBm @ 100 kHz

0.89 Vrms <	1.	< 1.12 Vrms
0.89 Vrms <	2.	< 1.12 Vrms
0.89 Vrms <	3.	< 1.12 Vrms
0.89 Vrms <	4.	< 1.12 Vrms
0.89 Vrms <	5.	< 1.12 Vrms
0.89 Vrms <	6.	< 1.12 Vrms
+9 dBm @ 10 MHz		
0.89 Vrms <	1.	< 1.12 Vrms
0.89 Vrms <	2.	< 1.12 Vrms
0.89 Vrms <	3.	< 1.12 Vrms
0.89 Vrms <	4.	< 1.12 Vrms
0.89 Vrms <	5.	< 1.12 Vrms
0.89 Vrms <	6.	< 1.12 Vrms

HP E1750A Performance Test Record
(Page 2 of 4)

Input Level Response Test **+7 dBm @ 1 MHz**		
0.89 Vrms <	1.	< 1.12 Vrms
0.89 Vrms <	2.	< 1.12 Vrms
0.89 Vrms <	3.	< 1.12 Vrms
0.89 Vrms <	4.	< 1.12 Vrms
0.89 Vrms <	5.	< 1.12 Vrms
0.89 Vrms <	6.	< 1.12 Vrms
+16 dBm @ 1 MHz		
0.89 Vrms <	1.	< 1.12 Vrms
0.89 Vrms <	2.	< 1.12 Vrms
0.89 Vrms <	3.	< 1.12 Vrms
0.89 Vrms <	4.	< 1.12 Vrms
0.89 Vrms <	5.	< 1.12 Vrms
0.89 Vrms <	6.	< 1.12 Vrms

HP E1750A Performance Test Record (Page 3 of 4)

Harmonic Distortion Test					
5 MHz, +9 dBm Input	10 MHz 2nd Harm.	15 MHz 3rd Harm.	20 MHz 4th Harm.	25 MHz 5th Harm.	Specification
	1.	1.	1.	1.	<-45 dBc
	2.	2.	2.	2.	
	3.	3.	3.	3.	
	4.	4.	4.	4.	
	5.	5.	5.	5.	
	6.	6.	6.	6.	
10 MHz, +9 dBm Input	20 MHz 2nd Harm.	30 MHz 3rd Harm.	40 MHz 4th Harm.	50 MHz 5th Harm.	Specification
	1.	1.	1.	1.	<-40 dBc
	2.	2.	2.	2.	
	3.	3.	3.	3.	
	4.	4.	4.	4.	
	5.	5.	5.	5.	
	6.	6.	6.	6.	

HP E1750A Performance Test Record (Page 4 of 4)

Spurious signals 20 Hz to 50 kHz other than Harmonic		
10 MHz, 9 dBm Input	1.	<-80 dBc
	2.	
	3.	
	4.	
	5.	
	6.	

2

Programming the HP E1750A

- About this Chapter
- Programming Overview
- Command Types
- HP E1750A SCPI Status Reporting Structure
- Register-Based Programming
- Register Descriptions
- Programming Examples

About this Chapter

This chapter contains the information about the programming command structure of the HP E1750A Broadband Distribution Amplifier. For an alphabetical listing of IEEE 488.2 common commands and SCPI commands, see Chapter 3, Command Reference.

Programming Overview

Programming the HP E1750A Broadband Distribution Amplifier is limited to controlling the SCPI Status System and querying the state of the one input and six outputs. Therefore, only two SCPI commands are necessary to query the input and outputs. These two commands are:

```
DIAGnostic:INPut?

DIAGnostic:OUTput[<output_channel>]?
```

Programming examples are given at the end of this chapter for easy reference. There are SCPI, C-SCPI, and Register-based programming examples.

Command Types

Commands are separated into two types: IEEE 488.2 Common Commands and SCPI Subsystem Commands.

Common Command Format

The IEEE 488.2 standard defines the common commands that perform functions like reset, self-test, status byte query, etc. Common commands are four or five characters in length, always begin with the asterisk character (*), and may include one or more parameters. The command keyword is separated from the first parameter by a space character. Some examples of Common Commands are shown below:

```
*RST      *CLS      *STB?
```

SCPI Subsystem Command Format

The SCPI subsystem commands perform functions like querying the status of the input and outputs, or retrieving data. A subsystem command structure is a hierarchical structure that in this case consists of a first level command, one or more lower level sub commands, and their parameters. The following example shows part of a typical subsystem:

Example:

```
DIAGnostic:OUTput5?
```

Where "DIAGnostic" is the first level command with "OUTput" as the second level subcommand and the "5" as an optional parameter.

Command Separator

A colon (:) always separates one command from the next lower level command as shown below:

```
DIAGnostic:INPut?
```

Colons separate the first level command (DIAGnostic), and the second level query (INPut?).

Abbreviated Commands

The command syntax shows most commands as a mix of upper and lower case letters. The upper case letters indicate the abbreviated spelling for the command. For shorter program lines, send only the abbreviated form. For better program readability, you may send the entire command. The instrument will only accept either the abbreviated form or the entire command. For example, if the command syntax shows DIAGnostic, then DIAG and DIAGNOSTIC are both acceptable forms. Other forms of DIAGnostic, such as DI or DIA will generate an error. You may use upper or lower case letters. Therefore, DiAgNoStIc is acceptable.

Parameters

The following table contains explanations and examples of parameter types you might see later in this chapter.

Parameter Types, Explanations and Examples

Numeric	Accepts all commonly used representations of whole numbers.
Boolean	Represents a single binary condition that is either "1" or "0".

Optional Parameters

Parameters shown within square brackets ([ ]) are optional parameters. (Note that the brackets are not part of the command and are not sent to the instrument.) If you do not specify a value for an optional parameter, the instrument chooses a default value. For example, consider the command:

```
    DIAGnostic:OUTput[<output_channel>]?
```

If you send the command without specifying the output_channel parameter:

```
DIAG:OUT?
```

the HP E1750A will return the a composite of all six outputs.

If you send the following command:

```
DIAG:OUT5?
```

the HP E1750A will return the status of output 5.

Linking Commands

When linking IEEE 488.2 Common Commands with SCPI Commands, use a semicolon between commands. When linking multiple SCPI commands use both a semicolon and a colon between commands. For example:

```
*RST;DIAG:OUT?;:DIAG:INP?
```

SCPI also allows several commands within the same subsystem to be linked with a semicolon. For example:

```
DIAG:OUT?;INP?
```

HP E1750A SCPI Status Reporting Structure

The HP E1750A status registers conform to the SCPI and IEEE 488.2 standards. Figure 2-1 shows all the status system register groups and queues in the HP E1750A. Refer to this figure for the commands that are recognized by each of the status registers.

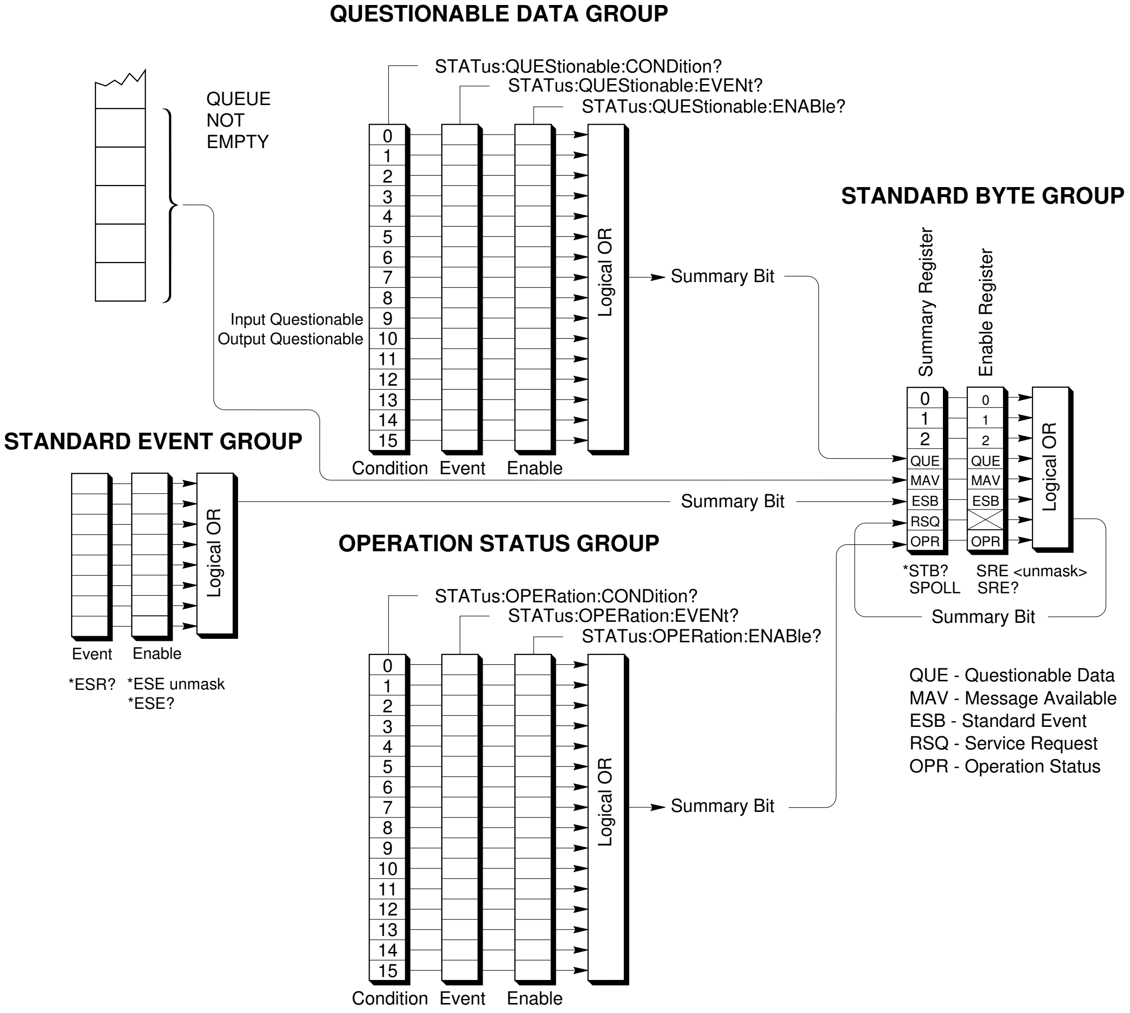


Figure 2-1. HP E1750A Status Reporting Summary Functional Diagram

Standard Byte Group

The Standard Byte Group contains a Summary Register which is the summary-level register in the status reporting structure. It contains summary bits that monitor activity in the other status registers and queues as shown in Figure 2-1. The Standard Byte Summary Register is a live register—its summary bits are set TRUE or FALSE (one or zero) by the presence or absence of the condition which is being summarized. See Table 2-2.

Table 2-2. Status Byte Summary Register

Bit	Weight	Symbol	Description
0 - 2			Not used
3	8	QUE	Questionable Data/Signal Status Register Summary Bit. This bit indicates whether or not one or more of the enabled Questionable Status events have occurred since the last reading or clearing of the Questionable Status Event Register.
4		MAV	Unused - Always 0
5	32	ESB	Standard Event Status Register Summary Bit. This bit indicates whether or not one of the enabled Event Status Register events have occurred since the last reading or clearing of the Standard Event Status Register.
6	64	RSQ	Request Status Summary When a serial poll is used to read the Status Byte Register, the RSQ bit indicates if the device was sending SRQ TRUE. The RSQ bit is set FALSE by a serial poll.
7		OPR	Always 0

Questionable Data Group

The Questionable Data Group contains a status register which monitors SCPI defined conditions.

Table 2-3. Questionable Status Register

Bit	Weight	Description
0 - 8		Not used
9	512	Input: 0=signal present 1= signal absent
10	1024	Output: 0= signal present 1= signal absent
11 - 15		Not used

Standard Event Group

The Standard Event Group contains status registers containing bits that monitor specific IEEE 488.2 defined events as shown in Figure 2-3.

Table 2-4. Standard Event Status Register

Bit	Weight	Symbol	Description
0		OPC	Operation Complete (always 0)
1		RQC	Request Control (always 0)
2	4	QYE	Query Error Indicates that either 1) an attempt was made to read the Output Queue when it was empty or 2) data in the Output Queue has been lost.

Table 2-4. Standard Event Status Register (continued)

Bit	Weight	Symbol	Description
3	8	DDE	Device Dependent Error. Indicates an operation id not properly complete due to some condition of the HP E1750A.
4	16	EXE	Execution Error. Indicates that a command could not be executed 1) because the parameter was out of range or inconsistent with the HP E1750A capabilities, or 2) because of some condition of the HP E1750A.
5	32	CME	Command Error. Indicates one of the following has occurred: 1) an IEEE 488.2 syntax error, 2) a semantic error indicating an unrecognized command.
6		(URQ)	User Request (always 0)
7	128	PON	Power ON (is "1" when first turned on)

Operation Status Group

In order to comply with SCPI standard the Operation Status Group is present, but not used.

Register-Based Programming

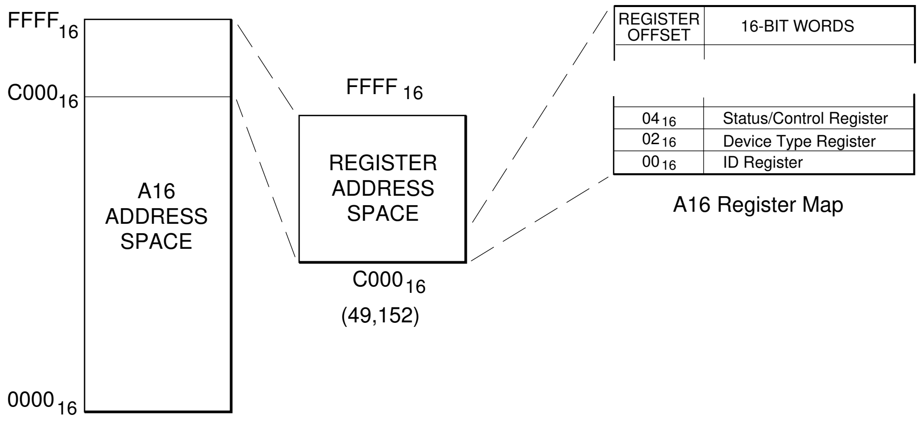
The HP E1750A is a register-based module which does not support the VXIbus word serial protocol. When a SCPI command is sent to the HP E1750A, the HP E1406A Command Module parses the command and programs the HP E1750A at the register level.

Register-based programming is a series of reads directly to the HP E1750A registers. This increases throughput speed since it eliminates command parsing and allows the use of an embedded controller. Also if Slot 0, the resource manager, and the computer
(HP-IB) interface are provided by other devices, a C-size system can be downsized by removing the Command Module.

Addressing the Registers

Register addresses for register-based devices are located in the upper 25% of VXI A16 address space. Every VXI device (up to 256) is allocated a 64-byte block of addresses. With three registers, the HP E1750A uses three of the 64 addresses allocated.

Figure 2-2 shows the register address location within A16. Figure 2-3 shows the location of A16 address space in the HP E1406A Command Module.



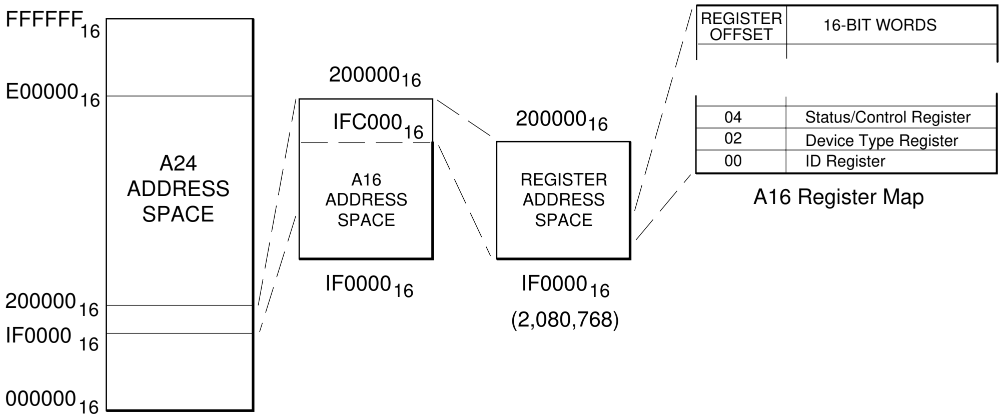
REGISTER OFFSET	16-BIT WORDS
04_{16}	Status/Control Register
02_{16}	Device Type Register
00_{16}	ID Register

A16 Register Map

Base Address = $C000_{16}$ + (logical address x $64)_{16}$
or
49,152 + (logical address x $64)_{10}$

Register Address = Base Address + Register Offset

Figure 2-2. HP E1750A Registers within A16 Address Space



Base Address = IFC000$_{16}$ + (logical address x 64)$_{16}$
or
2,080,768 + (logical address x 64)$_{10}$

Register Address = Base Address + Register Offset

Figure 2-3. Mainframe Command Module A16 Address Space

The Base Address

When you are reading a HP E1750A register, a hexadecimal or decimal register address is specified. This address consists of a base address plus a register offset.

The base address used in register-based programming depends on whether the A16 address space is outside or inside the HP E1406A Command Module.

A16 Address Space Outside the Command Module

When the HP E1406A Command Module is not part of your VXIbus system , the HP E1750A's base address is computed as:

49,152 + (LADDR * 64)

where (49,152) is the starting location of the register addresses, LADDR is the HP E1750A's logical address, and 64 is the number of address bytes per VXI device. For example, the HP E1750A's factory set logical address is 8. If this address is not changed, the HP E1750A will have a base address of:

49,152 + (8 * 64) = 49,152 + 512 = 49,664

A16 Address Space Inside the Command Module or Mainframe

When the A16 address space is inside the HP E1406A Command Module , the HP E1750A's base address is computed as:

2,080,768 + (LADDR * 64)

where 2,080,768 is the starting location of the VXI A16 addresses, LADDR is the HP E1750A's logical address, and 64 is the number of address bytes per register-based device. Again, the HP E1750A's factory set logical address is 8. If this address is not changed, the HP E1750A will have a base address of:

2,080,768 + (8 * 64) = 2,080,768 + 512 = 2,081,280

Register Descriptions

You can read the following HP E1750A registers:

ID Registers

base + 00	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

ID Register (base + 00): Reading this register returns: **65535**. This shows Hewlett-Packard as the manufacturer and that the module is an A16 register based device.

Device Type Registers

base + 02	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	0	0	0	0	0	0	1	0	1	1	0	1	0	1	0

Device Type Register (base+02). Reading this register returns: **362**. This shows the device is the HP E1750A.

Status Register

base + 04	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	0	1	0	0	Out6	Out5	Out4	Out3	Out2	Out1	In	0	1	1	0	0

Bits 6, 7, 8, 9, 10, and 11 report the status of the six outputs.
Valid Signal = 1
Invalid Signal = 0.

Bit 5 reports the status of the input register.
Valid Signal = 1
Invalid Signal = 0.

For example, if the Status Register (base + 04) returns 20460, all HP E1750A output signals are valid.

Programming Examples

The examples in this section demonstrate how to program the HP E1750A.

BASIC / UX Interrupt

Prior to running the following program you need to down load HP E1750A SCPI drivers to the HP E1406A.

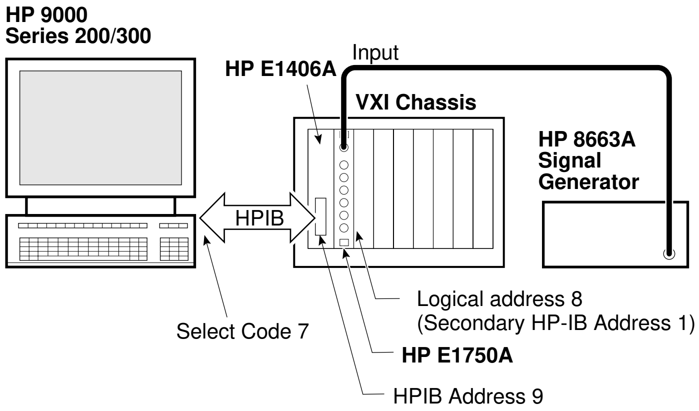


Figure 2-4. BASIC / UX Interrupt Example Setup

```
10    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20    ! This HPBASIC/UX program will set up the module to interrupt
30    ! on a transition from a valid state to an invalid state.
40    ! The program will then report which input/output is at
50    ! fault.
60    ! Asumptions:
70    !       HPIB address of Distribution Amp Module = 70901
80    !       Logical address of VXI module being read = 8
90    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
100   !
110   OUTPUT 70901;"*CLS"
120   OUTPUT 70901;"STAT:QUES:ENAB 1536"    ! Set up the enable masks
130   OUTPUT 70901;"*SRE 8"
140   ON INTR 7 CALL Err
150   ENABLE INTR 7;2
160   PRINT "Waiting for an interrupt..."
170   WHILE 1
180   ENABLE INTR 7;2                       ! enable the srq interrupt
190   END WHILE
200   END
210   !
220   ! This subroutine will read the input and output registers
230   !
240   SUB Err
250   PRINT "Received an interrupt"
260   OUTPUT 70901;"DIAG:INP?"
270   ENTER 70901;Result
280   IF Result=1 THEN
290     PRINT "FAIL: Input"
300   ELSE
310     FOR Output=1 TO 6
320       OUTPUT 70901;"DIAG:OUT"&VAL$(Output)&"?"
330       ENTER 70901;Result
340       IF Result=1 THEN PRINT "FAIL: Output "&VAL$(Output)
350     NEXT Output
360   END IF
370   PRINT
380   OUTPUT 70901;"*CLS"                   ! clear the status structures
390   SUBEND
```

BASIC / UX Register Programming Example

No down loaded drivers are needed to run this program.

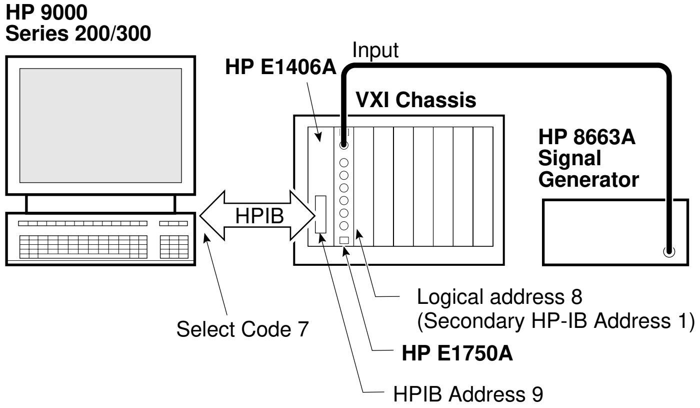


Figure 2-5. BASIC / UX Register Programming Setup

```
10    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
20    ! This HPBASIC/UX program will read the id, device, and status
30    ! registers of a VXI module and will print their values.
40    ! Asumptions:
50    !       HPIB address of Command module = 70900
60    !       Logical address of VXI module being read = 8
70    !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
80    !
90    ASSIGN @E1406 TO 70900
100   !
110   OUTPUT @E1406;"VXI:READ? 8,0"
120   ENTER @E1406;Id
130   OUTPUT @E1406;"VXI:READ? 8,2"
140   ENTER @E1406;Device
150   OUTPUT @E1406;"VXI:READ? 8,4"
160   ENTER @E1406;Status
170   !
180   PRINT "Id register     = ",Id
190   PRINT "Device register = ",Device
200   PRINT "Status register = ",Status
210   !
220   END
```

C Program Interrupt Example

For this program you must have an HP Vectra type PC using an HP-IB interface card (HP 82335A). You must have the device driver for the HP E1750A down loaded into the command module.

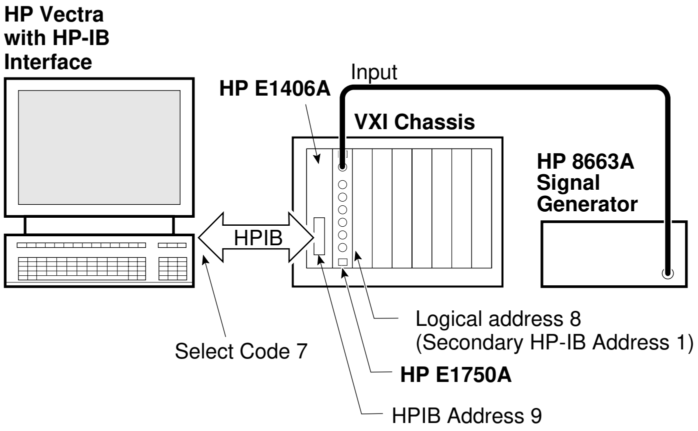

HP Vectra with HP-IB Interface
HP E1406A
Input
VXI Chassis
HP 8663A Signal Generator
HPIB
Select Code 7
Logical address 8 (Secondary HP-IB Address 1)
HP E1750A
HPIB Address 9

Figure 2-6. C Program Interrupt Setup

```
/***********************************************************************
* This C program will set up the command module to receive an interrupt
* from the distribution amp.  The program will poll the status
* until a SRQ is received.  After the SRQ is received, the program
* will find out what caused the interrupt (What failed.)  This program
* assumes you are using a PC with a hpib interface (HP23335) and a command
* module (E1406A).  You must have the device driver for the distribution
* amp down loaded into the command module.
* Assumptions:
*         HPIB interface select code = 7
*         E1406A hpib address = 9
*         VXI module logical address = 8  (secondary address 1)
*         interrupt level of distribution amp = 1
**********************************************************************/
#include <stdio.h>
#include <string.h>
#include <cfunc.h>
#include <chpib.h>

#define ISC 7
#define DIST_AMP 70901

/* This function will initialize the HP82335 hpib interface
*/
void Hpib_init (void)
{
          IORESET (ISC);
          IOTIMEOUT (ISC, 0.5);
          IOCLEAR (ISC);

          return;
}

/* This function takes care of any IO errors
*/
void Error(int error, char *text)
{
          if(error)
                    printf("Error %d in %s\n", error, text);

          return;
}

/* This function will send a character string to the command module
*  via hpib.
*/
void Hpib_write(char *input)
{
          float output;

          Error(IOOUTPUTS(DIST_AMP, input, strlen(input)), "IOOUTPUTS");
          return;
}
```

```
/* This function will send a character string to the command module
*  via hpib and will read the numeric response.
*/
int Hpib_read(char *input)
{
          float output;

          Error(IOOUTPUTS(DIST_AMP, input, strlen(input)), "IOOUTPUTS");
          Error(IOENTER(DIST_AMP, &output), "IOENTER");

          return (int)output;
}

void main(void)
{
          int in, out, response, loop;
          float float_temp;

          Hpib_init();

/* set up service requests */
          Hpib_write("*CLS");
          Hpib_write("STAT:QUES:ENAB 1536");
          Hpib_write("*SRE 8");

/* loop until there is a srq signal */
          printf("waiting for an interrupt\n");
          do {
             IOSTATUS(ISC, 1, &response);
          } while(!response);

/* find out what caused the interrupt */
          IOSPOLL(DIST_AMP,&response);
          if((response & 0x40) == 0x40)
          {
                    in = Hpib_read("DIAG:INP?");
                    out = Hpib_read("DIAG:OUT?");

                    if(in)
                       printf("FAIL: input at secondary address 1\n");
                    else
                    {
                       for(loop=0;loop<6;loop++)
                       {
                       if(out & (1 << loop))
                          printf("FAIL: output %d at secondary address 1\n",
                              loop+1);
                       }
                    }
          }

          return;
}
```

C Register Programming Example

For this example no drivers need to be down loaded.

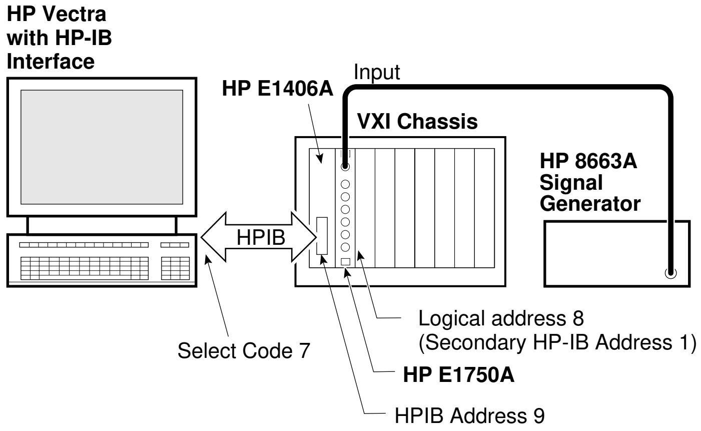


Figure 2-7. C Register Program Setup

```
/*****************************************************************
This program will read the id, device, and status registers of
the a VXI module and will print their values.  This program assumes
you are using a PC with a hpib interface (HP23335) and a command
module (E1406A).
Assumptions:
          HPIB interface select code = 7
          E1406A hpib address = 9
          VXI module logical address = 8
*****************************************************************/
#include <stdio.h>
#include <string.h>
#include <chpib.h>
#include <cfunc.h>

#define COMM_MODULE (70900)
#define ISC (7)

/* This function will initialize the HP82335 hpib interface
*/
void Hpib_init (void)
{
          IORESET (ISC);
          IOTIMEOUT (ISC, 0.5);
          IOCLEAR (ISC);

          return;
}

/* This function takes care of any IO errors
*/
```

```
void Error(int error, char *text)
{
          if(error)
                    printf("Error %d in %s\n", error, text);

          return;
}

/* This function will send a character string to the command module
*  via hpib and will read the numeric response.
*/
int Hpib_read(char *input)
{
          float output;

          Error(IOOUTPUTS(COMM_MODULE, input, strlen(input)), "IOOUTPUTS");
          Error(IOENTER(COMM_MODULE, &output), "IOENTER");

          return (int)output;
}

/* This is the main function that will print out the values of
*  the id, device, and status registers
*/
void main(void)
{
          unsigned int id, device,status;

          Hpib_init ();

          id     = Hpib_read("VXI:READ? 8,0");
          device = Hpib_read("VXI:READ? 8,2");
          status = Hpib_read("VXI:READ? 8,4");

          printf("id     = %u\n", id);
          printf("device = %u\n", device);
          printf("status = %u\n", status);

          return;
}
```

C-SCPI Programming Interrupt Example

To run this program your setup must be similar to that shown in Figure 2-8.

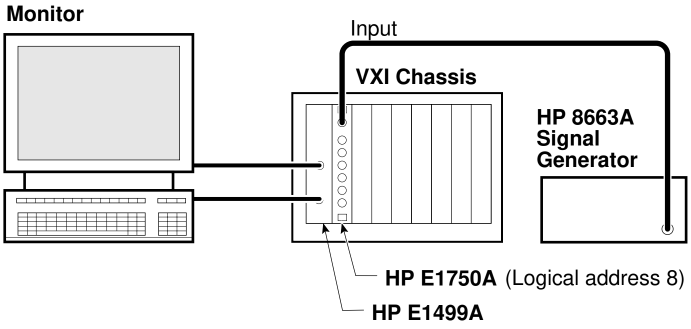


Figure 2-8. C-SCPI Interrupt Setup

```
/*
 * C-SCPI example - this program will set up the module for a SRQ.
 * When it receives and SRQ then it runs srq_routine which will
 * determine what caused the error
 *
 */
#include <stdio.h>
#include <cscpi.h>

INST_DECL( tst, "E1750A", REGISTER);
void srq_routine(INST_PARAM( id, "E1750A",REGISTER));
void tst_error(void);

main()
{
  INST_STARTUP();
  INST_OPEN( tst, "vxi,8");                     /* logical address 8 */
  INST_SEND( tst, "STAT:QUES:ENAB %d", 1536);   /* enable questionable reg */
  INST_SEND( tst, "*SRE %d", 8);
  INST_ONSRQ(tst, srq_routine);                 /* goto srq_routine on SRQ */
  cscpi_overlap(1);

  /* loop forever waiting for a SRQ*/
  while(1)
  {
  }

  return 0;
}

/* srq_routine - this function will run when there is a SRQ.  It will
 * read and report the status of the input and outputs to find out
 * what failed.
 */
void srq_routine(INST_PARAM( id, "E1750A",REGISTER))
{
  int data, output;

  INST_QUERY( tst, "STAT:QUES:EVEN?", "%d", &data); /*clear event reg */
  INST_QUERY( tst, "DIAG:INP?", "%d", &data);       /*find input status */
  if(data)
    printf("FAIL: input\n");
  else
  {
    INST_QUERY( tst, "DIAG:OUT?", "%d", &data);    /* find output status */

    for(output=1;output<7;output++)
    {
      if(data & (1 << (output - 1)))
        printf("FAIL: output %d\n", output);
    }
  }
  printf("\n");

  return;
}
```

3

SCPI Command Reference

About this Chapter

This chapter describes the Standard Commands for Programmable Instrumentation (SCPI) commands for the HP E1750A Broadband Distribution Amplifier.

Table 3-1. Common Command Reference

Command	Description
*CLS	Clear Status data structures (Event Registers and Error Queue)
*ESE <numeric value>	Specifies an 8 bit mask that determines which event bits from the Standard Event Register will be summarized by bit 5 (ESB) in the Standard Status Byte.
*ESE?	Returns the current Enable Standard Event mask byte.
*ESR?	Returns the current contents of the Standard Event Register which is cleared when read.
*IDN?	Returns identification string of the Distribution Amplifier module.
*OPC	Accepted but ignored.
*OPC?	Always returns a "1".
*RST	Resets the interrupt bit on board.
*SRE <numeric value>	Specifies an 8 bit mask that enables bits in the Status Byte Register that causes the generation of a Service Request. All bits enabled are summarized in bit 6 (RSQ) of the Status Byte Register.
*SRE?	Returns the current Service Request Enable Mask
*STB?	Returns the current contents of the Status Byte Register.
*TST?	Always returns a "0".
*WAI	Accepted but ignored.

Table 3-2. SCPI Command Reference

Subsystem	Command	Description
DIAGnostic	INPut?	Returns the status of the input signal detect circuit.
	OUTput<number>?	Returns the status of the output detect circuits 1 thru 6.
STATus:OPERation	CONDition?	Returns the contents of the Operation Condition register.
	ENABle<number>	Sets the contents of the Operation Event Enable register.
	ENABle?	Returns the contents of the Operation Event Enable register.
	[EVENt]?	Returns the contents of the Operation Event register.
STATus	PRESet	Clears the Operation and Questionable Enable registers.
STATus:QUEStionable	CONDition?	Returns the contents of the Questionable Condition register.
	ENABle<number>	Sets the enable mask for the Questionable Enable register.
	ENABle?	Returns the contents of the Questionable Enable register.
SYSTem	ERRor?	Returns the oldest uncleared error number and description from the HP IB error queue.
	VERSion?	Returns the SCPI version number for the instrument.

DIAGnostic:INPut?

INPut? query command, under the DIAGnostic system, returns the status of the input signal detect circuit.

Comments If a signal is detected, the HP E1750A will return a value of 0.
If no signal is detected, the HP E1750A will return a value of 1.

Example

```
DIAG:INP?       Queries input detect circuit

+0              HP E1750A signal detected response

                or

+1              HP E1750A signal not detected response
```

DIAGnostic:OUTput[<output_channel>]?

This command returns the status of the output signal detect circuit. <output_channel> is an optional parameter given in a range from 1 to 6.

Syntax

```
DIAGnostic:OUTput?
          :OUTput1?
          :OUTput2?
          :OUTput3?
          :OUTput4?
          :OUTput5?
          :OUTput6?
```

Comments

Specifying the output channel is optional. If the output channel is specified, the HP E1750A will return the value for the specified channel. If no output channel is specified then a composite number will be given for all the channels. It is a good practice to query the INPut first because loss of an input signal will cause all output detect circuits to detect loss of signal. (See examples).

If a signal is detected, the HP E1750A will return a value of 0.
If no signal is detected, the HP E1750A will return a value of 1.

With six outputs, each of the six outputs detect circuits sets a weighted bit described in the following table.

	Output 1	Output 2	Output 3	Output 4	Output 5	Output 6
Signal detected	0	0	0	0	0	0
No signal detected	1	2	4	8	16	32

Example 1 To repeat the following example simply remove the input source then send the following query. This will force all output detect circuits to report signal loss.

```
DIAG:INP?;OUT?   Queries input and all output registers

+1,+63           HP E1750A places a +1 in the input bit and adds
                 all output bits and responds with a sum. This
                 indicates no signal on the input and all outputs.
```

Example 2 To repeat the following example, replace the Input source and place a BNC short on Output 4 and Output 6. Send the following query.

```
DIAG:OUT?     Queries all Outputs

+40           HP E1750A adds bit weights 8+32 and responds with 40.
```

To decipher the HP E1750A, first subtract from the composite sum the highest bit weight possible. Then subtract the next highest and so on until you reach 0. Using the example above, you would subtract 32 (bit weight for Output 6) from 40. This leaves you with 8 (bit weight for Output 4). This means that Output 6 and Output 4 detect circuits fail to detect a "legal" signal. "Legal" signals are defined in Output Specifications in Chapter 5.

STATus:OPERation:CONDition?

The "STATus:OPERation:CONDition?" query returns the contents of the Operation Condition register.

The Operation Condition register is not used. It will always return a 0. It is included to conform with SCPI standards.

STATus:OPERation:ENABle <number>

The "STATus:OPERation:ENABle" command sets the contents of the Operation Event Enable register.

The Operation Enable register is not used. It is included to conform with SCPI standards.

Query Syntax

STATus:OPERation:ENABle?

The query of this command will return the value of the enable mask of the Operation Enable Register.

STATus:OPERation[:EVENt]?

The "STATus:OPERation[:EVENt]?" query returns the contents of the Operation Event register.

The Operation Event register is not used. It will always return a 0. It is included to conform with SCPI standards.

STATus:PRESet

The "STATus:PRESet" command clears the Operation and Questionable Enable registers.

STATus:QUEStionable:CONDition?

The "STATus:QUEStionable:CONDition?" query returns the contents of the Questionable Condition register.

The Questionable Condition register is constantly updated as questionable conditions change. No conditions are saved in this register.

When you read the contents of the Questionable Condition register, the value returned is the total bit weights of all the bits that are high at the time you read it. When you read the contents of the Questionable Condition register using this command, the contents of the register are not altered.

The decimal value of each bit (the bit weight) in the Questionable Condition register is shown in the following table.

Questionable Condition Register Bit Definitions

Bit	Weight	Condition
15 - 11		Not used - always 0.
10	1024	0 = All outputs present 1 = Not all outputs present.
9	512	0 = Input present. 1 = Input not present
8 - 0		Not used - always 0.

STATus:QUEStionable:ENABle <number>

The "STATus:QUEStionable:ENABle" command sets the enable mask for the Questionable Enable Register.

The <number> parameter is the number representing the value of bits in the Questionable Event Enable mask to be set. The number must be from 0 to 32767.

The Questionable Event Enable register contains a mask value for the bits to be enabled to set bit 3 in the status byte. A one (1) in the Questionable Event Enable register will enable the corresponding bit in the Questionable Event register to set bit 3 in the status byte. A zero (0) will disable the bit.

The decimal value of each bit (the bit weight) in the Questionable Event Enable register is shown in the following table.

Questionable Event Enable Register Bit Definitions

Bit	Weight	Condition
15		can't set.
14 - 11		X - don't care.
10	1024	0 will inhibit setting. 1 will enable setting.
9	512	0 will inhibit setting. 1 will enable setting.
8 - 0		X - don't care.

Query Syntax

STATus:QUEStionable:ENABle?

The query of this command will return the value of the enable mask of the Questionable Enable Register.

SYSTem:ERRor?

The "SYSTem:ERRor?" query returns the oldest uncleared error number and error description from the HP-IB error queue.

When an error is read, it is cleared as long as the error condition no longer exists. When the "SYSTem:ERRor?" query is sent, only the oldest unread error in the HP-IB error queue will be returned.

NOTE

The table of error messages in Chapter 4, is organized in ascending detailed error number order. Use the detailed error number enclosed in parentheses when looking up the error condition.

The HP-IB error queue returns the oldest error message when queried. Preset has no effect on the HP-IB error queue; it is only cleared at power up, by sending the "*CLS" command, or by reading its entire contents.

SYSTem:VERSion?

The "SYSTem:VERSion?" query returns the SCPI (Standard Commands for Programmable Instruments) version number that the HP E1750A supports.

4

Error Messages

Error Messages

The error messages listed in table 4-1 are those that are most likely to be generated. All error messages follow the SCPI Command Reference definition.

NOTE

If you use the query: SYST:ERR?
the HP 1750A will return a string describing the error message.

Table 4-1. Error Messages

Code	Message	Cause
-101	Invalid character	Unrecognized character in specified parameter.
-102	Syntax error	Command is missing a spece or comma between parameters.
-103	Invalid separator	Command parameter is separated by some character other than a comma.
-104	Data type error	The wrong data type (i.e. number, character, string expression) was used when specifying a parameter.
-108	Parameter not allowed	Parameter specified in a command in which a parameter is required.
-109	Missing parameter	No parameter specified in the command in which a parameter is required.
-113	Undefined header	Command header was incorrectly specified.
-123	Numeric overflow	A parameter specifies a value greater than the command allows.
-131	Invalid suffix	Parameter suffix incorrectly specified.
-138	Suffix not allowed	Parameter suffix is specified when one is not allowed.
-141	Invalid character data	The discrete parameter specified is not allowed
-178	Expression data not allowed	A parameter other than the channel list is enclosed in a parentheses.

Error Messages

Table 4-1. Error Messages (Continued)

Code	Message	Cause
-222	Data out of range	The parameter value specified is too large or too small.
-224	Illegal parameter value	The numeric value specified is not allowed.
-240	Hardware error	Hardware error detected during power-on cycle. Return to Hewlett-Packard for repair.
-350	Too many errors	The error queue is full as more than 30 error have occurred.
-410	Query interrupted	Data is not read from the output buffer before another command is executed.
-420	Query unterminated	Command which generates data not able to finish executing due to a configuration error.
-430	Query deadlocked	Command execution cannot continue since the mainframe's command input and data output buffers are full. Clearing the instrument restores control.

Error Messages

Printed: July 1993

Printed in USA

Manual part number
E1750-90001

Certification and Warranty

Certification

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

Warranty

This Hewlett-Packard instrument product is warranted against defects in material and workmanship for a period of three years from date of shipment. During the warranty period, Hewlett-Packard Company will, at its option, either repair or replace products which prove to be defective.

For detailed warranty information, see back matter.

Safety Considerations

General

This product and related documentation must be reviewed for familiarization with this safety markings and instructions before operations.

Warning Symbols That May Be Used In This Book

⚠

Instruction manual symbol; the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual.

⚡

Indicates hazardous voltages.

⏚

Indicates earth (ground) terminal.

⊥

or

⊥

Indicated terminal is connected to chassis when such connection is not apparent.

∿

Indicates Alternating current.

⎓

Indicates Direct current

Safety Considerations (Cont'd.)

WARNING

BODILY INJURY OR DEATH MAY RESULT FROM FAILURE TO HEED A WARNING. DO NOT PROCEED BEYOND A WARNING UNTIL THE INDICATED CONDITIONS ARE FULLY UNDERSTOOD AND MET.

CAUTION

Damage to equipment, or incorrect measurement data, may result from failure to heed a caution. Do not proceed beyond a CAUTION until the indicated conditions are fully understood and met.

Safety Earth Ground

An uninterruptible safety earth ground must be maintained from the mains power source to the product's ground circuitry.

WARNING

ANY INTERRUPTION OF THE PROTECTIVE GROUNDING CONDUCTOR (INSIDE OR OUTSIDE THE PRODUCT'S CIRCUITRY) OR DISCONNECTING THE PROTECTIVE EARTH TERMINAL WILL CAUSE A POTENTIAL SHOCK HAZARD THAT COULD RESULT IN PERSONAL INJURY. (GROUNDING ONE CONDUCTOR OF A TWO CONDUCTOR OUTLET IS NOT SUFFICIENT PROTECTION.) WHENEVER IT IS LIKELY THAT THE PROTECTION HAS BEEN IMPAIRED, THE PRODUCT MUST BE MADE INOPERATIVE AND BE SECURED AGAINST ANY UNINTENDED OPERATION.

For additional safety and acoustic noise information, see back matter.

Hewlett-Packard Company
Santa Clara Division
5301 Stevens Creek Boulevard
Santa Clara, California 95052-8059

Continued from front matter . . .

Warranty (Cont'd.)

For warranty service or repair, this product must be returned to a service facility designed by HP. Buyer shall prepay shipping charges to HP and HP shall pay shipping charges to return the product to Buyer. However, Buyer shall pay all shipping charges, duties, and taxes for products returned to HP from another country.

HP warrants that its software and firmware designed by HP for use with an instrument will execute its programming instructions when properly installed on that instrument. HP does not warrant that the operation of the instrument, or software, or firmware will be uninterrupted or error free.

Limitation of Warranty

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

NO OTHER WARRANTY IS EXPRESSED OR IMPLIED. HP SPECIFICALLY DISCLAIMS THAT IMPLIED WARRANTIES OR MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Exclusive Remedies

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLD AND EXCLUSIVE REMEDIES. HP SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, WHETHER BASED ON CONTRACT, TORT, OR ANY OTHER LEGAL THEORY.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Hewlett-Packard products.

For any assistance, contact your nearest Hewlett-Packard Sales and Service Office.

Safety Considerations (Cont'd.)

WARNING

INSTRUCTIONS FOR ADJUSTMENTS WHILE COVERS ARE REMOVED AND FOR SERVICING ARE FOR USE BY SERVICE-TRAINED PERSONNEL ONLY. TO AVOID DANGEROUS ELECTRIC SHOCK, DO NOT PERFORM SUCH ADJUSTMENTS OR SERVICING UNLESS QUALIFIED TO DO SO.

Acoustic Noise Emissions

LpA<47 dB at operator position, at normal operation, tested per EN 27779. All data are the results from type test.

GERAeUSCHEMISSION

LpA<47 dB am Arbeits platz, normaler Betrieb, geprueft nach EN 27779. DieAnbagen beruhen auf Ergebnissen von Typpruefungen.

Index