Errata

Title & Document Type: 5373A Programming Manual - January 1991

Manual Part Number: 05373-90003

Revision Date: January 1991

HP References in this Manual

This manual may contain references to HP or Hewlett-Packard. Please note that Hewlett-Packard's former test and measurement, semiconductor products and chemical analysis businesses are now part of Agilent Technologies. We have made no changes to this manual copy. The HP XXXX referred to in this document is now the Agilent XXXX. For example, model number HP8648A is now model number Agilent 8648A.

About this Manual

We've added this manual to the Agilent website in an effort to help you support your product. This manual provides the best information we could find. It may be incomplete or contain dated information, and the scan quality may not be ideal. If we find a better copy in the future, we will add it to the Agilent website.

Support for Your Product

Agilent no longer sells or supports this product. You will find any other available product information on the Agilent Test & Measurement website:

www.tm.agilent.com

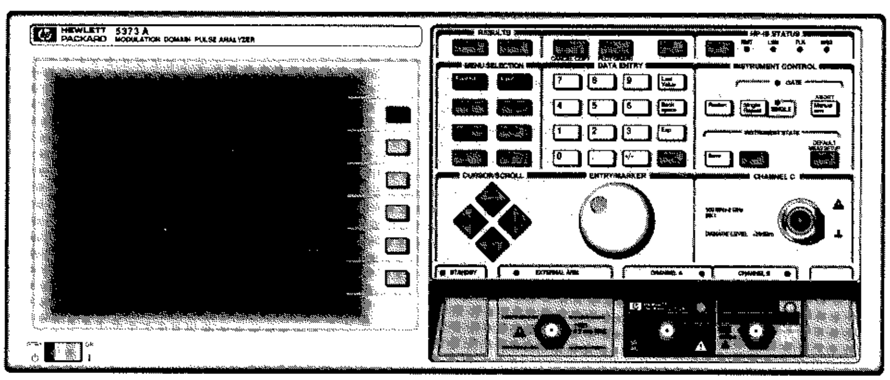
Search for the model number of this product, and the resulting product page will guide you to any available information. Our service centers may be able to perform calibration if no repair parts are needed, but no other support from Agilent is available.

Agilent Technologies

HP 5373A

Modulation Domain Pulse Analyzer

PROGRAMMING MANUAL



HEWLETT PACKARD

Certification and Warranty

CERTIFICATION

Hewlett-Packard Company certifies that this product met its published specifications 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 that organization'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 one year 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 warranty service or repair, this product must be returned to a service facility designated 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 designated 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 of the environmental specifications for the product, or improper site preparation or maintenance.

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

EXCLUSIVE REMEDIES

THE REMEDIES PROVIDED HEREIN ARE BUYER'S SOLE 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. Addresses are provided at the back of this manual.

PROGRAMMING MANUAL

HP 5373A
Modulation Domain Pulse Analyzer

MANUAL APPLICABILITY

This manual applies directly to an HP 5373A having the serial number prefix listed below. If this number does not match your instrument, refer to the "Manual Updating Changes" included with this manual.

For additional important information about serial numbers, see INSTRUMENTS COVERED BY THIS MANUAL in the introduction of the HP 5373A Operating Manual.

SERIAL NUMBER
Serial Number Prefix: 3102

Edition 1
E0191

Printed: JANUARY 1991

HEWLETT PACKARD

MANUAL PART NUMBER 05373-90003
Microfiche Part Number 05373-90004

Safety Considerations

GENERAL

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

This product is a Safety Class I instrument (provided with a protective earth terminal).

BEFORE APPLYING POWER

Verify that the product is set to match the available line voltage and the correct fuse is installed. Refer to instructions in Appendix B of the Operating Manual.

SAFETY EARTH GROUND

An uninterruptible safety earth ground must be provided from the mains power source to the product input wiring terminals or supplied power cable.

Safety Symbols

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 — Indicates terminal is connected to chassis when such connection is not apparent.

Alternating current.

Direct current.

WARNING

THIS DENOTES A HAZARD. IT CALLS ATTENTION TO A PROCEDURE, PRACTICE, OR THE LIKE, WHICH, IF NOT CORRECTLY PERFORMED OR ADHERED TO, COULD RESULT IN PERSONAL INJURY. DO NOT PROCEED BEYOND A *WARNING* SIGN UNTIL THE INDICATED CONDITIONS ARE FULLY UNDERSTOOD AND MET.

CAUTION

This denotes a hazard. It calls attention to an operating procedure, practice, or the like, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product. Do not proceed beyond a *CAUTION* sign until the indicated conditions are fully understood and met.

Safety Information

WARNING

Any interruption of the protective grounding conductor (inside or outside the instrument) 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 instrument must be made inoperative and be secured against any unintended operation.

If this instrument is to be energized via an autotransformer (for voltage reduction) make sure the common terminal is connected to the earthed pole terminal (neutral) of the power source.

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.

For continued protection against fire, replace the line fuse(s) only with 250V fuse(s) of the same current rating and type (for example, normal blow, time delay). Do not use repaired fuses or short circuited fuseholders.

When measuring power line signals, be extremely careful and always use a step-down isolation transformer whose output voltage is compatible with the input measurement capabilities of this product. This product's front and rear panels are typically at earth ground, so **NEVER TRY TO MEASURE AC POWER LINE SIGNALS WITHOUT AN ISOLATION TRANSFORMER.**

Table of Contents

Chapter 2
HP-IB GENERAL INFormatION
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 3
BINARY OUTPUT
(Continued)

Chapter 4 PROGRAMMING RULES AND GUIDANCE

Chapter 5 COMMAND REFERENCE

Chapter 5
COMMAND REFERENCE
(Continued)

Chapter 5
COMMAND REFERENCE
(Continued)

Chapter 5
COMMAND REFERENCE
(Continued)

Chapter 5
COMMAND REFERENCE
(Continued)

Chapter 5
COMMAND REFERENCE
(Continued)

Appendix A
HP-IB INTERCONNECTION

Appendix B DEFAULT MEASUREMENT SETUPS

Appendix C STATUS AND ERROR MESSAGES

Appendix C
STATUS AND ERROR MESSAGES
(Continued)

Appendix D PROGRAMMING THE ARMING MODES

Appendix E PROGRAM EXAMPLES

Appendix F PROGRAM CONVERSION

1

BEFORE YOU START TO PROGRAM

INTRODUCTION

The HP 5373A Modulation Domain Pulse Analyzer operates as a stand-alone test instrument or can work as part of an HP-IB instrumentation network. When connected to an HP-IB network, you can control the instrument using a program running on a compatible controller such as the Hewlett-Packard 9000 series 200/300 controller. This manual describes how to write programs to direct the operation of the instrument.

Before attempting to program the HP 5373A, take some time to become familiar with the content and organization of this manual. Topics in this chapter include:

- Who should read this manual
- How to use this manual
- Related documentation
- HP-IB overview
- Writing programs

WHO SHOULD READ THIS MANUAL

This manual is written for a diverse audience.

- First-time users
- Experienced HP-IB programmers.

First-Time Users

To quickly grasp and use the concepts and techniques described in this manual you should have some programming experience using a high-level language such as Pascal, BASIC, C or FORTRAN. Knowing the BASIC language is helpful, but not absolutely required. (For the sake of simplicity and continuity all the programming examples in this manual are written in HP BASIC.) If you know another version of BASIC such as GW BASIC, you should be able to transfer that knowledge to the requirements of programming the HP 5373A.

There is nothing to preclude the use of languages other than BASIC; however, you must infer the applicable commands and techniques (for such other languages) by first understanding the BASIC commands and techniques described here.

Experienced Programmers

If you are experienced in programming other HP-IB devices, you will know many concepts and procedures discussed in this manual. In this event, you may find Chapter 5, Command Reference, most helpful.

HOW TO USE THIS MANUAL

Because the manual is written for a diverse audience, different users are expected to approach the material from their unique perspectives and informational needs. Here is a chapter-by-chapter abstract to help you decide where to go for information.

- *Chapter 1, Before You Start to Program,* provides an overview of topics covered in the manual and describes their relationships. This is suggested reading for all users.
- *Chapter 2* describes the HP-IB in a general way and explains HP 5373A interface capabilities. Remote operating characteristics are elaborated. The status reporting scheme including hardware status and event status is described. Also covered are ASCII and Floating Point output formats.
- *Chapter 3* describes how to get measurement results from the instrument using the Binary data output.
- *Chapter 4* relates programming rules and gives you guidance needed to program the instrument. Topics such as command structure, command mnemonics, syntax, formatting rules, and command processing order are described. Here you will also find tips to help speed up the programming process. A multi-page reference illustrates how programming commands relate to instrument menus and corresponding front panel controls.
- *Chapter 5* is a complete reference to all the HP-IB commands. It describes in detail the proper syntax, path, and parameters of the commands as well as providing

HP BASIC programming examples. This chapter will probably be used continually as you write your programs.

- *Appendix A HP-IB Connection* illustrates and describes how to connect the HP 5373A to an HP-IB network (including an optional printer or plotter.)
- *Appendix B Default Measurement Setups* describes how the Default Measurement Setup function works. Setup values for all the measurement types are listed.
- *Appendix C Status and Error Messages* lists and describes the various error types and status messages.
- *Appendix D Programming the Arming Modes* gives arming modes that are available for respective instrument functions.
- *Appendix E Programming Examples* gives commented HP BASIC program listings. The programs are chosen and constructed to highlight useful programming techniques. You may be able to employ these programs directly or modify them to suit your individual situation and preferences.

RELATED DOCUMENTATION

Other manuals, guides, and tutorials you can refer to for information are:

- The HP 5373A Modulation Domain Pulse Analyzer Operating Manual (HP part number 05373-90001)
- The HP BASIC programming reference set (If you are using HP BASIC)
- HP-IB Tutorial Description of the Hewlett-Packard Interface Bus (HP part number 5952-0156)

The HP 5373A Operating Manual describes all the operational controls and the menu structure. Many tutorial examples lead you through specific measurements.

The HP BASIC programming reference is a multi-volume set of manuals that completely describes all HP BASIC commands. In addition, the work treats interface issues, and programming techniques at length.

Turn to the HP-IB tutorial description for the following topics:

- Introduction to HP-IB
- IEEE 488.1 bus
- IEEE 488.2 standard
- Data coding and formats
- Syntax
- Status reporting
- Common commands
- System initialization

HP-IB OVERVIEW

The HP-IB is a general purpose digital interface which simplifies the design and integration of instruments and computers into systems. The bus minimizes electrical/mechanical, hardware, and functional compatibility problems between devices, yet has sufficient flexibility to accommodate a wide and growing range of products.

HP-IB is Hewlett-Packard's implementation of IEEE-488. This standard permits the linking of diverse products into measurement systems. Accordingly, you may incorporate many test instruments (up to 15 devices on a single bus) in ways to suit individual needs and preferences.

WRITING PROGRAMS

Before attempting to program the HP 5373A you should first know how to use all the HP 5373A front panel controls and thoroughly understand the measurements you wish to perform. Knowing the front panel controls helps because the programming command scheme follows the general menu layout and thus mimics, to some extent, manual operation of the instrument. Measurement techniques and operational tips are described in the HP 5373A Operating Manual.

A Teach/Learn feature is described that enables you to store operating setups for later use. This programming tip and others are given in Chapter 4.

2 HP-IB GENERAL INFORMATION

2

HP-IB GENERAL INFORMATION

INTRODUCTION

This chapter contains programming information for remotely operating the HP 5373A through the Hewlett-Packard Interface Bus (HP-IB). The information in this chapter includes descriptions of:

- The HP-IB.
- The HP 5373A interface capabilities.
- Remote operation characteristics.
- Status reporting structure.
- Data output formats.

NOTE

All the HP 5373A HP-IB programming commands are described in Chapter 5, Command Reference. Command entries include one or more program lines. Appendix E has expanded program examples that demonstrate in greater detail some of the programming features of the HP 5373A.

The HP 5373A operates remotely via the HP-IB. All menu functions and front panel controls (excluding the power switch) are programmable through the HP-IB. At the simplest level, the HP 5373A can transmit data in the Talk Only mode to output devices such as a printer or plotter. In more sophisticated systems, a controller can direct the HP 5373A to perform a specific measurement, process the resulting measurement data, and transfer the results.

Most HP 5373A remote programming operations are extensions of the front panel functions. Other operations, such as Teach/Learn and instrument identification, are only available in remote operation.

You can use any computer that is compatible with the HP-IB standard. All program examples given in this manual are written assuming an HP 9000 Series computer as the controller. Refer to Appendix A for information about connecting a controller to the HP 5373A.

HP-IB DESCRIPTION

The HP-IB is a parallel interface bus. All devices on the bus can be addressed at one time, however, only one device may respond at a time. The controller commands a specific device to respond, and maintains the data flow and interface functions. Data is transferred at the speed of the slowest device.

Party Line Structure Explained

The HP-IB uses a party-line structure (devices share signal lines). A maximum of 15 devices may be connected in an HP-IB system, in virtually any configuration desired. There must be an uninterrupted path to every device operating on the bus. Sixteen lines interconnect devices in parallel. Eight lines transfer data and commands, three accomplish handshaking, and five lines manage the bus.

Interface System Terms

The following terms and concepts apply to HP-IB operations.

Address: Each device on the interface is assigned an address. The address specifies which device receives or sends information.

Byte: A byte is a unit of information consisting of eight binary digits (bits).

Device: Any instrument or unit that is HP-IB compatible is defined as a device.

Device-Independent Command: A command predefined by the interface standard to have a specified bit pattern and resulting action.

Device-Dependent Command: A command that is specific to a particular instrument or family of instruments, which is not predefined by the interface standard. Device-dependent commands are usually sent as ASCII strings of characters.

Polling: Polling is a process typically used by a controller to locate a device that requires service from the controller. There are two types of polling, Serial Poll and Parallel Poll:

Serial Poll: When the controller executes a serial poll, the addressed device sends one byte of operational information called a status byte. If more than one device on the interface is capable of requesting service, each device on the interface must be polled individually until the device that requested service is located.

Parallel Poll: Parallel poll is a controller-initiated operation to obtain information from several devices simultaneously. The HP 5373A does not have parallel poll capability.

Major Interface Functions

Each device on the interface bus may have one or more of the following major device capabilities: **Controller, Talker,** or **Listener.**

The Controller manages interface activity, and must have the applicable interface module. Controllers transmit all device-independent commands to other devices in the interface and usually have Talker and Listener capabilities. Only one device on the interface may be the active controller at any one time. The HP 5373A has no controller capabilities.

Talkers are devices that can send data or device-dependent commands through the interface. Note that a talker will not actually send data or information until the appropriate command is sent by the controller. The HP 5373A has Talker capabilities. When the instrument is talking on the interface, or is addressed to talk, the TLK LED on the front panel lights.

In special situations, a device may be classified as a Talk Only device. A Talk Only device sends information to Listen Always devices or any devices addressed to listen. Such a system has no controller. For example, the HP 5373A can be configured to Talk Only and send measurement results to a printer (configured as Listen Always) by setting the instrument to the Talk Only mode.

Listeners can receive information over the interface. When the HP 5373A is listening, or addressed to listen, the LSN LED on the front panel lights. Listeners must also be enabled by the Controller to receive data or information.

HP 5373A INTERFACE CAPABILITIES

The capabilities of a device connected to the bus are specified by the list of interface functions supported. These functions provide a way for a device to receive, process, and send messages over the bus.

Table 2-1 lists the HP-IB interface functions defined by the IEEE 488-1978 standard, including the name, mnemonic, and a brief description. Also listed in the table are the subset identifiers (the interface function mnemonic followed by a number). These identifiers indicate the specific HP-IB interface function capabilities of the HP 5373A.

Table 2-1. HP 5373A HP-IB Interface Function Capabilities

Name and Mnemonic	General Description	Subset Identifier	Specific HP 5373A Capability
Source Handshake (SH)	Capability to properly translate a multiline message.	SH1	The HP 5373A can generate messages.
Acceptor Handshake (AH)	Capability to guarantee proper reception of remote multiline messages.	AH1	The HP 5373A can interpret received messages.
Talker (T)	Capability to transmit data over the bus when addressed.	T5	The HP 5373A can function as a talker. In addition, it can operate as a Talk Only instrument and will respond to serial poll. It will unlisten if addressed as a talker.
Extended Talker (TE)	Talker capability with address extension.	TE0	The HP 5373A cannot function as an extended talker.
Listener (L)	Capability to receive data over the bus when addressed.	L4	The HP 5373A can function as a listener. In addition, it will untalk if addressed as a listener.
Extended Listener (LE)	Listener capability with address extension.	LE0	The HP 5373A cannot function as an extended listener.
Service Request (SR)	Capability permitting a device to asynchronously request service from the controller.	SR1	The HP 5373A can generate a service request.
Remote/Local (RL)	Capability to select between two sources of input information: local (front panel controls) and remote (input information from the bus).	RL1	The HP 5373A can operate both in remote and local modes. In addition, it can respond to local lockout.
Parallel Poll (PP)	Provides capability for a device to uniquely identify itself if it requires service and the controller is requesting a response. This capability differs from service request in that it requires a commitment of the controller to periodically conduct a parallel poll.	PP0	The 5373A does not support parallel poll.

Table 2-1. HP 5373A HP-IB Interface Function Capabilities (Continued)

Name and Mnemonic	General Description	Subset Identifier	Specific HP 5373A Capability
Device Clear (DC)	This function allows a device to be initialized to a predefined state.	DC1	The 5373A supports both the Device Clear (DCL) and Selected Device Clear (SDC) commands.
Device Trigger (DT)	This function permits a device to have its basic operation initiated by the talker on the bus.	DT1	The 5373A can be remotely triggered.
Controller (C)	This function permits a device to send addresses, universal commands, and addressed commands to other devices on the HP-IB. It may also include the ability to conduct polling to determine devices requiring service.	C0	The 5373A cannot function as a controller.
Drivers (E)	This code describes the type of electrical drivers used in a device.	E2	The 5373A has three-state drivers.

Via the HP-IB, you can remotely program most controls (except Power On/Off). You also receive measurement data via the HP-IB. The HP 5373A operates as both a talker and a listener, as described in *Table 2-1*. The output format is the same regardless of the mode (Talk Only/Addressable).

TALK: The HP 5373A can be addressed to Talk by a controller or by setting the instrument to the Talk Only mode. When addressed as a Talker, the instrument sends data to other devices on the bus. This data may result from a measurement, an error condition, a diagnostic test, or other operations.

LISTEN: When addressed as a Listener, the instrument accepts any number of commands from a controller on the bus. These commands are used to program the instrument operation.

SERVICE REQUEST: A Service Request (SRQ) is generated on the interface when an enabled status bit is set. The HP 5373A can asynchronously request service from the controller that is in charge of the bus.

REMOTE/ LOCAL:	At power-up, the instrument is under front panel (local) control. When the HP 5373A receives a programming command, it switches to Remote. Once in Remote, programmable functions cannot be affected by the front panel controls. The LOCAL key may be used to manually return to local control only if the Local Lockout (LLO) is off. If Lockout is on, the LOCAL key is ignored, and the bus command LOCAL must be sent to disable LLO and return to local control.
PARALLEL POLL:	The HP 5373A does not respond to parallel poll.
DEVICE CLEAR:	When a universal or selected Device Clear is received, the HP 5373A clears any errors present, clears all input and output buffers, and resets the hardware for a new measurement.
DEVICE TRIGGER:	When a Device Trigger is received, the instrument will start a new measurement, if the sample rate is set to SINGLE. If the sample rate is set to REPETITIVE, the Device Trigger command will start a new measurement.
CONTROLLER:	The HP 5373A cannot be used as a controller.

Front Panel Interface Status LEDs

The remote status of the HP 5373A is indicated by four Interface Status LED annunciators (RMT, LSN, TLK, SRQ) in the upper right corner of the front panel, as follows:

RMT:	Lights to indicate the instrument is under remote control.
LSN:	Lights to indicate the instrument is addressed to listen (receive commands) or is an active listener.

TLK: Lights to indicate the instrument is addressed to talk (send data) or is an active talker.

SRQ: Lights to indicate the instrument is requesting service from the interface controller.

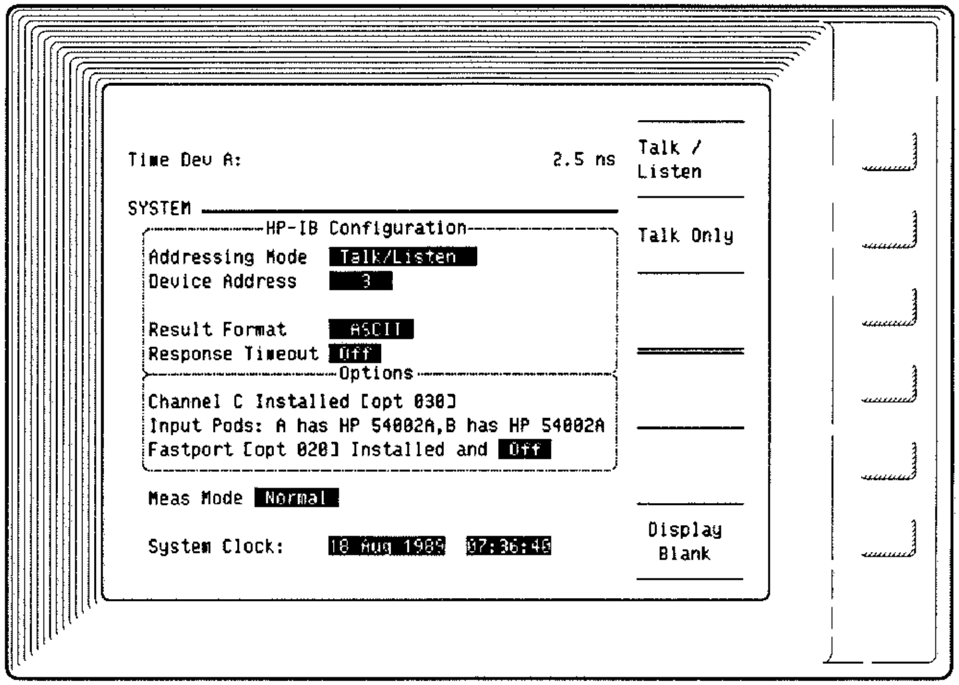
HP-IB ADDRESS SELECTION

To use the HP 5373A in an HP-IB system, you must set the desired HP-IB address. The address is used by the controller to identify the HP 5373A. The range of permitted addresses is "0" to "30", inclusive. The address is saved when the HP 5373A is off or unplugged. If the saved address cannot be recalled due to memory or battery failure, a default address of "3" is used.

Set the HP 5373A HP-IB address using front panel controls (*Figure 2-1*) Here is the procedure.

1. Press front panel SYSTEM key (not shown in figure).The System menu screen appears.
2. Check that Talk/Listen appears in Adressing Mode field. If not, press Talk/Listen softkey so that Talk/Listen is displayed.
3. Press down-arrow key (not illustrated) to move the menu cursor to Device Address field.
4. Use Increment Value and Decrement Value softkeys to select the address.

Figure 2-1.
System Menu —
Setting HP-IB
Address



When a controller is installed, the HP 5373A functions as a talker and a listener. In the absence of a controller, you can set the HP 5373A to Talk Only mode. In this way you can send measurement results to another device such as a printer or plotter. In Talk Only mode, the HP 5373A functions only in an output condition. The receiving device must be set to Listen Always.

INTERFACE COMMANDS

The HP 5373A recognizes two classes of commands: device-independent commands and device-dependent commands. Device-independent commands are defined by the interface standard document and are the same for all instruments. Device-dependent commands are unique to the instrument and are defined by the instrument designer.

The device-independent commands are described in the following paragraphs. Device-dependent commands are described in the Chapter 5, Command Reference.

Device-Independent Commands

Device-independent commands are identified by three-letter mnemonics such as GTL, which represents Go To Local. Device-independent commands are sent as encoded bytes on the interface bus and not as ASCII strings, thus these commands cannot be sent from the controller using the OUTPUT statement. Many controllers, however, do incorporate a command of the form SEND#;CMD#, where # is a decimal number.

Table 2-2 lists the mnemonic, full name, and a brief description of the instrument response for each supported device-independent command.

Table 2-2. Device-Independent Commands

Mnemonic	Command Name	Description
ATN	Attention	Alerts the instrument of each device independent message being sent, so the instrument is ready to accept data and interpret them as commands.
DCL	Device Clear	This command clears all errors, aborts all partially completed commands and pending send data commands, and clears all input and output buffers.
EOI	End Or Identify	If ATN is false and the instrument is a listener, EOI acts as a message delimiter, and indicates the last data byte of a multibyte sequence.
GET	Group Execute Trigger	If the instrument is addressed to listen, GET aborts the current measurement, and triggers the next measurement immediately.
GTL	Go To Local	If the instrument is addressed to listen, GTL returns the instrument to front panel (local) operation. Local Lockout is not cleared.
IFC	Interface Clear	The instrument untalks and unlistens, and the interface initializes to an idle state (no activity on the bus).
LAD#	Listen Address n	If n matches the instrument address, the instrument becomes a listener.
LLO	Local Lockout	The front panel LOCAL key is disabled if the instrument is in remote mode. All front panel key presses are ignored.
MLA	My Listen Address	MLA is the listen address (LAD#) that matches the instrument address.
MTA	My Talk Address	MTA is the talker address (TAD#) that matches the instrument address.

Table 2-2. Device-Independent Commands (Continued)

Mnemonic	Command Name	Description
NRE	Not Remote Enable	The instrument returns to front panel (local) operation; Local Lockout is cleared.
NUL	Null	No effect when received by the instrument.
REN	Remote Enable	The instrument enters the remote state, and is enabled to respond to interface commands when addressed as a listener.
SDC	Selected Device Clear	If the instrument is a listener, will cause the same response as DCL.
SPD	Serial Poll Disable	Terminates serial polling, and returns the instrument to a normal talker state to output device dependent data rather than status information.
SPE	Serial Poll Enable	Establishes serial polling, and enables the instrument to send the serial poll status byte when addressed to talk.
TAD#	Talk Address n	If n matches the instrument address, the instrument becomes a talker.
UNL	Unlisten	The instrument is unaddressed and terminates listening. A single device cannot be unaddressed without unaddressing all listeners.
UNT	Untalk	Unaddresses the instrument, if currently a talker, and terminates talking. Addressing another talker on the interface automatically unaddresses any current talker.

Meta Messages

To simplify using the HP-IB interface, Hewlett-Packard developed the Meta Message concept, shown in *Table* 2-3. Rather than require you to remember all the device-independent messages and their interactions, useful command sequences are integrated into a corresponding single command, on many Hewlett-Packard controllers. For example, to clear the instrument at address three using the device-independent commands, you must send the sequence ATN, UNL, MTA, LAD 3, SDC. The HP BASIC command "CLEAR 3" sends the same sequence with no further user interaction required, thus greatly simplifying the programming interface.

Many messages implemented in HP BASIC may be sent in either of two forms, with addressing and without addressing. The form with addressing normally addresses a particular device to listen. For example, the command "REMOTE 7" sends REN without making any device a listener, while the command "REMOTE 703" sends REN and makes the device at address 3 a listener.

NOTE

Care must be taken when using Meta messages. If a Meta message is issued shortly after sending another command, there is a possiblity that the Meta message will prevent the completion of the previously issued command. If problems are encountered, WAIT commands should be used to allow the command sufficient time to complete execution.

Table 2-3. Meta Messages

Meta Message	Command Sequence	General Description	Specific HP 5373A Response
DATA	UNL, MTA, LAD#, data	Transfers device dependent information from one device to one or more devices on the bus.	The 5373A sends measurement data as defined by the device-dependent command received from the controller.
TRIGGER	UNL, MTA, LAD#, GET	Causes a group of selected devices to simultaneously initiate a set of device-dependent actions.	Starts a new measurement.
CLEAR	UNL, MTA, LAD#, SDC	Causes the instrument to be set to a predefined state, such as a certain range or function.	Causes the 5373A to clear any errors present, clears all input and output buffers, and resets the hardware for a new measurement.
REMOTE	REN, UNL, MTA, LAD#	Permits selected devices to be set to remote operation, allowing parameters and device characteristics to be controlled by bus messages.	Causes the 5373A to go to remote operation if REN is true, and if instrument is addressed to listen. Locks out all front panel keys except LOCAL; instrument is controlled by bus messages. Until changed via the bus, remote operation is according to state of front panel settings just prior to going to remote.
LOCAL	UNL, MTA, LAD#, GTL	Causes selected devices to return to local (front panel) operation.	Returns the 5373A to front panel control. Instrument status is that set just prior to receipt of the Local message.

Table 2-3. Meta Messages (Continued)

Meta Message	Command Sequence	General Description	Specific HP 5373A Response
LOCAL LOCKOUT	LLO	Disables local (front panel) controls of selected devices.	Disables LOCAL key. The 5373A remains in remote operation until a Local message is received on the bus.
LOCAL/ CLEAR LOCAL LOCKOUT	LCLL	Returns all devices to local (front panel) control and simultaneously clears the Local Lockout message.	Returns 5373A to local control and clears Local Lockout message.
SERVICE REQUEST	SRQ	Indicates a device requires service by the controller. Service requests occur when specific hardware-related conditions occur such as a time base oscillator out of lock, an input pod removed, and so on.	The HP 5373A sends a Service Request message to the controller under certain conditions, as defined by the settings of the Event Status Enable and Hardware Status Enable registers. This message is ignored by the HP 5373A when received.
STATUS BYTE	UNL, MLA, TAD#, SPE, data, SPD, UNT	Presents status information of a particular device; one bit indicates whether or not the device currently requires service, the other seven bits (optional) are used to indicate the type of service required.	The HP 5373A sends status information to the controller. The assignment of the bits in the Status Byte are shown in *Table 2-6.*
STATUS BIT	Not applicable	A single bit of device-dependent status information which may be logically combined with status bit information from other devices on the controller.	The HP 5373A does not use this message.
PASS CONTROL	Not applicable	Passes bus controller responsibilities from the current controller to a device which can assume the bus supervisory role.	The HP 5373A does not use this message.
ABORT	IFC	Unconditionally terminates bus communications and returns control to the system controller.	All HP-IB activity terminated and control returns to the system controller. Talk and Listen are cleared for the HP 5373A and all other devices on the bus, which terminates all bus communications. The HP 5373A status remains as it was just prior to receipt of the Abort message. Any partially entered HP-IB data message is aborted. The ABORT Meta message is distinct from the HP5373A ABORT Command.

Through meta messages, bus devices can exchange control and measurement information. The command sequences are typical in that different controllers may send different sequences for a given meta message, but will produce the same result.

REMOTE OPERATION

When the HP 5373A operates in Remote mode, all front panel controls are disabled except the power switch and the LOCAL key (if Local Lockout is not active). Conditions for changing from Remote to Local, and the operational status of each mode are described in following paragraphs.

Local Mode

In Local mode, all front panel controls are operational, and the instrument will respond to input commands over the bus. If addressed to talk, the HP 5373A sends data messages or the status byte. Whether addressed or not, the instrument responds to:

- Remote
- Local
- Local Lockout
- LCLL
- Trigger
- Abort

The HP 5373A can send a require-service message when in Local mode.

The HP 5373A always switches from Remote to Local when a GTL or LCLL message is received. The LCLL message sets the remote enable control line (REN) false. If the HP 5373A is in Local Lockout, the LOCAL key on the front panel is disabled. Instrument settings remain unchanged during Remote to Local transition.

Switching from Local to Remote Operation

The HP 5373A switches from Local to Remote when one of the following conditions occur:

- The listen address is received, while the REN (Remote Enable) control line is true.
- The REMOTE programming command is received (use the LOCAL command to return to Local when using the REMOTE command).

The instrument settings remain unchanged during Local-to-Remote transitions. The front panel RMT LED lights while the HP 5373A is in Remote mode.

HP-IB Default Startup Conditions

To establish a generally preferred set of conditions, default HP-IB states are established during power-up. The HP 5373A address and addressing mode are saved in non-volatile memory.

HP-IB default conditions are:

- HP-IB Local mode active
- Local Lockout cleared
- Unaddressed (if in normal addressing mode)
- RQS bit in the status byte register is set to 0
- Status byte register is cleared

Switching from Remote to Local Operation

Under certain predefined conditions the HP 5373A switches from Remote mode to Local mode. These conditions arise via commands sent from the controller or from internal changes.

Local Lockout

If the HP 5373A is under Remote (program) control and the front panel LOCAL key is inadvertently pressed, the instrument returns to Local control. Accordingly, data or settings could change. To prevent undesired front panel control, use the Local Lockout message. This message allows return-to-local only under program control.

You can cycle the power switch to return to local control; but, this technique has serious disadvantages (such as loss of data). Such action may make the system controller lose control of the instrument. During such a cycle, several HP-IB conditions are reset to the default state at power up, and the status of any operation that was in progress is unknown by the controller. It is possible that this could stop activity on the system bus as well.

Required Conditions

The HP 5373A switches from Remote to Local mode if one of the following conditions occurs:

- The GTL (Go To Local) interface command is received.
- The REN (Remote Enable) line goes false, in other words the LCLL meta-message is received.
- The LOCAL programming command is received. (Use this command when the REMOTE command is used to go to Remote.
- An internal return-to-local message goes active (generated by the front panel LOCAL key, but only if Local Lockout is inactive).

The HP 5373A settings remain unchanged during remote-to-local transitions. The front panel RMT LED is unlighted while the HP 5373A is in the Local mode.

Addressing

When the "TALK/LISTEN" HP-IB mode is selected from the front panel, the instrument may be addressed to talk or listen using the bus commands. When so addressed the instrument remains configured until you send an abort message (IFC), send a listen address matching the listen address of the instrument, or send a universal untalk command (UNT).

The instrument is shipped from the factory in the addressable mode, with the talk and listen adresses set to 3. You can set the HP 5373A to Talk Only. Talk Only mode enables limited bus operation without an HP-IB controller connected. You can display or change the addresses from the front panel.

When set to talk-only mode, the instrument does not respond to any bus message. Select this mode if you want to directly output data to a plotter or printer without using a controller.

Receiving the Data Message

The instrument responds to data messages when set to Remote mode and addressed to listen. A complete description of data messages is given in the chapter on HP-IB programming commands. Refer to that chapter for syntax and other detailed information.

Sending Data Messages

The instrument sends data messages when set to Local or Remote modes, when addressed to talk, or when in the Talk Only mode. Measurement results will be available for reading when the HP 5373A is a talker or when queried.

Receiving the Device Clear Message

In response to either the Device Clear message or the Selected Device Clear message, the HP 5373A:

- Clears the input and output buffers.
- Resets the command parser to enable parsing of the next message.
- Discards all deferred commands and queries.
- Terminates any measurement or acquisition process.

Receiving the Group Execute Trigger Message

The HP 5373A responds to the Group Execute Trigger (GET) command by triggering a measurement. That is, the instrument responds exactly as if it had received the *TRG or RESTart command to restart a measurement.

DATA MESSAGES

The HP 5373A communicates on the HP-IB primarily with data messages. The instrument interprets a byte on the eight data lines as a data message when the bus is in the data mode. The bus is in the data mode when the Attention (ATN) control line is false. If the ATN control line is true, the bus is in the command mode.

In the data mode, the HP 5373A can receive and send data messages. Input data messages include the program front panel functions and remote functions. Output data messages include instrument status information the setting of specific functions, measurement results, and the learn string.

The learn string is a binary data string that contains a condensed coding of the entire instrument state. Refer to SETup, SETup?, and CALibrate? in Chapter 5, Command Reference.

STATUS REPORTING AND SERVICE REQUEST

The HP 5373A can send a Service Request (SRQ) to the controller to indicate the need for attention, and can interrupt the current sequence of events. Typically, the status reporting structure of the HP 5373A sends an SRQ to indicate that data is ready for transmission or that an abnormal condition exists. See *Figure* 2-2 for information on the HP 5373A Status Reporting Structure.

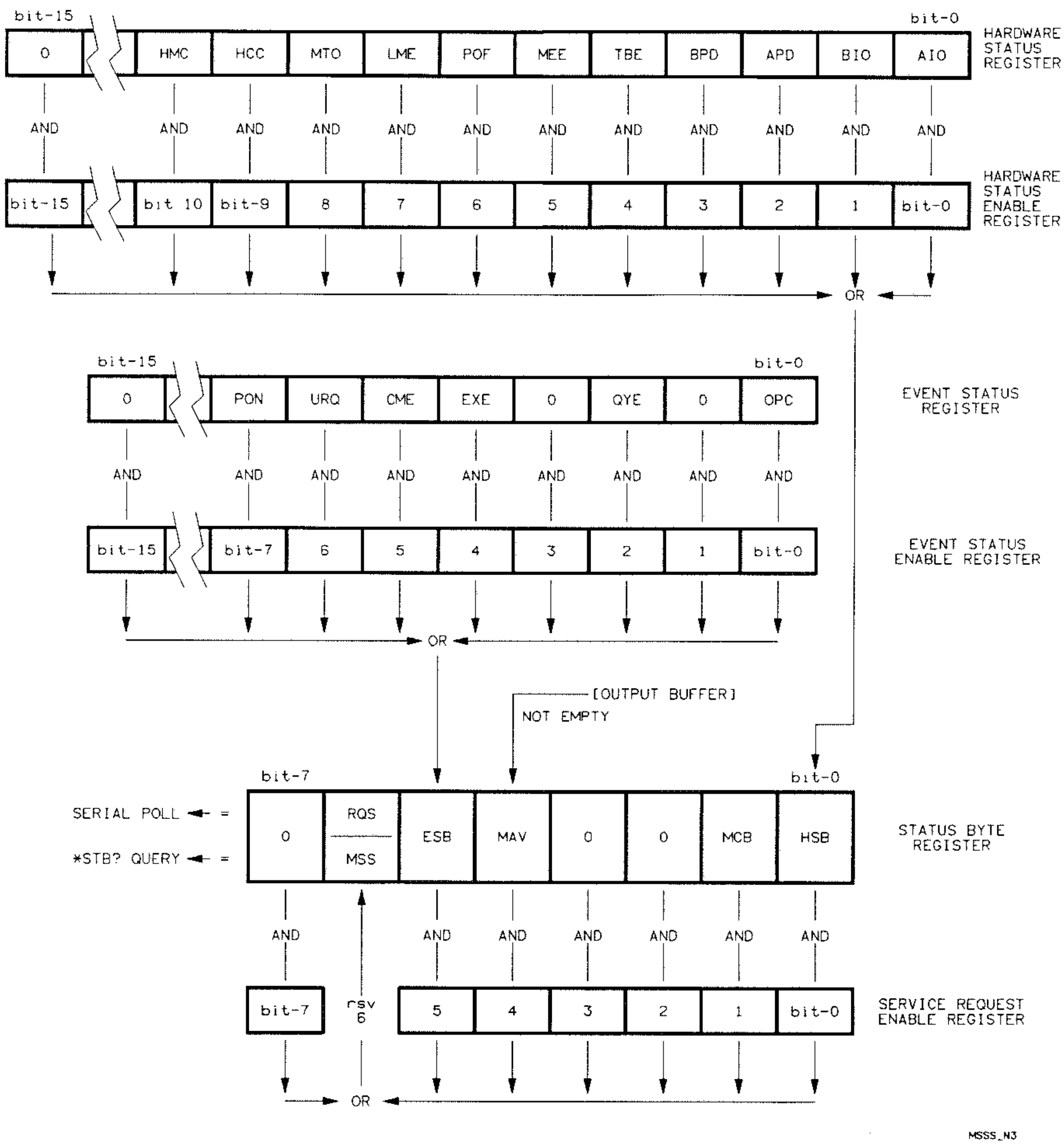


Figure 2-2. 5373A Status Reporting Structure

The status reporting structure of the HP 5373A consists of three status registers containing single-bit summary messages representing current hardware status or the occurrence of an event. Each status register has a corresponding enabling register so you can set the conditions under which the HP 5373A generates an SRQ.

Status Registers

The three status registers of the HP 5373A are:

- **Hardware Status Register** — indicates status of the measurement and input amplifier circuitry.
- **Event Status Register** — indicates parsing and execution errors.
- **Status Byte Register** — contains the summary messages from the Event Status and Hardware Status Registers. In addition, the Status Byte Register reports the status of the Output Buffer and measurement computer.

Status Enabling Registers

The three Status Enable Registers set the conditions under which the HP 5373A generates an SRQ. The HP 5373A sends an SRQ to the controller after a 0-to-1 transition of an enabled condition. The Enable Registers are:

- **Hardware Status Enable Register**
- **Event Status Enable Register**
- **Service Request Enable Register**

Each bit in an Enable Register is logical-ANDed with the corresponding bit in the associated Status Register. To enable a condition to generate an SRQ, the bit in the enable register corresponding to the desired condition must be set to 1.

All the enabled bits in the status register are logical-ORed with each other; the resultant summary of either the Hardware Status Register or the Event Status Register appears in the Status Byte Register (HSB, bit 0 and ESB, bit 5, respectively). If a bit in the Status Byte Register and the corresponding bit in the Service Request Enable Register are high, an SRQ will be generated.

HARDWARE STATUS REGISTER

The Hardware Status Register is a 16-bit register that reflects the status of the measurement hardware. These status conditions are summarized by the Hardware Status Bit (HSB) summary message, which appears in bit 0 of the Status Byte register.

Table 2-4 defines the status bits in the Hardware Status Register. Binary weighting is as shown.

Table 2-4. 5373A Hardware Status Register

Bit #	Binary Weight	Status Bit Condition
15 - 11	—	Not used; set to zero
10	1024	(HMC) High indicates measurement completed.
9	512	Hardcopy Complete (HCC) - High indicates that a hardcopy output (printer or plotter) is completed.
8	256	Measurement Timeout Error (MTO) - High indicates that a measurement timeout occurred.
7	128	Limits Error (LME) - High indicates that a measurement limit testing error occurred.
6	64	Power-On Failure (POF) - High indicates that a power-on failure occurred.
5	32	Missing Event Error (MEE) - High indicates that time interval measurement events were lost.
4	16	Time Base Error (TBE) - High indicates that the timebase oscillator is out of lock.
3	8	Channel B Pod (BPD) - High indicates that the Channel B input pod has been removed.
2	4	Channel A Pod (APD) - High indicates that the Channel A input pod has been removed.
1	2	Channel B Input Overload (BIO) - High indicates that the Channel B input amplitude exceeds the maximum level.
0	1	Channel A Input Overload (AIO) - High indicates that the Channel A input amplitude exceeds the maximum level.

Reading the Hardware Status Register

The Hardware Status Register is read with the *HSR? query. This query returns an integer value representing the sum of the binary-weighted values of the register bits. For example, a returned integer value of 68 (64+4) means that bit 6 (POF: Power-On Failure) and bit 2 (APD: Channel A Pod) of the Hardware Status Register are set to 1. When the Hardware Status Register is read, all register bits are cleared.

Clearing the Hardware Status Register

The Hardware Status Register is cleared by any one of the following:

- The *CLS command
- A power-on transition
- The read operation initiated by the *HSR? query.

HSR Program Example

You can use the hardware status register to effectively control the interactions between the instrument and the controller if a predetermined condition is met: for instance, to report a measurement block that contains an out of limit test condition.

An example program in Appendix F demonstrates how to use the HSR in a PRF or frequency measurement involving multiple blocks. The technique employs low and high limit tests. If either test fails, the HP 5373A enables the SRQ. This action triggers a query of the HSR. Based on the result of that query, a screen message gets displayed. Refer to the commented program listing for more details. (The same techniques apply to using the Event Status Register. For a description of this register, read on.)

HARDWARE STATUS ENABLE REGISTER

The Hardware Status Enable Register is a 16-bit register that allows one or more events in the Hardware Status Register to be reflected in the Hardware Status Bit (HSB) message. Each bit of the Enable register is ANDed with its corresponding bit in the Hardware Status Register; the resulting bits are ORed to determine the summary HSB message. The HSB message appears at bit 0 of the Status Byte Register. If HSB is set to 1, and bit 0 of the Service Request Enable Register is set to 1, an SRQ is generated.

Reading the Hardware Status Enable Register

The Hardware Status Enable Register is read with the *HSE? query. This query returns an integer value representing the sum of the binary-weighted values of the register bits, as previously described for the Hardware Status Register. For example, a returned integer value of 68 (64+4) indicates that bit 6 and bit 2 of the Hardware Status Enable Register are set to 1, thus enabling bit 6 (POF: Power-On Failure) and bit 2 (APD: Channel A Pod) of the Hardware Status Register.

Writing to the Hardware Status Enable Register

The Hardware Status Enable Register is written to with the *HSE command. The numeric argument of this command must be the integer representation of the sum of the binary-weighted values of the enabled bits. Using the example of the previous paragraph, an integer value of 68 would be written to the register to set bit 6 and bit 2 to 1.

Clearing the Hardware Status Enable Register

The Hardware Status Enable Register is cleared by:

- Sending the *HSE command with a numeric data value of zero.
- A power-on transition, if *PSC is true.

EVENT STATUS REGISTER

The Event Status Register is a 16-bit register that reflects generic error conditions and operating states. These status conditions are summarized by the Event Status Bit (ESB) summary message, which appears in bit 5 of the Status Byte Register.

Table 2-5 defines the status bits contained in the Event Status Register, as well as the specific HP 5373A implementation. Note that the HP 5373A uses only some of the available bits in the register; those not used are always set to zero.

Reading the Event Status Register

The Event Status Register is read with the *ESR? query. This query returns an integer value representing the sum of the binary-weighted values of the register's bits. For example, a returned integer value of 36 (32+4) would indicate that bit 5 (CME: Command Error) and bit 2 (QYE: Query Error) of the Event Status Register were set to 1. When the Event Status Register is read, all bits are cleared.

Clearing the Event Status Register

The Event Status Register is cleared by:

- The *CLS command
- A power-on transition
- The read operation initiated by the *ESR? query

Table 2-5. 5373A Event Status Register

Bit #	Binary Weight	Name and Defined Event	5373A Implementation
15 - 8	—	(Reserved by IEEE)	Not used; set to zero
7	128	Power-On (PON) - High indicates that the ac power has been cycled.	(same)
6	64	User Request (URQ) - High indicates that Local control has been activated.	LOCAL key pressed
5	32	Command Error (CME) - High indicates that a syntax or semantic error has occurred.	(same)
4	16	Execution Error (EXE) - High indicates that invalid range values or conflicting settings have been selected.	(same)
3	8	Device-Dependent Error (DE) - Not defined.	Not used; set to zero
2	4	Query Error (QYE) - High indicates that no output is present or pending.	(same)
1	2	Request Control (RQC) - High indicates that Controller mode is requested by the instrument.	Not used; set to zero
0	1	Operation Complete (OPC) - High indicates that parsing is completed.	(same)

EVENT STATUS ENABLE REGISTER

The Event Status Enable Register is a 16-bit register that allows one or more events in the Event Status Register to be reflected in the Event Status Bit (ESB) summary message. Each bit of the Enable register is ANDed with its corresponding bit in the Event Status Register; the Enable register bits are ORed to determine the summary ESB message. The ESB message appears at bit 5 of the Status Byte Register. If ESB is set to 1, and bit 5 of the Status Byte Enable Register is set to 1, an SRQ is generated.

Reading the Event Status Enable Register

The Event Status Enable Register is read with the *ESE? query. This query returns an integer value representing the sum of the binary-weighted values of the register bits, as previously described for the Event Status Register. For example, a returned integer value of 36 (32+4) indicates that bit 5 and bit 2 of the Event Status Enable Register are set to 1, thus enabling bit 5 (CME: Command Error) and bit 2 (QYE: Query Error) of the Event Status Register.

Writing to the Event Status Enable Register

The Event Status Enable Register is written to with the *ESE command. The numeric argument of this command must be the integer representation of the sum of the binary-weighted values of the enabled bits. Using the example of the previous paragraph, an integer value of 36 would be written to the register to set bit 5 and bit 2 to 1.

Clearing the Event Status Enable Register

The Event Status Enable Register is cleared by:

- Sending the *ESE command with a numeric data value of zero.
- A power-on transition, if *PSC is true.

STATUS BYTE REGISTER

The Status Byte Register (*Table* 2-6) is an 8-bit register containing the summary messages generated by the Hardware Status Register, the Event Status Register, Output Buffer, and other instrument conditions. It also contains the Request Service (RQS)/Master Status Summary message (in bit 6).

Table 2-6. 5373A Status Byte Register

Bit #	Binary Weight	Status Bit Condition
7	128	Not used.
6	64	RQS/MSS (Request Service/Master Status Summary) - High indicates that the 5373A has a reason for requesting service.
5	32	Event Status Summary (ESB) - High indicates that an event in the Event Status Register (ESR) has occurred.
4	16	Message Available (MAV) - High indicates that the 5373A is ready to output data.
3	8	Not used.
2	4	Not used.
1	2	Measurement complete (MCB).
0	1	Hardware Status Summary (HSB) - High indicates that an event in the Hardware Status Register (HSR) has occurred.

Summary Messages

As shown in *Table 2-6*, four bits in the Status Byte Register contain summary messages reflecting instrument status. The summary messages are defined below.

MASTER STATUS SUMMARY (MSS) MESSAGE

When high, the MSS summary message indicates that the HP 5373A has a reason for requesting service. The MSS message appears in bit 6 of the Status Byte Register. The other 7 bits of the Status Byte Register are ANDed with the corresponding bits in the Service Request Enable Register, and the resulting values are ORed to determine the MSS summary message.

Although the MSS message is sent as bit 6 of the response to the *STB? query, it is not sent in response to a serial poll. Instead, the RQS (Request For Service) message is returned when the HP 5373A is serial polled.

EVENT STATUS BIT (ESB) SUMMARY MESSAGE

When high, the ESB summary message indicates that one or more enabled events in the Event Status Register occurred since this register was last read or cleared.

The ESB message appears in bit 5 of the Status Byte Register. The bits of the Event Status Register are ANDed with the corresponding bits of the Event Status Enable Register, and the resulting values are ORed to determine the ESB message.

MESSAGE AVAILABLE (MAV) MESSAGE

When high, the MAV summary message indicates that the HP 5373A is ready to accept a request by the controller to output data bytes. The MAV message will be set low when the output buffer is empty.

The MAV message appears in bit 4 of the Status Byte Register. This message can be used to synchronize data transfers with the controller. After sending a query to the HP 5373A, the controller can enable the MAV bit (by setting the Service Request Enable Register) to generate a request for service when data becomes available. During this time, the HP-IB interface is available for other use, since it is not used by the HP 5373A.

MEASUREMENT COMPLETE (MCB)

This bit will go high when a measurement has completed. It will be set low when the current measurement is cleared or when a new measurement is started. This bit can be used to synchronize data transfers to a computer.

HARDWARE STATUS BIT (HSB) SUMMARY MESSAGE

When high, the HSB summary message indicates that one or more enabled events in the Hardware Status Register occurred since this register was last read or cleared.

The HSB message appears in bit 0 of the Status Byte Register. The bits of the Hardware Status Register are ANDed with the corresponding bits of the Hardware Status Enable Register, and the resulting values are ORed to determine the HSB message.

Reading the Status Byte Register

The Status Byte Register may be read by a serial poll or with the *STB? query. Both methods will return an integer value representing the sum of the binary-weighted values of the bits of the register, as described for the Hardware Status and Event Status Registers. The values returned for bit 6, however, will depend on the method used:

- When serial polled, bit 6 is the RQS message, indicating whether the HP 5373A is actively requesting service (by asserting the SRQ interface signal). The RQS message will be set low (False) following a serial poll.
- When queried with the *STB? query command, the MSS message is sent as bit 6, in place of the RQS message.

In either case, reading the Status Byte Register does not alter the summary messages in the register.

Clearing the Status Byte Register

The Status Byte Register can be cleared indirectly by sending the *CLS command. Since the *CLS command clears all event registers, their corresponding summary messages will also be cleared. The output buffer, measurement complete, and the MAV summary message are exceptions; they are not cleared by *CLS.

SERVICE REQUEST ENABLE REGISTER

The Service Request Enable Register is an 8-bit register that enables corresponding summary messages in the Status Byte Register. When enabled, a summary message will generate a Request Service message.

Reading the Service Request Enable Register

The Service Request Enable Register is read with the *SRE? query. The *SRE? query returns an integer value representing the sum of the binary-weighted values of the bits of the register. For example, a returned value of 48 (32+16) indicates that bit 5 and bit 4 of the Service Request Enable Register are set to 1, thus enabling bit 5 (ESB) and bit 4 (MAV) of the Status Byte Register. Note that the value of bit 6 will always be zero.

Writing to the Service Request Enable Register

The Service Request Enable Register is written to by the *SRE command. The numeric argument of the *SRE command must be the integer representation of the sum of the binary-weighted values of the enabled bits. Using the example of the previous paragraph, an integer value of 48 would be written to the register to set bit 5 and bit 4 to 1.

Clearing the Service Request Enable Register

The Service Request Enable Register is cleared by:

- Sending the *SRE command with a data value of zero.
- A power-on transition, if *PSC is true.

RECEIVING THE ABORT MESSAGE

The ABORT message (IFC control line true) halts all bus activity. When the instrument receives the abort message, it becomes unaddressed and stops talking or listening. The Request Service message (RQS) and the Status Byte are unaffected by the abort message.

ASCII AND FLOATING POINT FORMATS

You can get measurement results in either ASCII or Floating Point Format. Appendix E lists annotated HP BASIC program examples that demonstrate programming techniques (for ASCII, Binary, and Floating Point).

Measurement Messages and Data Types

Measurement results are sent in measurement messages. Measurement messages contain coded information about measured parameters such as PRF, Frequency, PRI, Period, Events and so forth. Information contained within these messages is represented by one of three data types:

- numeric
- block
- character

In general, the HP 5373A sends measurement results in numeric or block formats. Status information is sent in character format.

Choosing an Appropriate Format

The 5373A offers three measurement result formats: ASCII, binary, or floating point. The data format you choose depends upon your particular requirements. Output rates and ease of use are two factors to consider. Your options are:

- Use processed data (ASCII and floating point format from the HP 5373A)
- Transfer raw time and events data to a controller via the HP-IB and perform your own calculations (binary).

Measurement Output Rates

Table 2-7 compares measurement output rates for the three data formats. For example, the Normal Binary output rate for the Continuous Time Interval mode is shown in the table to be 19,500 measurements per second. This should be interpreted as "over 19 1000-measurement blocks can be transferred in one second to the controller."

The values in *Table 2-7* are typical values. Performance is also affected by other instrumentation on the bus, the performance of the external controller, and the particular measurement software.The following conditions apply:

- The HP 5373A is set to the PRESET condition, and then the appropriate measurement function is selected.
- Statistical and Math functions are disabled.
- The MANUAL input triggering mode is used to set the input voltage trigger levels.
- Except where noted in the table, a sample size of 10 blocks of 1000 measurements (10,000 total measurements) is used to obtain the values.
- All values include the measurement time as well as the transfer time of the data using an input signal of 13 MHz (76.9 ns).
- For BINARY output rates, the values represent the number of measurements sent to the controller and stored in a buffer without processing. Processing time in the controller will vary with the controller, the program language, and the particular program.

Table 2-7. Measurement Output Rates

Measurement Function	ASCII*	Floating Point*	Binary*	
			Normal	Fast
Time Interval	550	1200	12,000	16,000
Continuous Time Interval	650	2000	19,500	25,000
±Time Interval[1]	550	1200	12,000	16,000
Frequency (Single-Channel)	600	1600	14,000	20,000
Frequency (Dual-Channel, Single-Result)	450	750	6,000	9,500
Frequency (Dual-Channel, Dual-Result)	300	750	6,000	9,500
Totalize (Single-Channel)[2]	650	2100	8,000	n/a
Totalize (Dual-Channel)[2]	550	1400	6,000	n/a
Rise/Fall Times, Pulse Width, Pulse Offtime, Duty Cycle	500	1000	12,000	16,000
Phase (Dual-Channel)	450	850	12,000	16,000
Phase Deviation	600	1500	14,000	20,000
Frequency Deviation	600	1500	14,000	20,000
Time Deviation	650	1000	14,000	20,000
Histogram TI, Histogram ±TI, Histogram Continuous TI[3]	650	3000	n/a	n/a
Envelope Power[4], Amplitude Modulation[4]	8	8	n/a	n/a

NOTES TO TABLE 2-7:

1. Event/Event Arming for ±Time Interval A or B.
2. Interval Sampling of 100 ns.
3. Characterized using full Histogram format; consequently, all 200 Histogram bin counts are transferred. Results are bin counters per second.
4. Characterized using ten measurements.

Aborting a Measurement

Sometime you may want to abort a measurement before the process is complete. A program listed in Appendix E demonstrates how to abort any HP 5373A measurement. Use the ABORT command, to halt measurement at any time. You can retrieve all data captured up to the time the ABORT command executes.

BINARY OUTPUT FORMAT

When set for Binary data output, the HP 5373A outputs raw data from its internal count registers. For more information on use of the Binary output format, see Chapter 3.

HOW TO USE ASCII AND FLOATING POINT FORMATS

ASCII and floating point data formats are described in the following paragraphs. See Appendix E for example programs which demonstrate use of data in these formats.

ASCII Measurement Result Format

Use the ASCII measurement result format to transmit processed measurement data. To accomodate a wide range of values, the ASCII measurement results are formatted as illustrated in the syntax diagrams in *Figure* 2-3.

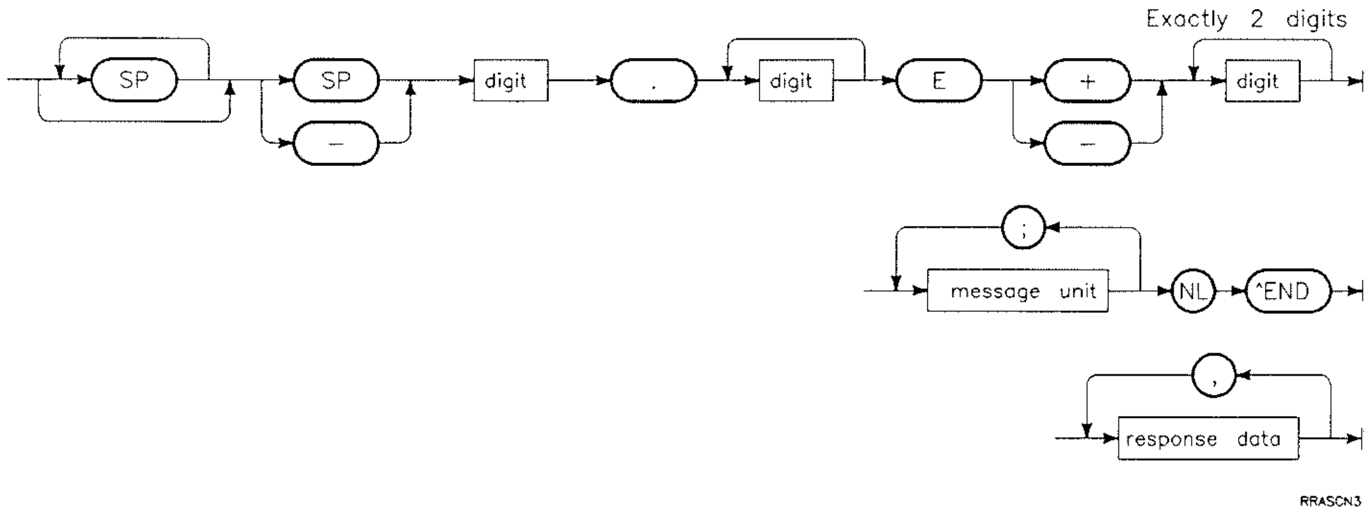


Figure 2-3. ASCII Measurement Result Syntax Diagrams

ASCII RESULT FORMAT

Each ASCII measurement result consists of a fixed-length, 22-character data field (top of *Figure* 2-3). To maintain a constant length, leading spaces right-justify the number within the field. Embedded or trailing spaces are suppressed. Leading spaces, if any, are followed by the algebraic sign of the number. For positive values and zero, the sign is sent as the ASCII space <SP> character; otherwise, a minus sign <–> precedes the first digit of the number. All representations of zero are expressed as "<SP>0.0E+00" (there is no representation for negative zero). The block labled digit (to the right and left of the decimal point) represents one of the 7-bit ASCII codes from 48 through 57 (decimal), which correspond to the numerals 0 through 9, respectively. A decimal point <.> is always sent as part of the ASCII measurement result message. The exponent field consists of exactly two digits. It is expressed in scientific format, and unscaled units are always assumed.

Measurement Message

An ASCII measurement message is one or more message units that completely describe the measurement result. When multiple data fields are contained within a measurement message, they are explicitly delimited by the ASCII semicolon <;>. Within each message unit are the data elements. When several data elements comprise a message unit, each data element is delimited
by the ASCII comma <,>. To terminate the complete message, the <NL>,^END sequence is sent at the end of the final message unit.

Single Message Unit

A single message unit is terminated by the concurrent transmission of the EOI signal with the ASCII new line <NL> character, represented in the format examples in this manual by "<NL>^END".

ASCII Measurement Result Format Examples

The following ASCII output format examples are divided into categories. In the first group, specific numeric values are shown to illustrate formatting of results. In the second group, alphanumeric descriptions (such as <resultA>) are used to simplify the examples. In the third group, statistics, examples of raw statistics data are given followed by examples relating to limit testing.

NOTE

The value 1.0E+38 is transmitted instead of a calculated result if the calculated result is invalid. For example, if a Totalize A/B result has 0 for the Channel B denominator, 1.0E+38 is transmitted.

Numeric Examples

Single Result:

<SPs> 1.23456789E+08 <NL>^END	PRF A = 123.456789 MHz
<SPs> -5.371E-03 <NL>^END	Period B = -5.371E-03
<SPs> 1.0E-09 <NL>^END	Time Interval A→ B = 1 ns
<SPs> 0.0E+00 <NL>^END	Time Interval A→ B = 0 ns

Note that <SPs> denotes one or more leading spaces.

Multiple Results:

Continuous Frequency A:

```
<SPs> 2.04E+07 <;>          Frequency A [#1] = 20.4 MHz
<SPs> 2.05E+07 <;>          Frequency A [#2] = 20.5 MHz
      .                           .
      .                           .
      .                           .
<SPs> 2.01E+07 <NL>^END     Frequency A [#n] = 20.1 MHz
```

Continuous Frequency A&B:

```
<SPs> 2.04E+07 <,>          Frequency A [#1] = 20.4 MHz
<SPs> 5.00E+03 <;>          Frequency B [#1] = 5.00 kHz
<SPs> 2.05E+07 <,>          Frequency A [#2] = 20.5 MHz
<SPs> 4.99E+03 <;>          Frequency B [#2] = 4.99 kHz
      .                           .
      .                           .
      .                           .
<SPs> 2.01E+07 <,>          Frequency A [#n] = 20.1 MHz
<SPs> 5.20E+03 <NL>^END     Frequency B [#n] = 5.20 kHz
```

NOTE

Dual channel results are transferred in the sequence A,B;A,C;B,C for simultaneous two-channel measurements.

Alphanumeric Examples

SECOND GROUP:

In the following examples, a simplified notation represents the different formats of response elements . To receive this data, the Numeric screen must be in either NUMERIC, BOLD, or SPLIT mode. In the following examples:

<res A> = result for the primary measurement channel

<res B> = result for the secondary measurement channel

<limit A, limit B> = "PASS", "HIGH", "LOW", "INS", or "NA"

<gate A> = gate time for the primary measurement channel

<gate B> = gate time for the secondary measurement channel

<mean>...<allan variance> = the statistical values available

<low values A, low values B> = the number of values which fell below the low limit

<pass values A, pass values B> = the number of values which fell between the low and high limits; or if the limits are reversed the number of values outside the low and high limits.

<high values A, high values B> = the number of values which fell above the high limit

<inside values A, inside values B> = the number of values which fell inside the reversed limits.

<NA values A, NA values B> = the number of values for which limit testing was not applicable. These included time stamps, inter-block data, data extended by Inhibit, and invalid data.

Single Result with Limit Testing:

```
<res A> <,> <limit A> <;>
          .
          .
          .
<res A> <,> <limit A> <NL>^END
```

Single Result with Gate Time Data:

```
<res A> <,> <gate A> <;>
          .
          .
          .
<res A> <,> <gate A> <NL>^END
```

Single Result with Limit Testing and Gate Time Data:

```
<res A> <,> <limit A> <,> <gate A> <;>
          .
          .
<res A> <,> <limit A> <,> <gate A> <NL>^END
```

Dual Result with Limit Testing:

```
<res A> <,> <limit A> <,> <res B> <,> <limit B> <;>
          .
          .
          .
<res A> <,> <limit A> <,> <res B> <,> <limit B> <NL>^END
```

Dual Result with Gate Time Data:

<res A> <,> <res B> <,> <gate A> <,> <gate B> <;>
.
.
.
<res A> <,> <res B> <,> <gate A> <,> <gate B> <NL>^END

Dual Result with Limit Testing and Gate Time Data:

<res A> <,> <limit A> <,> <res B> <,> <limit B> <,>
<gate A> <,> <gate B> <;>
.
.
.
<res A> <,> <limit A> <,> <res B> <,> <limit B> <,>
<gate A> <,> <gate B> <NL>^END

Statistics Examples

When Statistics mode is enabled and the HP 5373A is displaying the STATISTICS Numeric screen, the statistical data is sent instead of the measurement results data . Note that when the HP 5373A is displaying the SPLIT Numeric screen (which displays results plus four of the statistical values: Maximum, Mean, Minimum, and Standard Deviation), all eight statistical values are still returned over the bus, after the applicable measurement results (as shown above) have been sent.

Single Result Statistics:

<mean A> <;> <std dev A> <;> <max A> <;> <min A><;>
<variance A> <;> <root variance A> <;> <rms A> <;>
<allan variance A> <NL>^END

Dual Result Statistics:

<mean A> <;> <std dev A> <;> <max A> <;> <min A>
<;> <variance A> <;> <root variance A> <;> <rms A> <;> <allan variance A> <;>
<mean B> <;> <std dev B> <;> <max B> <;> <min B>
<;> <variance B> <;> <root variance B> <;> <rms B> <;>
<allan variance B> <NL>^END

Limit Testing (Statistics) Examples

When the HP 5373A displays the LIMITS screen, the statistical data above is not available. Instead, information about the number of measurements which passed and failed the limit testing (high, low, and inside limits) is sent. The format is:

Single Result in LIMIT Screen:

<low values A> <;> <pass values A> <;> <inside values A> <;> <NA values A> <;>
<high values A> <;> <NL>^END

Dual Result in LIMIT Screen:

<low values A> <;> <pass values A> <;> <inside values A> <;> <NA values A> <;>
<high values A> <;> <low values B> <;> <pass values B> <;> <inside values B> <;>
<NA values B> <;> <high values B> <NL>^END

Floating Point Measurement Result Format

The floating point format (*Figure* 2-4) offers faster transfer rates (compared to ASCII rates) by sending each result in a packed, eight-byte format. The HP 5373A implements the double precision (64-bit) floating point representation specified by ANSI/IEEE Standard 754-1985.

The floating point block message syntax is shown in *Figure* 2-4. Note that each block of data is terminated with EOI, denoted in this text as ^END.

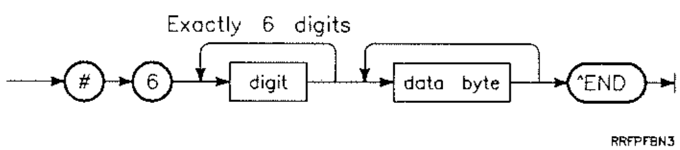


Figure 2-4. Floating Point Format Block Message Syntax

Floating Point Format Data Structure

Each floating point measurement result consists of eight bytes. Each result is represented as follows:

```
Byte #1   Byte #2 . . . .                                                Byte #8
SEEEEEEE EEEEFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF
```

where

S is the sign bit

E represents an 11-bit exponent biased by 1023

F represents a 52-bit fraction which, together with an implicit leading 1, yields the significant digit field "1.-"

Therefore, a real number (N) can be defined as follows:

1. If E is greater than or equal to 1, then:

   $N = (-1)^S * 2^{E-1023} * (1.F)$

2. If E equals 0, then:

   $N = (-1)^S * 2^{-1022} * (0.F)$

3. If E = F = 0, then:

   N = 0

Floating Point Measurement Result Format Examples

The following floating point output format examples are divided into categories. In the first group, specific numeric values are shown to illustrate formatting of results. In the second group, alphanumeric descritions (such as <result A>) are used to simplify the examples. In the third group, statistics, examples of raw statistics data are given followed by examples relating to limit testing.

NOTE

The value 1.0E+38 is transmitted instead of a calculated result if the calculated result is invalid. For example, if a Totalize A/B result has 0 for the Channel B denominator, 1.0E+38 is transmitted.

Single result example

Frequency A = 10 MHz:

<#>	Block
<6>	Preamble
<0>	
<0>	
<0>	Length byte
<0>	
<0>	
<8>	
01000001	
01100011	
00010010	
11010000	10.0 MHz
00000000	
00000000	
00000000	
00000000	
^END	

Multiple result example

Frequency A&B (Frequency A ≅10 MHz, Frequency B ≅ 5 MHz):

<#>	Block
<6>	Preamble
<0>	
<0>	
<0>	Length byte
<0>	
<1>	
<6>	
01000001	
01100011	
00010010	
11010000	10.0 MHz
00000000	
00000000	
00000000	
00000000	
01000001	
01010011	
00010010	
11010000	5.0 MHz
00000000	
00000000	
00000000	
00000000	
^END	

NOTE

> *Dual channel results are transferred in the sequence A,B;A,C;B,C for simultaneous two-channel measurements.*

Alphanumeric Examples

SECOND GROUP:

In the following examples, a simplified notation represents the different formats of response elements . To receive this data, the Numeric screen must be in either NUMERIC, BOLD, or SPLIT mode. In the following examples:

<header> = header consisting of "#6" and 6 digits giving length

<res A> = result for the primary measurement channel

<res B> = result for the secondary measurement channel

<limit A, limit B> = 0E0 for pass results
1E0 for high results
–1E0 for low results
1E38 for inside results
–1E38 for NA results

<gate A> = gate time for the primary measurement channel

<gate B> = gate time for the secondary measurement channel

<mean>...<allan variance> = the statistical values available

<low values A, low values B> = the number of values which fell below the low limit

<pass values A, pass values B> = the number of values which fell between the low and high limits; or if the limits are reversed the number of values outside the low and high limits.

<high values A, high values B> = the number of values which fell above the high limit

<inside values A, inside vaues B> = the number of values which fell inside the reversed limits.

<NA values A, NA values B> = the number of values for which limit testing was not applicable. These included time stamps, inter-block data, data extended by Inhibit, and invalid data.

Single Result with Limit Testing:

```
<header><res A> <limit A>
              .
              .
              .
<res A> <limit A> ^END
```

Single Result with Gate Time Data:

```
<header><res A> <gate A>
              .
              .
              .
<res A> <gate A> ^END
```

Single Result with Limit Testing and Gate Time Data:

```
<header><res A> <limit A> <gate A>
              .
              .
<res A> <limit A> <gate A> ^END
```

Dual Result with Limit Testing:

```
<header><res A> <limit A> <res B> <limit B>
              .
              .
              .
<res A> <limit A> <res B> <limit B> ^END
```

Dual Result with Gate Time Data:

```
<header><res A> <res B> <gate A> <gate B>
              .
              .
              .
<res A> <res B> <gate A> <gate B> ^END
```

Dual Result with Limit Testing and Gate Time Data:

```
<header><res A> <limit A> <res B> <limit B> <gate A> <gate B>
              .
              .
              .
<res A> <limit A> <res B> <limit B> <gate A> <gate B> ^END
```

Statistics Examples

When Statistics mode is enabled and the HP 5373A is displaying the STATISTICS Numeric screen, the statistical data is sent instead of the measurement results data . Note that when the HP 5373A is displaying the SPLIT Numeric screen (which displays results plus four of the statistical values: Maximum, Mean, Minimum, and Standard Deviation), all eight statistical values are still returned over the bus, after the applicable measurement results (as shown above) have been sent.

Single Result Statistics:

<header><mean A> <std dev A> <max A> <min A>
<variance A> <root variance A> <rms A> <allan variance A> ^END

Dual Result Statistics:

<header><mean A> <std dev A> <max A> <min A> <variance A> <root variance A>
<rms A> <allan variance A> <mean B> <std dev B> <max B> <min B> <variance B>
<root variance B> <rms B> <allan variance B> ^END

Limit Testing (Statistics) Examples

When the HP 5373A displays the LIMITS screen, the statistical data above is not available. Instead, information about the number of measurements which passed and failed the limit testing (high, low, and inside limits) is sent. The format is:

Single Result in LIMIT Screen:

<header><low values A> <pass values A> <inside values A> <NA values A>
<high values A> ^END

Dual Result in LIMIT Screen:

<header><low values A> <pass values A> <inside values A> <NA values A>
<high values A> <low values B> <pass values B> <inside values B> <NA values B>
<high values B> ^END

3 BINARY OUTPUT

3

BINARY OUTPUT

INTRODUCTION

Binary data output is the one of the fastest ways of transferring data from the HP 5373A — 50 to 100 times faster than other formats. Transfer rates of 19,500 measurements per second or more can be achieved using the binary output. Much of this speed is achieved because the binary output is unprocessed data from the HP 5373A measurement circuits. Binary data must be processed externally in order to obtain measurement results (PRF, Frequency, Time Interval, etc.). The data may also be combined in new ways to fulfill the needs of special applications. This chapter describes the various binary output data formats, how to process the data from these formats, and how to combine the processed data to obtain measurement results.

How Binary Data Is Created

The HP 5373A captures data in two types of count registers: event and time. Both are 32-bit counters consisting of two 16-bit integrated circuits in series. The accuracy and resolution of the time counter is increased by a circuit called an interpolator. See *Figure 3-1*.

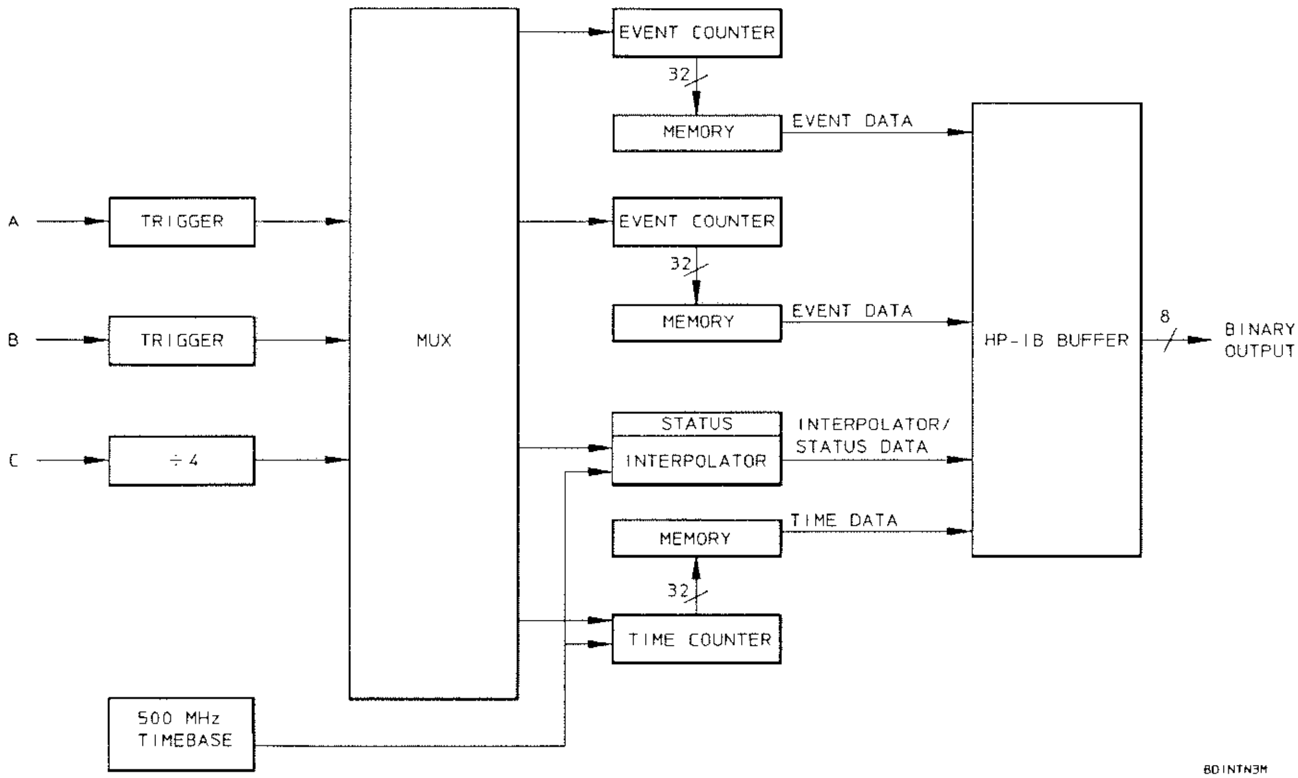


Figure 3-1. Simplified Block Diagram of HP 5373A Measurement Section with Binary Output

EVENT COUNTER Inputs for the event counter come from Channel A, B, or C. When a signal is applied to Channel A or B, the event counter increments every time the signal meets the selected trigger criteria. To increase count rate, Channel C contains a high speed 4:1 frequency divider. When measurements are made via Channel C, the event counter increments on every fourth cycle of the input signal.

TIME COUNTER The time counter runs continuously, counting the 500 MHz internal time base clock. This counter keeps track of the time at which input events occur. The 500 MHz clock frequency gives the time counter a resolution of 2 ns. To increase resolution, an interpolator circuit measures the small amount of time between the occurrence of an event count, and the next 500 MHz clock pulse. See *Figure* 3-2.

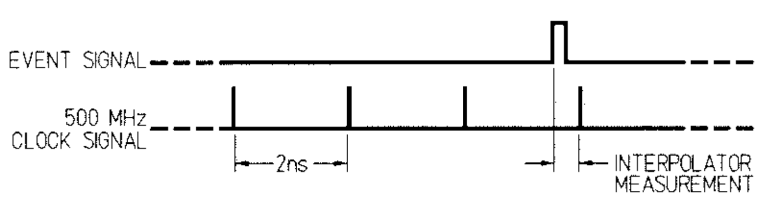


Figure 3-2. Interpolation Between 2 ns Clock Pulses

MAKING MEASUREMENTS

Although the HP 5373A is a complex instrument, most of the measurement data produced in the binary mode originates in the circuits described above. To understand how the output data is formed, consider the following simplified Frequency measurement. The input frequency is <10 MHz and the measurement uses the Automatic arming mode.

The measurement begins when an input signal is applied. Every time the input signal meets the input trigger and arming requirements, a pulse is generated which increments the event counter. This makes three things happen:

1. The count in the event counter is instantly transferred to memory.

2. The interpolator begins to measure the time between the event pulse and the next 2 ns clock pulse to be captured in the time counter.

3. At the next time count, the number in the time counter is transferred to memory.

When the next event occurs, the process is repeated. Periodically, event, time and interpolator data is transferred to the HP-IB buffer. There it is formatted and sent to the rear panel HP-IB port as binary data.

How Measurement Results are Generated

To generate measurement results, the binary data must be captured and processed by your program in an external computer. In simplified form, the process is:

1. The event count data is examined to be sure that the event counter has not overflowed during the measurement. If the event counter has overflowed, a correction is applied so that the data increases continuously. Processed event data is called event stamps.

2. The time count data is processed for overflow in the same way as event data, then multiplied by 2 ns to convert it to time data.

3. Interpolator data is also converted to time data, then subtracted from the time-count data. Time and interpolator data combined in this manner are called time stamps. Time stamps, because of the interpolator, have a resolution of 200 ps.

PRF or Frequency, as well as all other measurement results, are computed from the time and event stamps. To compute PRF or Frequency, the process is:

1. Subtract adjacent event stamps to get elapsed events.
2. Subtract adjacent time stamps to get elapsed time.
3. Compute PRF or Frequency by dividing elapsed events by elapsed time. The formula is:

$$\text{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

To review our simple example, an event count occurred at each end of an input cycle. The time counter logged the precise time of occurrence of each event. Subtracting adjacent event stamps gives, in our simple example, one (cycle). Subtracting the corresponding time stamps gives the time for one cycle (the PRI or Period). Performing the division (events/time) gives PRF or frequency. Conversely, dividing time by events gives PRI or Period.

If the input PRF or frequency is greater than 10 MHz, not all cycles will be timed, but all will be counted in the event counter. Since all events are counted, accuracy of the measurement is not affected.

Time Interval measurements are made in a similar manner. Start and stop signals, when accumulated in the event counters, result in the generation of time data, (similar to the PRF or Frequency measurement). Time Interval results, the time between start and stop signals, are computed from the resulting time stamps.

Binary Output Constraints

The binary output mode operates with all HP 5373A measurements except:

- Histogram measurements
- Envelope Power measurements
- Amplitude Modulation measurements
- Totalize (FAST mode)

All other measurement modes can be used with the binary output, and are covered in this chapter. In order to simplify the descriptions in this chapter, the following assumptions are made:

- Multiple block measurements are not used.
- Pre-trigger is OFF.

- The HP 5373A is in the Normal mode (not Fast).
- The Inhibit function is not being used.
- Manually Gated Totalize measurements are not being made.

For more information on the use of binary output in these modes, refer to the *Special Topics* section in this chapter.

NOTE

HP-IB query commands are not allowed in the Binary output mode. Any output in response to a query, while the HP 5373A is in the binary mode, will be unreadable.

What Follows

As you may suspect, actual measurements are somewhat more complex than the examples. The remainder of this chapter describes the complexities.

The HP 5373A can output binary data in 20 different formats. The specific output format depends on the measurement function used (PRF or Frequency, Time Interval, etc.), the specified arming mode, and whether or not you have requested extra data in addition to the measurement data (Expanded Data ON/OFF). All of the output data formats are described in the *Binary Format* section at the end of this chapter. To find the format for your measurement, use *Table 3-1, Format Guide,* in the *Binary Format* section. This table lists all measurement and arming mode combinations supported on the binary output, and indicates the output data format for each. The 20 binary output format descriptions immediately follow the table.

Each individual format section describes the specific data format, and gives detailed instructions for reducing the binary output data to measurement results.

Appendix E, Example Programs, contains programs for receiving and processing binary data. These programs, written in HP BASIC, show the details of setting up arrays, manipulating the binary data, computing time and event stamps, and producing measurement results.

To write your data capture and reduction program, you should be familiar with the manual method for making measurements on the HP 5373A, then read the procedures which follow.

HOW TO PRODUCE MEASUREMENT RESULTS

General Information

In order to obtain measurement results from the binary data output, you must write a program to organize and process the data. Writing this program requires that you have a good working knowledge of programming and programming techniques. To properly process the data, the program you write must do the following things:

- Set up an array or several arrays to store the data.
- Collect the data from the HP 5373A HP-IB port and store it.
- Convert the 32-bit (4-byte) binary data to a form which is easy to manipulate in your programming language.
- Scan the data to detect counter rollovers and apply corrections.
- Compute event stamps (see *Glossary*) by correcting event count data. Event stamps represent the cumulative number of input events in a measurement.
- Compute time stamps (see *Glossary*) from time count and interpolator data. In some measurements a correction for internal signal path length is also applied.
- Compute measurement results from time and event stamps.

To write the program you should be familiar with the process for manually making measurements on the HP 5373A. Work with the Getting Started Manual initially, then use the Operating Manual as needed to set up and make your measurements. In operation, the HP 5373A can be set up via the HP-IB, then switched to the binary output mode with the appropriate code (INT;OUTP,BIN). If you do this manually, use the **RESULTS FORMAT** field in the **System** menu.

HOW TO WRITE YOUR PROGRAM

This chapter is organized to help you write your program as efficiently as possible. The process outlined below will help you quickly obtain the information you need.

Getting Started

The first step is to familiarize yourself with the general characteristics of the binary output format, the data structure, an overview of how to process the data, specific terms used, etc. We recommend the following steps.

1. Read the following material in this chapter:
   - *Header Information:* tells how to recognize the beginning of new data and how to determine the number of bytes in each binary data transmission.
   - *Measurement Data and Data Fields:* how the various data fields are organized.
   - *How to Process Binary Data:* general description of the steps required to turn binary data into measurement results.
   - *Glossary:* terms you will see often in this text.
   - *Special Topics:* details you will need to know.
   - *Example Programs:* Appendix E contains two example programs. These programs, written in HP BASIC, contain examples of all the processes required to turn binary output data into measurement data.
2. You may want to set up the measurement in the HP 5373A using the HP-IB interface. See Chapters 4 and 5 of this manual for HP-IB information.

Writing the Program

A data capture and reduction program can be written with the following information. First, determine the exact output format for your data and how to process it to obtain measurement results. Next, determine how your program will interrogate the HP 5373A and store the data in your computer.

1. Find the number of your data format in *Table 3-1*. Be sure to read the short instructions for using *Table 3-1*. To use the table, you must have the following information about your measurement:
   - Expanded Data feature: ON or OFF.
   - Measurement Function (PRF, Frequency, PRI, Period, etc.)
   - Measurement Channel (Channel A, Channel A→ B, etc.)
   - Arming Mode (Automatic, Edge/Interval, etc.)
2. Turn to the format indicated in *Table 3-1*. There you will find the exact binary output format produced by your measurement. This is the key information required to write your binary data reduction program. Each format section will contain:
   - Output data structures and how they are related to the input signal.

- Specific instructions for generating time and event stamps.
- Formulas for computing measurement results from the time and event stamps.
- Information on how to process Expanded Data information.

3. Write program steps to:
   - Query the HP 5373A for data.
   - Read header string to recognize the beginning of the data stream and set up the proper amount of array space.
   - Transfer the binary measurement data to your computer.
   - Convert the binary data to a format which is suited for numerical manipulation of large, high-resolution numbers.
   - Process your data to obtain measurement results.

CHARACTERISTICS OF THE BINARY OUTPUT DATA

Data Output

Each binary output data transmission contains two types of information (see *Figure* 3-3):

- Header information (in ASCII code)
- Binary coded measurement data

If the measurement set-up on the HP 5373A calls for 1 block of 1000 measurements, then the data transmission will contain the data required to generate 1000 measurement results.

NOTE

Since the binary output is the HP-IB port, the data comes out in 8-bit bytes. The first byte is the most significant byte.

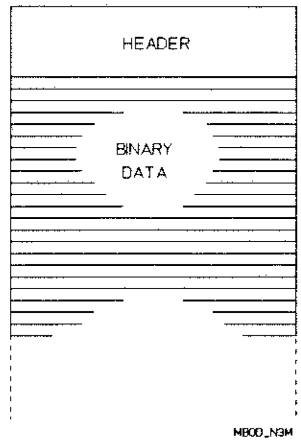


Figure 3-3. Binary Output Data

HEADER INFORMATION

The header is an 8 byte field which contains a measurement-start identifier (ASCII # and 6) and a six digit number which tells how many bytes of data are in the measurement field. *Figure 3-4* below shows the contents of the header. All header data is in ASCII code. All subsequent measurement data is in binary code.

EXAMPLE: One block of 500 PRF or frequency measurements:

500 PRF or frequency measurements contain 501 samples

Each sample contains the following:

event data	4 bytes
time data	4 bytes
interpolator data	2 bytes
Total	10 bytes

501 samples × 10 bytes/sample = 5010 bytes of data

The header format will be: #6005010

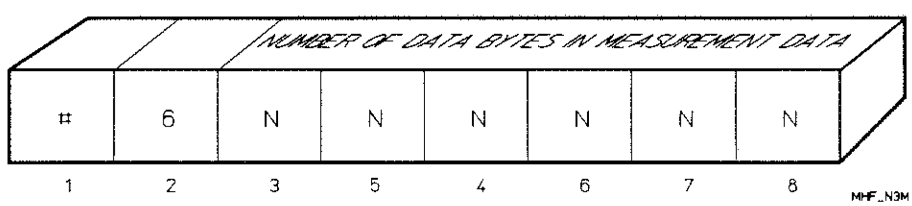


Figure 3-4. Header Format (ASCII code)

MEASUREMENT DATA

The format of the measurement data varies with the measurement function, input channel, arming mode, and whether Expanded Data is ON or OFF. The exact format for your data can be found in the *Binary Formats* section of this chapter. To find your format, determine your:

- Expanded Data status
- Measurement function (PRF, Frequency, PRI, Period, etc.)
- Measurement channel (Channel A, A→ B, etc.)
- Arming mode (Automatic, Edge/Event, etc.)

Next look in the Format Guide (*Table 3-1*) using these parameters to find your format number. When you turn to that format number, you will find the exact output data format produced by your measurement. You will also find detailed information on how to reduce that data to measurement results. The following paragraphs contain general information on the various output data fields.

Data Fields

The binary output data stream contains three types of data fields. These are event data, time data, and interpolator/status data. These terms are defined in the *Glossary* for this chapter, and discussed in some detail below.

EVENT DATA

Event data is a four byte (32-bit) unsigned binary number which is transmitted most significant byte first. Event data comes from the event counter in the HP 5373A, and represents the cumulative number of times the signal being measured has met the input trigger and arming requirements. Event data increases as the number of trigger events accumulate in the event counter. In most cases, the event data rolls over, or goes from maximum count to minimum count every 4,294,967,296 counts.

TIME DATA

Time data is also a four byte (32-bit) unsigned binary number which comes from the time counter. It represents the cumulative count of the HP 5373A 500 MHz time base. In most cases, whenever an event datum occurs you will get a corresponding time datum. The output of the time counter can be thought of as a clock which gives you the time of occurrence for each timed event. Time data increases monotonically and rolls over (goes from maximum count to minimum count) every 4,294,967,296 counts, or once about every 8.59 seconds.

INTERPOLATOR/ STATUS FIELD DATA

The 16-bit Interpolator/Status Field is shown in *Figure 3-5*. It provides space for interpolator data and four status bits. The 16-bit field is delivered most-significant-byte first during binary output.

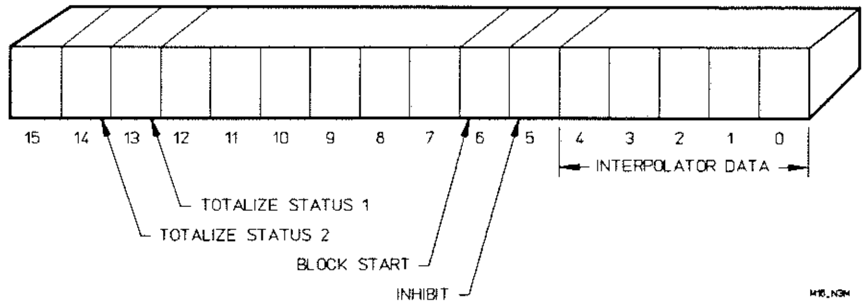


Figure 3-5. 16-bit Interpolator/Status Field

Interpolator Data

Interpolator Data is a 5-bit unsigned binary number. This number represents the number of 0.1 nanosecond intervals which have occurred between a timed event count, and the next 500 MHz clock pulse. Interpolator time data is computed by multiplying the interpolator output by 0.1 nanoseconds. The high resolution result is subtracted from its associated time datum to produce a time stamp. These instructions are repeated for each format. For more detail see paragraph entitled *Interpolator Data* under *Special Topics.*

STATUS BITS

Inhibit (Bit 5)

Bit 5 of the Interpolator/Status field indicates whether the Inhibit function was asserted prior to the current sample. If Inhibit was asserted, bit 5 is set to 1. With no Inhibit, bit 5 is 0. See *Measurement Inhibit* paragraph under *Special Topics* for a more complete description of the Inhibit function.

Block Start (Bit 6)

Bit 6 enables you to find the first sample in a block of data. The first sample in each block will have bit 6 (of the Interpolator/Status field) set to 1. In all subsequent samples, bit 6 will be 0. For more details, see *Block Start* paragraph under *Special Topics.*

Totalize Status 1 (TS1) and Totalize Status 2 (TS2)

TS1 and TS2 are only active when the HP 5373A is performing Totalize functions. The purpose of status bits TS1 and TS2 is to indicate which Totalize sample has valid event data. For more details see paragraph entitled *Totalize Status 1 And Totalize Status 2* under *Special Topics.* Each of the Totalize formats at the back of this chapter contain specific instructions for use of TS1 and TS2.

HOW TO PROCESS BINARY DATA

Binary data must be processed several ways before it can be used to compute measurement results. Each format description contains specific information on how to process the output data generated by that format. The following paragraphs contain general information designed to acquaint you with the various processes for computing time stamps and event stamps. Time and event stamps are used to compute the final measurement results.

Correct For Counter Rollover

The time and event counters have a finite capacity (4,294,967,296 counts in Normal (32-bit) mode, 65,536 counts in Fast (16-bit) mode). In many cases the counters will roll over (go from maximum count to zero) during a measurement block. This causes discontinuities in the data which prevents the computation of measurement results. To eliminate the discontinuities caused by counter rollover, your computer program must scan the data comparing each datum to its immediate predecessor. When a datum is found which is smaller than its predecessor, a rollover has occurred. The program must then add 4,294,967,296 (or 65,536 in Fast mode) to the smaller datum and to all subsequent data. The result of this process is that data will continuously increase throughout the measurement block. Some formats (such as those which describe Time Interval) will require a variation of this process, so be sure to read the information for your specific measurement format thoroughly. To turn time counter data into actual time stamps requires one or two further manipulations, these are described below.

Process Interpolator Data

Interpolator data is an unsigned 5-bit binary number which is part of the Interpolator/Status field (see paragraph above entitled *Interpolator/Status Field*). Every time datum has an associated interpolator datum. The interpolator data represents the number of 100 ps intervals which have elapsed between a timed input event and the next clock pulse. To decode the interpolator data, the 5 bits are extracted from the Interpolator/Status field and their numerical value is multiplied by 100 ps. The resulting number will be subtracted from its associated time datum resulting in a high resolution time stamp.

Determine Offset

When making any measurement with start and stop inputs, a correction is needed to compensate for the differences in the electrical path length between measurement channels. The electrical path length is changed by such things as Attenuation and Separate/Common channel connection. *Technical Note 4* describes how to compute the offset. Each format section contains information on whether or not an offset correction is required, and how to apply the correction to the data.

Compute Time Stamps

A time stamp is the result of correcting the binary time count data for counter rollover, subtracting interpolator data and, in some cases, adding the offset information. The time stamp, in other words, represents the true time of occurrence of an event in a block of measurements, and is used to compute measurement results. The formula for computing time stamps is given in each format and repeated below:

$$\textbf{time stamp} = (\text{time datum}) \times 2\ \text{ns} - (\text{interpolator datum}) \times 0.1\ \text{ns} + (\text{offset correction, if required})$$

NOTE

In some formats the offset correction is subtracted from rather than added to the time stamp data. Check your specific format for proper handling of offset data.

Compute Event Stamps

Event stamps are derived from the event count data by correcting for counter rollovers. With this correction, the event data can be used to compute measurement results. Each format section describes the process for converting event count data to event stamps.

Compute Measurement Results

Measurement results are computed directly from the time and event stamps. Each format contains the formulas required to make your measurement computations. The most widely used formulas are reproduced below for reference:

$$\textbf{PRF or frequency}_i = \frac{\text{event stamp}_{i+1} - \text{event stamp}_i}{\text{time stamp}_{i+1} - \text{time stamp}_i}$$

$$\textbf{Time Interval}_i = \text{stop time stamp}_i - \text{start time stamp}_i$$

$$\textbf{Totalize result}_i = \text{stop event stamp}_i - \text{start event stamp}_i$$

Special Topics

SPECIAL TOPICS

Introduction

The purpose of this section is twofold. First, it provides additional information on features of the HP 5373A which are not covered in the general or *Binary Format* sections of this chapter. The purpose here is to give you additional insight into the capabilities of the HP 5373A. Secondly, Special Topics gives more in-depth information on certain points only covered briefly elsewhere (to make the material easier to understand on initial reading). Where appropriate, references to *Special Topics* are made from both the general information and *Binary Format* sections of this chapter.

Measurement Inhibit

The Measurement Inhibit feature allows the user to selectively suppress data collection by the HP 5373A. When Inhibit is asserted during measurements, measurement activity stops and restarts only when Inhibit is de-asserted.

EFFECTS OF INHIBIT ON OTHER INSTRUMENT FUNCTIONS

Arming

Inhibit does not interfere with arming or capture of block arming data. Blocks and samples are armed normally when Inhibit is active. Time data generated when a block is armed will still be collected, even if a block is armed while Inhibit is asserted. Measurement data, on the other hand, will not be taken when Inhibit is asserted.

Pre-trigger

Inhibit does not stop detection of a Pre-trigger edge. However, if the Pre-trigger edge is to be followed by measurements, no measurements will be made until Inhibit is de-asserted.

Inhibit will prevent detection of Pre-trigger if the HP 5373A is set to Pre-trigger on a specified Time Interval. Inhibit prevents the 5373A from monitoring the data stream for the Pre-trigger criteria. The Pre-trigger function will return to normal when Inhibit is de-asserted.

Measurements Requiring More Than One Sample

Assertion of the Inhibit function will not cause incomplete two-sample measurements. The HP 5373A responds to Inhibit by discarding measurements in which either start sample, stop sample, or both have been Inhibited. Thus, measurements requiring a start and stop sample (such as Time Interval), or measurements requiring first and second attempts at data capture (such as Totalize), will always be delivered to the binary output in pairs, as expected.

Inhibit Status Bit

The Interpolator/Status Field (see paragraph entitled Interpolator/Status Field which follows) contains information regarding the Inhibit function. Bit 5 in this field is the Inhibit status bit. If bit 5 is 1, the current measurement sample was preceded by Inhibit. Otherwise bit 5 is 0.

Measurement Results

When activated, Inhibit extends the elapsed time between samples. Because the HP 5373A services Inhibit selectively, many measurement results will not be affected. There are two ways Inhibit will affect measurements:

- During measurements which have both a start and a stop signal, Inhibit extends the interval between measurements. It does not affect the time between start and stop.
- For all other measurements, asserting Inhibit affects data by extending the measurement gate time.

Time Stamps Associated With Block Arming Edge

Certain measurements produce not only standard measurement data, but arming information as well. For these measurements, the trigger event which arms the block produces the first time datum in each block. When this occurs, there will be no corresponding event datum.

Measurements which produce this type of data are single channel PRF, Frequency, PRI, Period, Continuous Time Interval, Phase Deviation, Frequency Deviation, or Time Deviation measurements armed in the following manner:

- Edge Holdoff
- Event Holdoff
- Time Holdoff
- Edge/Cycle
- Edge/Edge
- Edge/Interval
- Event/Interval
- Time/Interval

Your data reduction program must identify the block arming sample and process it as an exception. To identify an arming sample, examine bit 6 in the Interpolator/Status field. Bit 6 is always 1 in the first sample of each block. For the single channel measurements and the arming modes given above, the first sample in the block will be the block arming sample. In all subsequent samples, bit 6 will be 0.

When your program encounters arming data, it should consider the special characteristics of arming sample data:

- The measurement sample will include an UNUSED event data field if your binary output is Format 2B or Format 3.
- The block arming signal uses a different electrical path than the other signals. Thus, an offset correction is required to compensate for the differential path length. See *Technical Note 4*, at the end of the format section, for offset corrections.

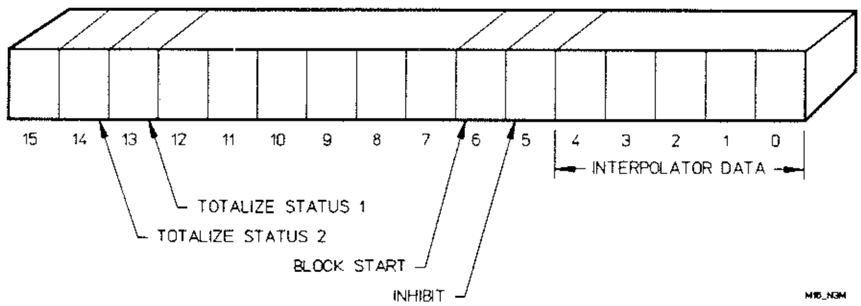


Figure 3-6. 16-bit Interpolator/Status Field

Interpolator/Status Field

The 16-bit Interpolator/Status Field is shown in *Figure 3-6*. This field provides space for interpolator data as well as four status bits, and is delivered most-significant-byte first during binary output.

INTERPOLATOR DATA

The five interpolator bits are numbered 4 through 0. These bits represent a binary number from 0 to 18. Bit 4 is the most significant bit.

The 5-bit interpolator datum is used to improve resolution on its associated time datum. The interpolator datum represents an even number of 100 picosecond intervals (bit 0 is always 0). The Interpolator datum is multiplied by 100 ps to obtain a correction value, then subtracted from its associated time datum to produce a time stamp. For example, if the binary Interpolator datum is 00110, 0.6 ns must be subtracted from the associated time datum to form the time stamp.

STATUS BITS

Inhibit

Bit 5 indicates whether Inhibit was asserted prior to the current sample. If Inhibit was asserted, and at least one sample was discarded as a consequence, the Inhibit bit is set to 1. With no Inhibit, bit 5 is 0.

Block Start

Bit 6, when set to 1, indicates that the current sample is the first sample in the block; otherwise bit 6 is 0. Bit 6 information is particularly useful with multiple-block acquisitions, or multiple-block Pre-triggered acquisitions. During these types of acquisitions, samples which initiate blocks may be embedded in the output data.

Totalize Status 1, Totalize Status 2

The Totalize function requires that data be captured by making two attempts at each sample. One attempt will be valid data. The other will not. Bit 13, Totalize Status 1 (TS1), and bit 14, Totalize Status 2 (TS2) are used only for Totalize measurements. Bit 13 is used in Totalize formats 11, 13, 14, and 15. Bit 14 is used in formats 12, 13, and 15. These bits indicate which sample contains a valid event count. If the bit is 1, the second attempt at data capture represents the valid event datum. If the bit is 0, the first attempt at data capture is valid. Use of bit 13 and/or 14 is specified, where necessary, for each binary format at the back of this chapter.

Fast Mode

The HP 5373A has been designed to include Fast mode, a data transmission mode tailored to users who need very high speed binary data delivery, along with a faster sample interval (75 ns).

In Fast mode, the HP 5373A collects fewer bits of data with each sample, but collects samples much more quickly. Specifically, time and event fields in Fast mode contain 16 bits of data. In Normal mode, these fields contain 32 bits. The size of the Interpolator/Status Field is unchanged (16 bits).

The use of 16 bits to count events limits the HP 5373A to to a maximum count of 65,536 (2^{16}) events prior to counter rollover. Thus, to be unambiguously interpreted, there must be no more than 65,536 input events between samples. The use of 16 bits to count time data allows the HP 5373A to count for a maximum time of 131.072 μs prior to counter rollover. Thus, for time data to be unambiguously interpreted, incoming signals must be sampled at least once every 131.072 μs. Further restrictions are imposed on Formats 4B, 5B, 10B, and 10D. For these formats, elapsed time must not exceed 65 μs. If samples occur at a slower rate, you will not be able to compute correct measurement results.

Pre-trigger

A thorough introduction to the Pre-trigger feature is provided in Chapter 10 of the HP 5373A Operating Manual. We suggest you read that material so you may more easily relate the information below to the task of reducing binary output data.

Important characteristics of Pre-triggered acquisitions are:

- The instrument monitors the data stream for a Pre-trigger event. During the monitoring phase, samples are stored to memory, but are not output to the binary Bus.
- When the Pre-trigger event is detected, the "capture window" is positioned about the Pre-trigger event. Samples falling within the capture window are recovered from memory and delivered as binary output data.
- The capture window is wide enough to recover at least 1 block's worth of samples. For example:

  Measurement: PRF or frequency, Channel A
  Arming: Automatic
  Single block of 10 measurements
  The capture window is 11 samples wide.

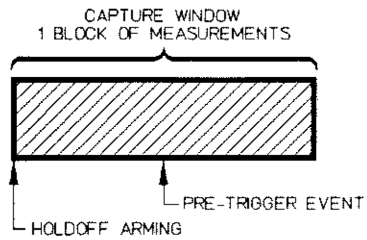
- If one Pre-trigger event occurs before the instrument has collected enough samples to fill the capture window, then all samples collected are delivered as binary data. The number collected will be less than enough to generate a full block of results.
- With Pre-trigger ON, the user may select between a single block of N measurements or a multiple block of N measurements.

The important distinctions between these two modes are discussed below.

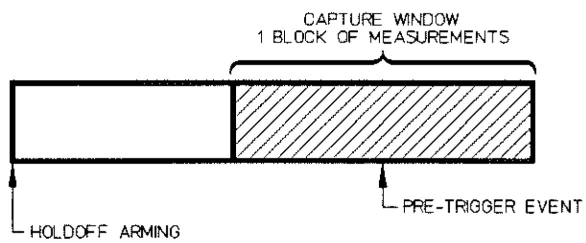
SINGLE BLOCK PRE-TRIGGERED ACQUISITION

If single block is selected, the instrument satisfies the holdoff condition once. The corresponding binary output data takes one of two forms.

1. The sample which initiated the block of measurements is the first sample in the binary output data. If your arming choice produces a timestamp for the block arming edge, it will be important to identify this first sample. See Key Facts paragraph which follows for more information.



2. The sample which initiated the block of measurements is no longer in the capture window. The first sample in the binary output data is valid data AFTER the block-initiation sample. See figure below.



MULTIPLE BLOCK PRE-TRIGGERED ACQUISITIONS

If Multiple blocks are selected, the instrument repeatedly executes the 1-block cycle shown below until the Pre-trigger is detected.

1. Satisfy block holdoff condition.

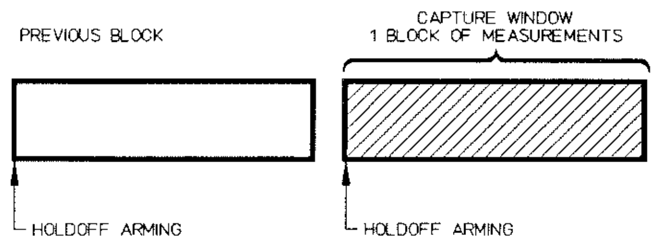
2. Enable the hardware which detects the Pre-trigger.

3. Collect the N measurements in the block.

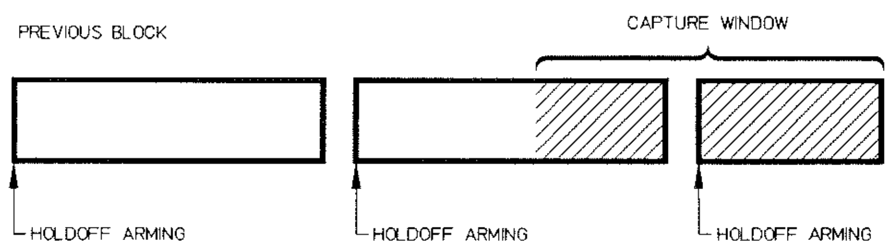
4. Disable the hardware which detects the Pre-trigger. (Note that a Pre-trigger edge which occurs between blocks cannot be detected.)

When the Pre-trigger event is detected, the instrument positions the capture window about the Pre-trigger as previously described. For the Multiple block acquisition, the capture window is likely to contain the sample which initiated a block, but the position of this sample within the capture window depends on the input data. There are three possibilities described below. You must write code which detects a block initiation sample in any of the positions described below. See Key Facts paragraph below for more details.

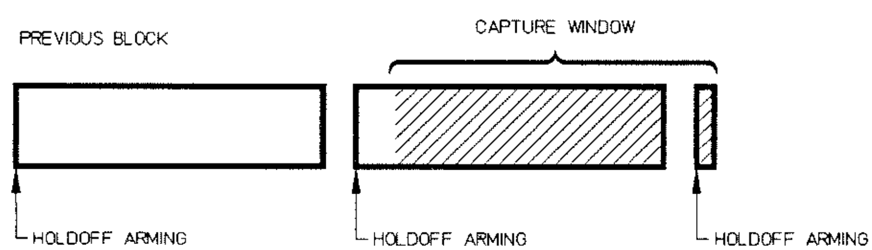
1. The sample which initiated a block of measurements is the first sample in the Binary Output Data.



2. The sample which initiates a block of measurements is embedded in the Binary Output Data, (is not the first sample).



3. The sample which initiates a block of measurements is the last sample in the Binary Output Data.



PROCESSING PRE-TRIGGER BINARY OUTPUT DATA

Key Facts

From the above you can see the Block Arming Sample may occur anywhere in the capture window (but always at the beginning of a block). The Block Arming Sample contains a time datum which is generated by the Block Arming condition, and thus is not necessarily correlated with the signal being measured. Further, the unused field produced in the same sample as the Block Arm time data may contain invalid data. Thus your program must recognize the Block Arm time datum, and its associated event datum (by use of status bit 6), and distinguish this data from the actual measurement data.

The following set of rules summarizes the use of Block Arming and inter-block data when making Pre-trigger measurements.

1. Identify Block Arming data using Bit 6 of the Interpolator/ Status field. Note that bit 6 of the Interpolator/Status Field is 1 when the corresponding sample initiates a block. Use this bit to determine the Block Arming time data. See paragraph entitled Time Stamps Associated with the Block Arming Edge in this Special Topics section for more details.

2. Use the Block Arming time datum the same as other time data to detect counter rollovers. Once the time data has been corrected for rollovers, set aside the Block Arming time data. It cannot be used to compute measurement results.

3. Discard data in the unused field associated with Block Arming time data. This data cannot be used to compute measurement results.

4. Do not compute measurement results using adjacent end data from different blocks (i.e. last sample from previous block and first sample from current block). Compute all measurement results using only data from the same block. Besides the Block Arming data problem, event counters are reset between blocks in some arming modes. These arming modes are: Time Holdoff, Event Holdoff, Edge/Cycle, Time/Interval, Event/Interval, Time Sampling, Edge/Time, Edge/Event, Time/Time, Event/Event. When event counters are reset between blocks, event data in adjacent blocks will be uncorrelated.

GLOSSARY

Counter Rollover/ Counter Overflow: This occurs when a time or event counter within the HP 5373A goes from maximum count to zero during a measurement. Because the HP 5373A acquires data continuously, any counter rollover must be detected and corrected before computing final measurement data.

Event Data: Data generated from the event counters in the HP 5373A. In general, event data represents the number of trigger events which occur in the input channel. Specifically, when measuring PRF or frequency, event data represents cycles of the incoming signal; in Time Interval measurements, an event count (as well as a corresponding time count) results from properly armed start and stop input signals. In the Binary output, each event datum is represented by an unsigned 32-bit binary number (16-bit number in Fast mode). Event data must be corrected for counter rollovers by the user.

Event Stamp: A computed value based on event data. To compute an event stamp, event data is first checked for counter rollover, and a correction applied if counter rollover is detected. The corrected event data is called an event stamp.

Expanded Data: Expanded Data has two meanings. First, it is an operating mode for the HP 5373A. Expanded Data mode tells the Binary output to include all available data from a measurement in its output. For example, when the HP 5373A makes a Time Interval measurement, and it is NOT in the Expanded Data mode, only the time data required to compute Time Interval results is available at the binary output (and the data is output at a much higher rate). When making the same measurement in Expanded Data mode, the binary output also includes event data. Event data is used to compute the Expanded Data "Missed Events". This leads to the second meaning of Expanded Data: *the set of extra results which can be calculated in the Expanded Data mode.* In the example, Expanded Data for Time Interval measurements is Missed Events. Missed Events can be computed from the extra event data which is available in the Expanded Data mode. The format description for your measurement will tell you the contents of the Expanded Data mode output, and the required algorithm for computing the Expanded Data results.

Inhibit: When enabled, the Inhibit function allows you to start or stop HP 5373A measurement activity with a signal applied to the rear panel **Inhibit Input** connector. Refer to the *Pre-trigger*

Menu chapter in the Operating Manual and the *Inhibit* paragraphs in the *Special Topics* section of this chapter for more details.

Interpolator — Circuits within the HP 5373A which measure the precise time between the occurrence of an event and the next 2 ns clock pulse. The interpolator has a resolution of 200 ps. Information from the interpolator is used to increase measurement resolution when computing time stamps.

Interpolator Data — Binary data output from the interpolator circuits. Each interpolator datum is an unsigned 5-bit binary number. See the *Interpolator Data* paragraph in the *Special Topics* section of this chapter for details.

Minus Time-Interval — A Time Interval measurement made when the stop signal is received before the start signal. When this occurs, special processing of the time and event data is required. See *Technical Note 3* for details.

Missed Events — When events occur at a rate greater than can be time stamped by the HP 5373A, the instrument will not generate time data for each event (although each event will be counted). These events which do not have associated time data, are called Missed Events. Data sufficient to calculate the number of Missed Events is available when Expanded Data is ON. Each applicable format contains information on how to calculate Missed Events.

Sample — Usually an event datum with its corresponding time datum. Samples are the building blocks from which measurement data (PRF, Frequency, PRI, Period, Time Interval, etc.) is constructed. In certain cases, samples will not have an associated event datum. This occurs in some Time Interval measurements or when a block is armed. All exceptions are covered in the *Binary Formats* section of this chapter.

Start Sample — The sample which results from a trigger event caused by an input start signal, after the HP 5373A is armed.

Stop Sample — The sample which results from a trigger event caused by an input stop signal, after the HP 5373A is armed.

Time Data

Data generated from the time counters in the HP 5373A. The time counters continuously accumulate counts from the 500 MHz system clock. The accumulated time count for each timed event is sent to the HP-IB output as time data, an unsigned 32 bit binary number (16 bit number in Fast mode). Time data has a resolution of 2 ns.

Time Stamp

The expression time stamp can be used as a noun or a verb. As a noun, a time stamp is the precise time of occurrence of each event. Each time stamp represents an elapsed time (see definition of time data above). When using the binary output, time stamps are computed external to the HP 5373A. The computation uses time data, interpolator data, and internal signal path data. The basic time stamp is computed by correcting the time data for any counter rollovers, multiplying the corrected time data by 2 ns, then subtracting the interpolator data. Interpolator data gives the time stamp a resolution of 200 ps. Final corrections are made to a time stamp (if required) for differences in internal signal path length. The process for computing time stamps is described in detail in the *Binary Formats* section of this chapter.

As a verb, time stamp means that a given arming signal followed by an event, or an arming signal alone, has caused corresponding time and interpolator data to be generated. For example, when we say an event has been time stamped, it means that the time of occurrence of that event has been captured with both time and interpolator data.

Time Stamp the Block Arm

This feature, which makes Block Averaging possible, enables the Binary Output to produce time and interpolator data corresponding to the precise time at which a block was armed. This feature is automatically enabled under several arming conditions when making PRF, Frequency, PRI, Period, Continuous Time Interval, Phase Deviation, Time Deviation, and Frequency Deviation measurements. See *Block Averaging* in *Function Menu* chapter of the HP 5373A Operating manual for more details.

Trigger Event

A trigger event occurs when an input signal meets the slope and voltage criteria set on the Input menu. Trigger events are accumulated in the event counter.

Binary Formats

HOW TO USE TABLE 3-1, FORMAT GUIDE

Table 3-1 is your map to the Format tables which follow. To find the format which describes your particular measurement, you must have the following information about your measurement:

- Expanded Data function ON or OFF
- Measurement function
- Input channel(s) to be used
- Arming mode

Table 3-1 is in two parts. One part describes formats for the Expanded Data ON function, the other covers Expanded Data OFF. The table titles tell which function is described in each table. To determine your format:

1. Select the table which corresponds to the Expanded Data function you plan to use.
2. Choose the measurement function and channel across the top of the table. Mark the column.
3. Choose the arming mode row from the left side of the chart and mark the row.
4. Locate the output data format for your measurement at the row/column intersection.

For example:

Expanded Data: ON
Measurement function: PRF or Frequency
Input channel: A
Arming mode: Edge Sampling

The output format for this measurement is 2A.

Table 3-1(A). Format Guide — EXPANDED DATA ON

ARMING MODE	MEASUREMENT FUNCTION												
	TIME INTERVAL		CONTINUOUS TIME INTERVAL	±TIME INTERVAL		PRF, FREQUENCY, PRI, PERIOD		TOTALIZE		PULSE WIDTH, OFFTIME, RISE TIME FALL TIME DUTY CYCLE	PHASE	ENVELOPE POWER, AMPLITUDE MODULATION	PHASE DEVIATION, TIME DEVIATION, FREQUENCY DEVIATION
	A	A → B	A	A	A → B	A	DUAL[1]	A	DUAL[1]	A	A rel B	A	A
	B	B → A	B	B	B → A	B	RATIO[2]	B	RATIO[2]		B rel A	B	B
						C	SUM[3]		SUM[3]				
							DIFF[4]		DIFF[4]				
AUTOMATIC													
AUTOMATIC	5A	10A	2A		10B	2A	10C			10B	10B	2A	2A
HOLDOFF													
EDGE HOLDOFF	5A	10A	2B		10B	2B					10B		2B
TIME HOLDOFF	8	6	3			3							
EVENT HOLDOFF	8	6	3			3							
SAMPLING													
INTERVAL SAMPLING	5A	10A	2A		10B	2A	10C	11	13		10B		2A
TIME SAMPLING						5A							
CYCLE SAMPLING						2A							
EDGE SAMPLING						2A	10C	11	13				
PARITY SAMPLING					10B								
REPET EDGE SAMPLING	5A	10A	2A		10B								
REPET EDGE-PARITY SAMPLING					10B								
RANDOM SAMPLING	5A	10A			10B								
HOLDOFF/SAMPLING													
EDGE/INTERVAL	5A	10A	2B		10B	2B	10C	11	13		10B		2B
EDGE/TIME						5A							
EDGE/EDGE						2B		11	13				
EDGE/CYCLE						2B							
EDGE/EVENT				5B	4B	5A							
EDGE/PARITY					10B								
EDGE/RANDOM	5A	10A			10B								
TIME/INTERVAL						3		12					
TIME/TIME				4B	4B	10D							
EVENT/INTERVAL						3							
EVENT/EVENT				4B	4B	10D							
EXTERNALLY GATED						5A		14	15				
MANUAL								14	15				

1. DUAL. Simultaneous Dual-channel, (2 results). Frequency and Period options are: A&B, A&C, B&C. Totalize option is: A&B.
2. RATIO. Frequency and Period ratio options are: A/B, A/C, B/A, B/C, C/A, C/B. Totalize ratio options are: A/B, B/A.
3. SUM. Frequency and Period sum options are: A+B, A+C, B+C. Totalize sum option is: A+B.
4. DIFFERENCE. Frequency and Period difference options are: A–B, A–C, B–A, B–C, C–A, C–B. Totalize difference options are: A–B, B–A.

ARMING CATEGORIES

Category	Continuous Arming Modes	Non-Continuous Arming Modes
Automatic	Block Holdoff is Automatic Sample Arm is Automatic	none
Holdoff Modes	Block Holdoff is User-defined Sample Arm is Automatic	none
Sampling Modes	Block Holdoff is Automatic Sample Arm is User-defined	Start Arm is Automatic Stop Arm is User-defined
Holdoff/Sampling Modes	Block Holdoff is User-defined Sample Arm is User-defined	Start Arm is User-defined Stop Arm is User-defined

Table 3-1(B). Format Guide — EXPANDED DATA OFF

ARMING MODE	MEASUREMENT FUNCTION												
	TIME INTERVAL		CONTINUOUS TIME INTERVAL	±TIME INTERVAL		PRF, FREQUENCY, PRI, PERIOD		TOTALIZE		PULSE WIDTH OFFTIME, RISE TIME, FALL TIME, DUTY CYCLE	PHASE	ENVELOPE POWER, AMPLITUDE MODULATION	PHASE DEVIATION, TIME DEVIATION, FREQUENCY DEVIATION
	A	A→B	A	A	A→B	A	DUAL[1]	A	DUAL[1]	A	A rel B	A	A
	B	B→A	B	B	B→A	B	RATIO[2]	B	RATIO[2]		B rel A	B	B
						C	SUM[3]		SUM[3]				
							DIFF[4]		DIFF[4]				
AUTOMATIC													
AUTOMATIC	4A	4A	1A		4B	2A	10C			10B	10B	2A	2A
HOLDOFF													
EDGE HOLDOFF	4A	4A	1B		4B	2B					10B		2B
TIME HOLDOFF	4A	4A	1B			3							
EVENT HOLDOFF	4A	4A	1B			3							
SAMPLING													
INTERVAL SAMPLING	4A	4A	1A		4B	2A	10C	11	13		10B		2A
TIME SAMPLING						5A							
CYCLE SAMPLING						2A							
EDGE SAMPLING						2A	10C	11	13				
PARITY SAMPLING					4B								
REPET EDGE SAMPLING	4A	4A	1A		4B								
REPET EDGE-PARITY SAMPLING					4B								
RANDOM SAMPLING	4A	4A			4B								
HOLDOFF/SAMPLING													
EDGE/INTERVAL	4A	4A	1B		4B	2B	10C	11	13		10B		2B
EDGE/TIME						5A							
EDGE/EDGE						2B		11	13				
EDGE/CYCLE						2B							
EDGE/EVENT				4B	4B	5A							
EDGE/PARITY					4B								
EDGE/RANDOM	4A	4A			4B								
TIME/INTERVAL						3		12					
TIME/TIME				4B	4B	10D							
EVENT/INTERVAL						3							
EVENT/EVENT				4B	4B	10D							
EXTERNALLY GATED						5A		14	15				
MANUAL								14	15				

1. DUAL. Simultaneous Dual-channel, (2 results). Frequency and Period options are: A&B, A&C, B&C. Totalize option is: A&B.
2. RATIO. Frequency and Period ratio options are: A/B, A/C, B/A, B/C, C/A, C/B. Totalize ratio options are: A/B, B/A.
3. SUM. Frequency and Period sum options are: A+B, A+C, B+C. Totalize sum option is: A+B.
4. DIFFERENCE. Frequency and Period difference options are: A–B, A–C, B–A, B–C, C–A, C–B. Totalize difference options are: A–B, B–A.

ARMING CATEGORIES

Category	Continuous Arming Modes	Non-Continuous Arming Modes
Automatic	Block Holdoff is Automatic Sample Arm is Automatic	none
Holdoff Modes	Block Holdoff is User-defined Sample Arm is Automatic	none
Sampling Modes	Block Holdoff is Automatic Sample Arm is User-defined	Start Arm is Automatic Stop Arm is User-defined
Holdoff/Sampling Modes	Block Holdoff is User-defined Sample Arm is User-defined	Start Arm is User-defined Stop Arm is User-defined

FORMAT 1A

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Continuous Time Interval	A, B	OFF	Automatic Interval Sampling Repetitive Edge Sampling

How Signals Are Converted to Binary Data

Example: Continuous Time Interval | Automatic

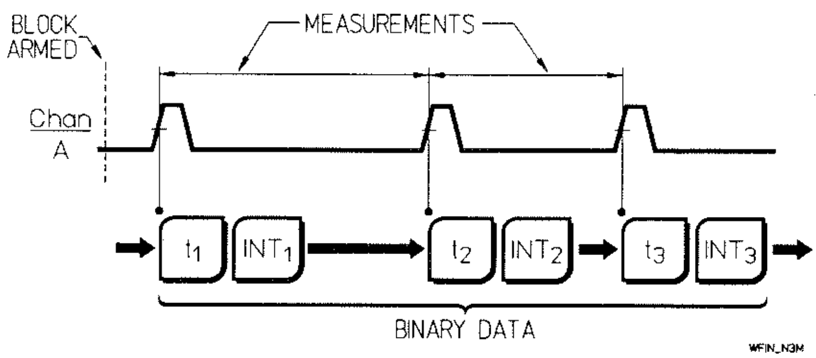


Figure 1A-A. Continuous Time Interval Measurement on Channel A With Automatic Arming Mode Showing Corresponding Binary Data Output

Producing Results

Continuous Time Interval results are generated by calculating elapsed time between contiguous samples.

NOTE

With Expanded Data OFF, the count of elapsed trigger events is unavailable.

Binary Data Output

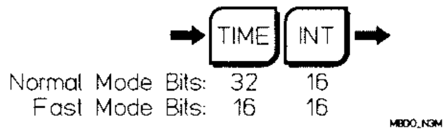


Figure 1A-B. Format 1A Binary Output

Format 1A binary output is shown in *Figure 1A-B* above. Each sample generates the binary data defined below.

TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid Status Bits: Block Start, Inhibit.

Converting Binary Data to Time Stamps

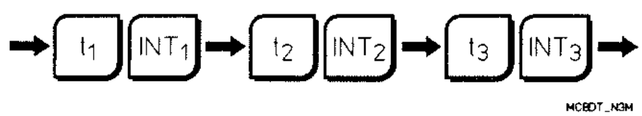


Figure 1A-C. Format 1A Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 1A-A*. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.

2. Incorporate interpolator data.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Generating Final Function Results from Time Stamps

Continuous Time Interval results are generated by calculating elapsed time between contiguous time stamps. For each contiguous pair:

Continuous Time Interval$_i$ = time stamp$_{i+1}$ – time stamp$_i$

FORMAT 1A

FORMAT 1B

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Continuous Time Interval	A, B	OFF	Edge Holdoff Time Holdoff Event Holdoff Edge/Interval

How Signals Are Converted to Binary Data

Example: Continuous Time Interval B | Edge Holdoff

Producing Results

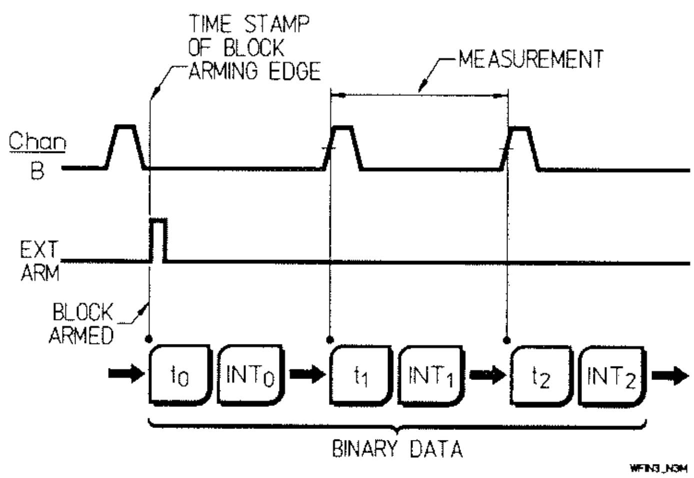


Figure 1B-A. Continuous Time Interval Measurement on Channel B with Edge Holdoff Arming Mode Showing Corresponding Binary Data Output.

In this format, the first sample in each block is unique: it captures time of occurrence for the block arming edge.

Continuous Time Interval results are generated by calculating elapsed time between contiguous samples.

Binary Data Output

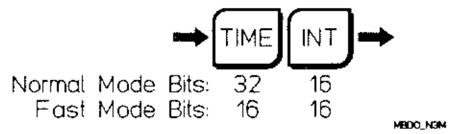


Figure 1B-B. Format 1B Binary Output

Format 1B binary output is shown in *Figure 1B-B* above. Each sample generates the binary data defined below.

TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid Status Bits: Block Start, Inhibit.

Converting Binary Data to Time Stamps

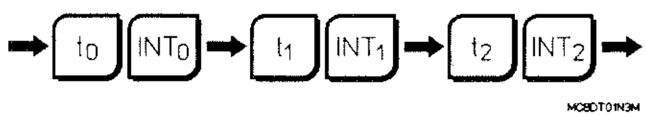


Figure 1B-C. Format 1B Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 1B-A*. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between block arming channel and signal channel.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. The first sample in each block is not an exception to this rule. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Block Arming Channel and Signal Channel

There are several possible signal paths running from the signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, subtract a correction value, called an offset, from the time stamp associated with the first sample in each block. Subsequent time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the first sample in each block:

corrected first time stamp = time stamp – offset

Generating Final Function Results from Time Stamps and Event Stamps

NOTE

The Block Start Status Bit is bit 6 of the interpolator/status field. Block Start bit 6 is 1 when the sample is the first sample in a block. When this occurs, the corresponding time datum contains block arming data and is not part of the measurement. See the paragraph entitled Interpolator/Status Fields *in the* Special Topics *section for more information.*

CONTINUOUS TIME INTERVAL RESULTS

Continuous Time Interval results are generated by calculating the elapsed time between contiguous time stamps. The elapsed time between the first and second time stamp in each block represents elapsed time between the block arming edge and the first signal edge. Because of this, the elapsed time between the first and second sample is not part of the Continuous Time Interval measurement data. See the paragraphs entitled *Time Stamps Associated With Block Arming Edge* in the *Special Topics* section of this chapter for more information.

FORMAT 1B

For the first sample in each block:

Block arming data = first measurement channel time stamp – block arming time stamp

For subsequent samples:

Continuous Time Interval$_i$ = time stamp$_{i+1}$ – time stamp$_i$

FORMAT 2A

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
PRF, Frequency, PRI, Period	A, B, C	ON/OFF	Automatic Interval Sampling Cycle Sampling Edge Sampling
Continuous Time Interval	A, B	ON	Automatic Interval Sampling Repetitive Edge Sampling
Phase Deviation	A, B	ON/OFF	Automatic Interval Sampling
Time Deviation	A, B	ON/OFF	Automatic Interval Sampling
Frequency Deviation	A, B	ON/OFF	Automatic Interval Sampling

How Signals Are Converted to Binary Data

Example: Frequency A | Edge Sampling

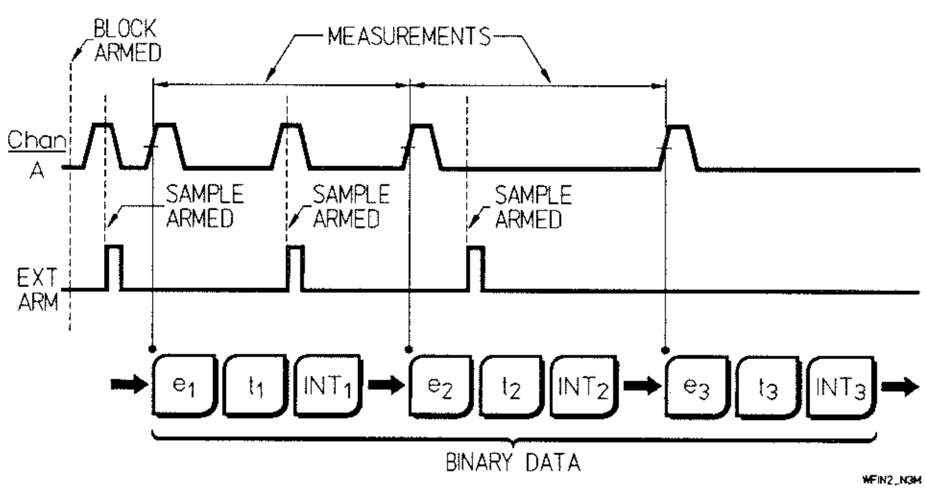


Figure 2A-A. Frequency Measurement on Channel A with Edge Sampling Arming Mode Showing Corresponding Binary Data Output.

Producing Results

Continuous Time Interval results are generated by calculating elapsed time between contiguous samples.

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient:

$$\text{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\text{PRI or Period} = \frac{\text{elapsed time}}{\text{elapsed events}}$$

Phase Deviation, Time Deviation, and Frequency Deviation results are generated by use of a special algorithm. The algorithm and an explanatory diagram are included at the end of this format section.

EXPANDED DATA

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are trigger events which fall between samples, but which are not time stamped. See the *Glossary* in this chapter for more details. The formula for calculating Missed Events is:

Missed Events = [elapsed events between samples] – 1

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Binary Data Output

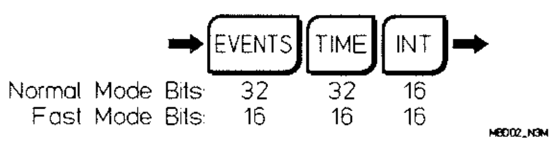


Figure 2A-B. Format 2A Binary Output

Format 2A binary output is shown in *Figure 2A-B* above. Each sample generates the binary data defined below.

EVENT: binary event count.
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

Figure 2A-C. Format 2A Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 2A-A*. To produce valid event stamps, your program must:

1. Correct the binary event data for counter rollovers.
2. If the measurement was made on Channel C, multiply the event stamp data by four.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

Successive event counts increase monotonically unless a counter overflow has occurred. An overflow-processing routine must scan the event data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and all subsequent data, then continue with the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

Channel C Correction

When making measurements on Channel C, the event counter sees only one out of every four input events. To correct for this, multiply event data by four. This must be done after correcting for counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Generating Final Function Results from Time Stamps and Event Stamps

CONTINUOUS TIME INTERVAL RESULTS

Continuous Time Interval results are generated by calculating elapsed time between contiguous time stamps. For each contiguous pair:

$$\textbf{Continuous Time Interval}_i = \text{time stamp}_{i+1} - \text{time stamp}_i$$

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between samples, but are not time stamped. The formula for calculating Missed Events is:

$$\textbf{Missed Events}_i = [\text{event stamp}_{i+1} - \text{event stamp}_i] - 1$$

PRF, FREQUENCY, PRI, AND PERIOD RESULTS

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient.

$$\textbf{PRF}_i \textbf{ or Frequency}_i = \frac{\text{event stamp}_{i+1} - \text{time stamp}_i}{\text{time stamp}_{i+1} - \text{event stamp}_i}$$

$$\textbf{PRI}_i \textbf{ or Period}_i = \frac{\text{time stamp}_{i+1} - \text{time stamp}_i}{\text{event stamp}_{i+1} - \text{event stamp}_i}$$

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating the elapsed time between samples.

$$\textbf{Measurement Gate Time}_i = \text{time stamp}_{i+1} - \text{time stamp}_i$$

TIME DEVIATION, PHASE DEVIATION, AND FREQUENCY DEVIATION RESULTS

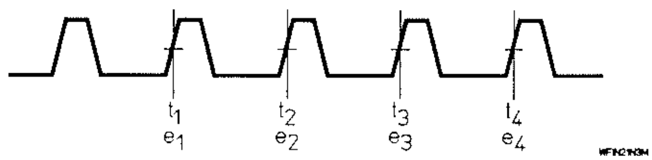


Figure 2A-D. Time Deviation and Phase Deviation Example Signal showing Time and Event Stamp Measurement Points.

$$\text{Time Deviation}_i = (e_{i+1} - e_1) \times P - (t_{i+1} - t_1)$$

$$\text{Phase Deviation}_i = \left[(e_{i+1} - e_1) \times P - (t_{i+1} - t_1)\right] \times \frac{360°}{P}$$

$$\text{Frequency Deviation}_i = \frac{(e_{i+1} - e_i)}{(t_{i+1} - t_i)} - f$$

NOTES:

1. P = Period of reference signal; f = frequency of reference signal (i.e., f = 1/P) — The signal used as the reference (carrier) is identified on the Math menu. For a Phase Deviation, Time Deviation, or Frequency Deviation measurement, a carrier can be any one of:

   - an estimated pulse, CW carrier, or linear carrier specified manually (by entering the slope and starting point), or
   - a manually entered carrier frequency value, or
   - the automatically calculated mean of a block of measurements. If the automatic carrier frequency is calculated in the Pulse AUTO mode, any measurement that includes off-time data is not used in the calculation.

   For more information about carrier frequency determination, see Section 5 of this manual.

2. Sample t_1 is used as a reference edge; the first result is generated by manipulation of t_2.

3. The result is negative if the signal lags the reference.

FORMAT 2B

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Continuous Time Interval	A, B	ON	Edge Holdoff Edge/Interval
PRF, Frequency, PRI, Period	A, B,C	ON/OFF	Edge Holdoff Edge/Interval Edge/Edge Edge/Cycle
Phase Deviation	A, B	ON/OFF	Edge Holdoff Edge/Interval
Time Deviation	A, B	ON/OFF	Edge Holdoff Edge/Interval
Frequency Deviation	A, B	ON/OFF	Edge Holdoff Edge/Interval

How Signals Are Converted to Binary Data

Example: Frequency B | Edge Holdoff

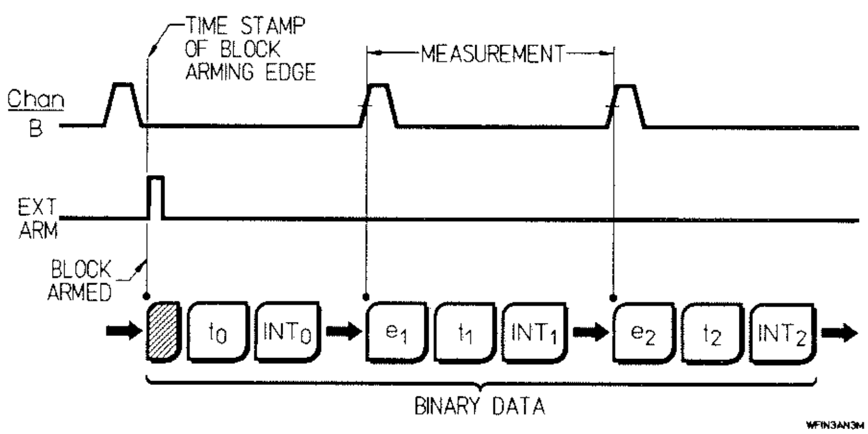


Figure 2B-A. Frequency Measurement on Channel B with Edge Holdoff Arming Mode Showing Corresponding Binary Data Output.

Producing Results

For all five functions, the first sample in each block is unique: it captures time of occurrence for the block arming edge. No event data is provided for this sample. Subsequent samples generate standard measurement results.

Continuous Time Interval results are generated by calculating elapsed time between contiguous samples.

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient. Avoid accidental misuse of the first sample in each block. The formulas are:

$$\textbf{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\textbf{PRI or Period} = \frac{\text{elapsed time}}{\text{elapsed events}}$$

Phase Deviation, Time Deviation, and Frequency Deviation results are generated by use of a special algorithm. The algorithm and an explanatory diagram are included at the end of this format section.

EXPANDED DATA

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between samples, but which are not time stamped. See *Glossary* in this chapter for more details. Avoid accidental misuse of the first sample in each block. The formula for Missed Events is:

Missed Events = [elapsed events between samples] – 1

Binary Data Output

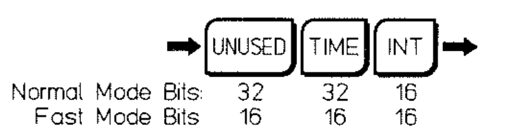


Figure 2B-B. Format 2B, First Sample in Each Block.

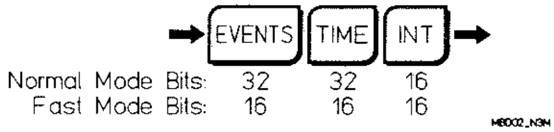


Figure 2B-C. Format 2B, Subsequent Samples.

Format 2B binary output is shown in *Figures 2B-B* and *2B-C*. The first sample in each block contains an unused data field as shown. The first sample and subsequent samples generate the binary data defined below.

First sample:

UNUSED FIELD
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Subsequent samples:

EVENT: binary event count.
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data To Event Stamps and Time Stamps

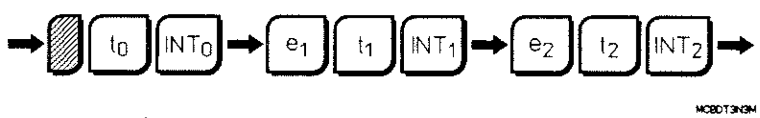


Figure 2B-D. Format 2B Binary Output

The binary data stream shown above was derived from the signal in *Figure 2B-A*. To produce valid event stamps, your program must:

1. Correct the binary event data for counter rollovers.

2. If measurement was made on Channel C, multiply the event stamp data by four.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path lengths between block arming channel and signal channel.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

With the exception of the first sample in each block, successive valid event counts increase monotonically. An overflow-processing routine must scan the event data for any datum which is both valid (not associated with the first sample in the block) and which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum, and to all subsequent data, then continue with the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

Channel C Correction

When making measurements on Channel C, the event counter sees only one out of every four input events. To correct for this, multiply event data by four. This must be done after correcting for counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. The first sample in each block is not an exception to this rule. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue with the scan. An example is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Block Arming Channel and Signal Channel

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, subtract a correction value, called an offset, from the time stamp associated with the first sample in each block. Subsequent time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the first sample in each block:

corrected time stamp = time stamp – offset

Generating Final Function Results from Time Stamps and Event Stamps

NOTE

The Block Start Status Bit is bit 6 of the interpolator/status field. The Block Start Status Bit is 1 for the first sample in a block. When this occurs, the corresponding time datum contains block arming data which is not part of the measurement. See the paragraph entitled Interpolator/Status Fields *in the* Special Topics *section of this chapter for more information.*

CONTINUOUS TIME INTERVAL RESULTS

Continuous Time Interval results are generated by calculating elapsed time between contiguous time stamps. For each contiguous pair:

Continuous Time Interval$_i$ = time stamp$_{i+1}$ – time stamp$_i$

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between samples but are not time stamped. The formula for calculating Missed Events is:

Missed Events$_i$ = [event stamp$_{i+1}$ – event stamp$_i$] – 1

PRF, FREQUENCY, PRI, AND PERIOD RESULTS

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient to produce the final result. The elapsed time between first and second sample in each block represents the elapsed time between the block arming edge and the first signal edge.

For the first sample in each block:

Block Arming Data = first measurement channel time stamp – block arming time stamp

For subsequent samples:

$$\text{PRF}_i \text{ or Frequency}_i = \frac{\text{event stamp}_{i+1} - \text{event stamp}_i}{\text{time stamp}_{i+1} - \text{time stamp}_i}$$

$$\text{PRI}_i \text{ or Period}_i = \frac{\text{time stamp}_{i+1} - \text{time stamp}_i}{\text{event stamp}_{i+1} - \text{event stamp}_i}$$

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating the elapsed time between samples.

Measurement Gate Time$_i$ = time stamp$_{i+1}$ – time stamp$_i$

TIME DEVIATION, PHASE DEVIATION, AND FREQUENCY DEVIATION RESULTS

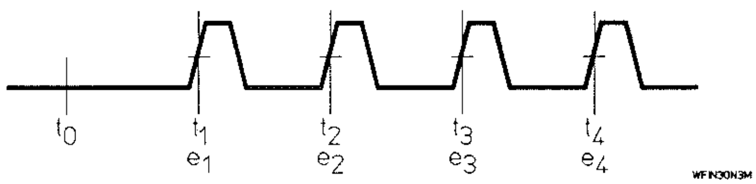


Figure 2B-E. Time Deviation and Phase Deviation Example Signal showing Time and Event Stamp Measurement Points.

$$\text{Time Deviation}_i = (e_{i+1} - e_1) \times P - (t_{i+1} - t_1)$$

$$\text{Phase Deviation}_i = \left[(e_{i+1} - e_1) \times P - (t_{i+1} - t_1)\right] \times \frac{360°}{P}$$

$$\text{Frequency Deviation}_i = \frac{(e_{i+1} - e_i)}{(t_{i+1} - t_i)} - f$$

Notes:

1. P = Period of reference signal
2. f = frequency of reference signal
3. Results calculation assumes that the time stamp of the block arming edge is sample t_0, excluded from the calculation shown. Sample t_1 is used as a reference edge; the first result is generated by manipulation of sample t_2.
4. Result is negative if signal lags reference.

FORMAT 3

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Continuous Time Interval	A, B	ON	Time Holdoff Event Holdoff
PRF, Frequency, PRI, Period	A, B, C	ON/OFF	Time Holdoff Event Holdoff Time/Interval Event/Interval

How Signals Are Converted to Binary Data

Example: Frequency B | Time Holdoff

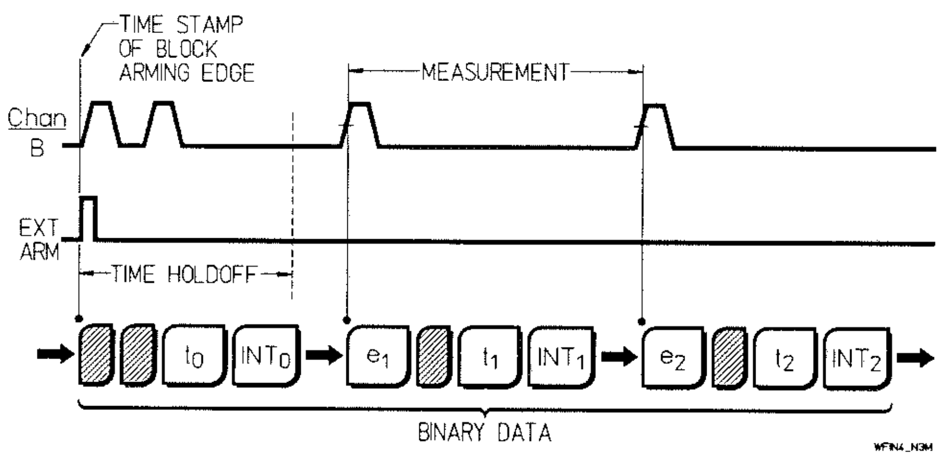


Figure 3-A. Frequency Measurement on Channel B with Time Holdoff Arming Mode Showing Corresponding Binary Data Output

Producing Results

For all three functions, the first sample in each block is unique: it captures time of occurrence for the block arming edge. No event data is provided for this sample. Subsequent samples generate standard measurement results.

Continuous Time Interval results are generated by calculating elapsed time between contiguous samples.

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient. Avoid accidental misuse of the first sample in each block. The formulas are:

$$\textbf{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\textbf{PRI or Period} = \frac{\text{elapsed time}}{\text{elapsed events}}$$

EXPANDED DATA

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between samples, but which are not time stamped. See the Glossary in this chapter for more details. Avoid accidental misuse of first sample in each block. The formula for calculating missed events is:

Missed Events = [elapsed events between samples] – 1

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Binary Data Output

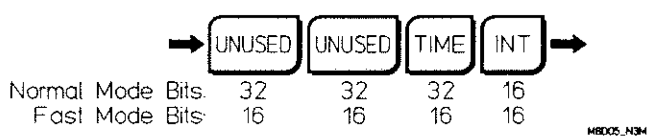


Figure 3-B. Format 3, First Sample in Each Block

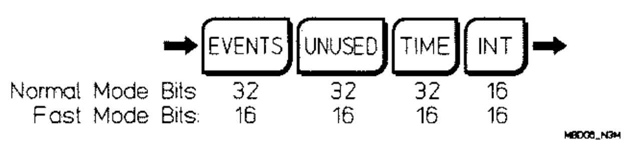


Figure 3-C. Format 3, Subsequent Samples

Format 3 binary output is shown in *Figures 3-B* and 3-C above. The first sample in each block contains two unused data fields as shown. Each sample generates the binary data defined below.

First Sample:

UNUSED FIELD
UNUSED FIELD
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

EVENT: binary event count current at sampling.
UNUSED FIELD
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

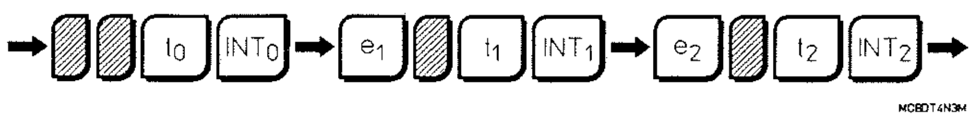


Figure 3-D. Format 3 Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 3-A*. To produce valid event stamps, your program must:

1. Correct the binary event data for counter rollovers.

2. If the measurement was made on Channel C, multiply the event stamps data by four.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.

2. Incorporate interpolator data.

3. Correct for differences in electrical path length between the block arming channel and the signal channel.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

With the exception of the first sample in each block, successive valid event counts increase monotonically. An overflow-processing routine must scan the event data for any datum which is both valid (not associated with the first sample in the block), and is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and to all subsequent data, then continue with the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

Channel C Correction

When making measurements on Channel C, the event counter sees only one out of every four input events. To correct for this, multiply event data by four. This must be done after correcting for counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. The first sample in each block is not an exception to this rule. An overflow- processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue the scan. An example is provided in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Block Arming Channel and Signal Channel

There are several possible signal paths running from input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, subtract a correction value, called an offset, from the time stamp associated with the first sample in each block. Subsequent time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the first sample in each block:

corrected time stamp = time stamp – offset

Generating Final Function Results from Time Stamps and Event Stamps

NOTE

The Block Start Status Bit is bit 6 of the interpolator/status field. The Block Start Status bit is 1 for the first sample in a block. See Interpolator/Status Field Data *in the* Special Topics *section of this chapter for more information.*

CONTINUOUS TIME INTERVAL RESULTS

Continuous Time Interval results are generated by calculating elapsed time between samples. Elapsed time between the first and second sample in each block represents the elapsed time between the block arming edge and the first signal edge.

For the first sample in each block:

Block arming data = first measurement channel time stamp – block arming timestamp.

For subsequent samples:

Continuous Time Interval$_i$ = time stamp$_{i+1}$ – time stamp$_i$

Expanded Data for Continuous Time Interval measurements is Missed Events. Missed Events are defined as trigger events which fall between samples but are not time stamped. The formula for calculating Missed Events is:

Missed Events$_i$ = [event stamp$_{i+1}$ – event stamp$_i$] – 1

PRF, FREQUENCY, PRI, AND PERIOD RESULTS

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between samples, then taking the appropriate quotient to produce the final result. The elapsed time between the first and second sample in each block represents the elapsed time between the block arming edge and the first sampled signal edge.

For first sample in each block:

Block arming data = first measurement channel time stamp – block arming time stamp

For subsequent samples:

FORMAT 3

$$\textbf{PRF}_i \textbf{ or Frequency}_i = \frac{\text{event stamp}_{i+1} - \text{event stamp}_i}{\text{time stamp}_{i+1} - \text{time stamp}_i}$$

$$\textbf{PRI}_i \textbf{ or Period}_i = \frac{\text{time stamp}_{i+1} - \text{time stamp}_i}{\text{event stamp}_{i+1} - \text{event stamp}_i}$$

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating the elapsed time between samples.

Measurement Gate Time$_i$ = time stamp$_{i+1}$ – time stamp$_i$

FORMAT 4A

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Time Interval	A, B, A → B, B → A	OFF	Automatic Edge Holdoff Time Holdoff Event Holdoff Interval Sampling Repetitive Edge Sampling Random Sampling Edge/Interval Edge/Random

How Signals are Converted to Binary Data

Example: Time Interval A → B | Automatic

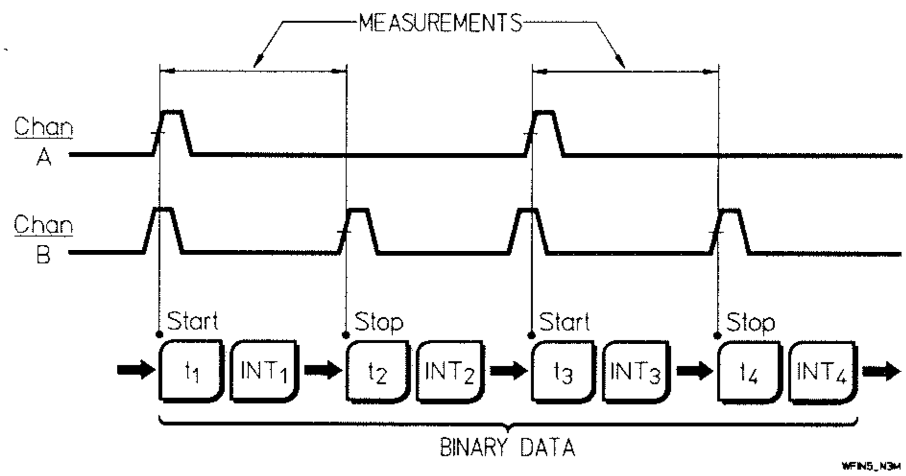


Figure 4A-A. Time Interval Measurement on Channel A → B with Automatic Arming Mode Showing Corresponding Binary Data Output

Producing Results

Time Interval results are generated by calculating elapsed time between start and stop samples.

NOTE

With Expanded Data OFF, the counts of elapsed trigger events on Channel A and Channel B are unavailable.

Binary Data Output

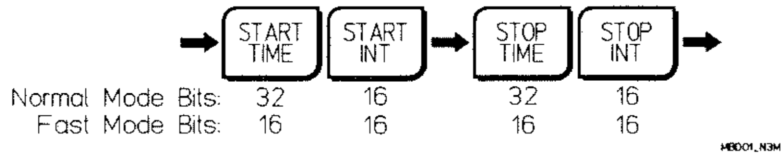


Figure 4A-B. Format 4A Binary Output

Format 4A binary output is shown in *Figure 4A-B* above. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* to determine the correct start and stop channels. Each sample generates the binary data defined below:

TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Time Stamps

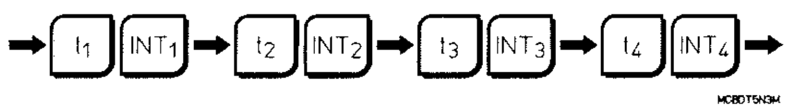


Figure 4A-C. Format 4A Binary Output

The binary data stream shown above was derived from the signals shown in *Figure 4A-A*. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent data, then continue with the scan.

An example routine is given in *Technical Note* 3. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels

There are several possible signal paths running from input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps

Time Interval results are generated by calculating elapsed time between start time stamp and stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Format 4B

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
±Time Interval	A, B, A→ B B→ A	OFF	Automatic Interval Sampling Edge Holdoff Parity Sampling Repetitive Edge Sampling Repetitive Edge-Parity Sampling Random Sampling Edge/Interval Edge/Event Edge/Parity Edge/Random Time/Time Event/Event
±Time Interval	A, B, A→ B B→ A	ON	Time/Time Event/Event
±Time Interval	A→ B B→ A	ON	Edge/Event

NOTE

In the first of the three measurement modes listed above, Expanded Data OFF gives this format (4B). Event data is not available with Expanded Data OFF. In the second and third measurement modes, event data is unavailable regardless of whether Expanded Data is ON or OFF. Thus, in all three measurement modes listed above, event data is not available in this format.

How Signals are Converted to Binary Data

Example: ±Time Interval B → A | Automatic

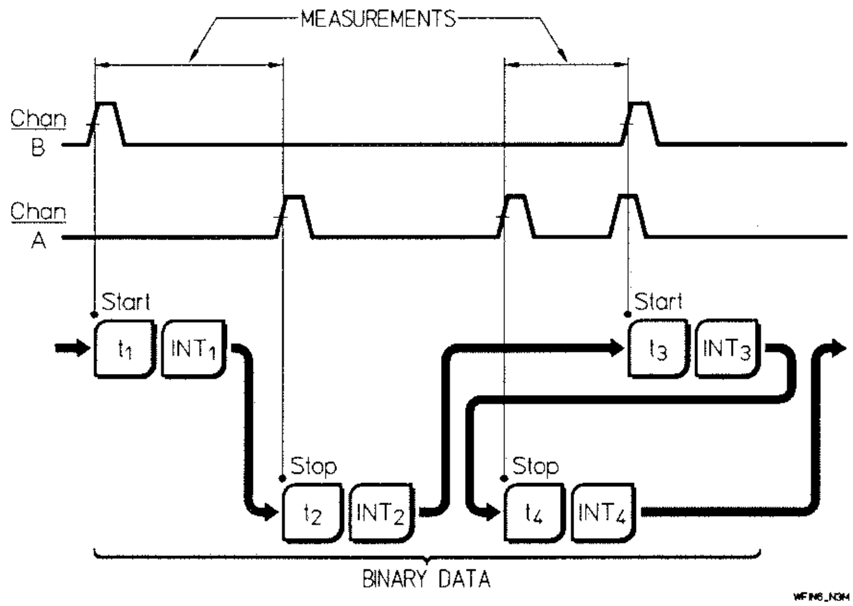


Figure 4B-A. Time Interval Measurement on Channel A → B with Automatic Arming Mode Showing Corresponding Binary Data Output

Producing Results

±Time Interval results are generated by calculating the elapsed time between start and stop events. Characteristic of ±Time Interval measurements, start may follow stop, as shown above. In these cases the measurement result is negative.

Binary Data Output

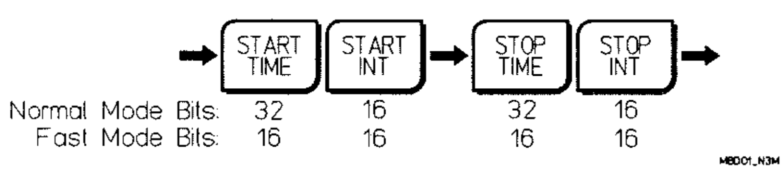


Figure 4B-B. Format 4B Binary Output

Format 4B binary output is shown in *Figure 4B-B* above. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Note that the binary output stream will always deliver the start channel time datum first, followed by the stop channel datum — even if the start sample was collected after the stop sample.

Each sample generates the binary data defined below:

TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Time Stamps

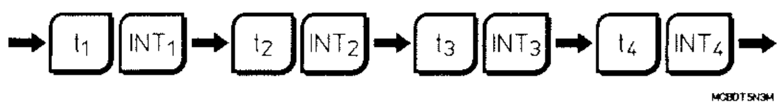


Figure 4B-C. Format 4B Binary Output

The binary data stream shown above was derived from the input signals shown in *Figure 4B-A*. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Since the ±Time Interval function will allow the stop sample to be collected before the start sample, time data does not always increase monotonically. When start and stop are reversed, (stop before start) the stop time count is (correctly) smaller than the start time count. Overflow processing routines must distinguish between start-stop reversals and counter rollovers. Overflow processing routines must correct the data only when counter rollover is detected.

Correction for counter rollovers can be accomplished with two scans of the time data. A scan correcting for rollovers which occur between start and stop is described first; second, a scan is described which corrects for rollovers which occur between measurements.

Correcting for Rollovers Between Start and Stop

Counter rollovers which occur between start and stop are detected when subtraction of start data from stop data produces an out-of-range result. If a subtraction produces a number greater than +Maximum, leave the stop datum uncorrected and add the appropriate Time Overflow Correction to the start datum. Make the same correction to all subsequent data (both start and stop). Following the corrections, continue subtracting start data from stop data to detect further rollovers.

If a subtraction produces a number smaller than –Maximum, leave the start datum uncorrected, and add the appropriate Time Overflow Correction to the stop datum. Make the same correction to all subsequent data (both start and stop). Following the corrections, resume the scan at the next start datum. Time Overflow Corrections are given in *Technical Note 1*. Values for +Maximum and –Maximum are given in *Technical Note 2*.

Correcting for Rollovers Between Measurements

Once the first scan is complete, a second scanning routine detects and corrects for counter rollovers which occur between start-stop pairs. The first scan leaves the time data in this condition: the smallest time value in each pair should be greater than the largest time value in the immediately preceding pair (unless a counter rollover has occurred between pairs). The second scan identifies the smaller time value in each pair, called "true start", then identifies the larger time value in the preceding pair, that pair's "true stop". If true stop is greater than true start, a counter rollover has occurred between pairs. The appropriate Time Overflow Correction should be added to all data subsequent to true stop. Following the corrections, resume the scan at the next pair. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps

±Time Interval results are generated by calculating elapsed time between start time stamp and stop time stamp. Characteristic of ±Time Interval measurements, start may follow stop as shown in *Figure 4B-A*. In such a case, the resulting measurement is negative.

±Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Format 5A

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Time Interval	A, B	ON	Automatic Edge Holdoff Interval Sampling Repetitive Edge Sampling Random Sampling Edge/Interval Edge/Random
PRF, Frequency, PRI, Period	A, B, C	ON/OFF	Time Sampling Edge/Time Edge/Event Externally Gated

How Signals are Converted to Binary Data

Example: Frequency A | Externally Gated

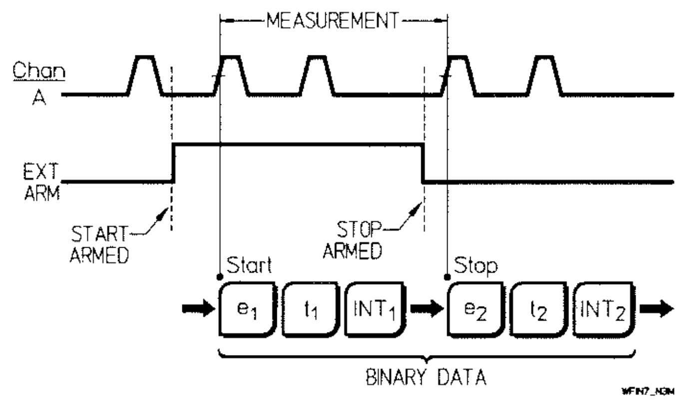


Figure 5A-A. Frequency Measurement on Channel A with Externally Gated Arming Mode Showing Corresponding Binary Data Output

Producing Results

Time Interval results are generated by calculating the elapsed time between start and stop samples.

PRF, Frequency, PRI, and Period results are generated by calculating both the elapsed time and elapsed events between start and stop samples, then taking the appropriate quotient. The formulas are:

$$\text{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\text{PRI or Period} = \frac{\text{elapsed time}}{\text{elapsed events}}$$

EXPANDED DATA

Expanded Data for Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between measurements, but which are not time stamped. See the *Glossary* in this chapter for more details.

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between start and stop.

Binary Data Output

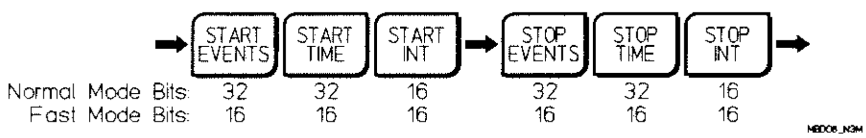


Figure 5A-B. Format 5A Binary Output

Format 5A binary output is shown in *Figure 5A-B*. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Each sample generates the set of binary data defined below:

EVENT: binary event count.
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

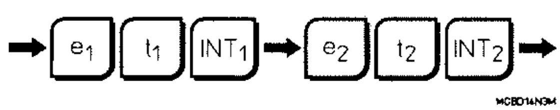


Figure 5A-C. Format 5A Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 5A-A*. To produce valid event stamps, your program must:

1. Correct the binary event data for counter rollovers.
2. Multiply the event stamp data by four if the measurement was made on Channel C.

To produce valid time stamps your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

Successive event counts increase monotonically (see NOTE below) unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and to all subsequent data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1.*

NOTE

Measurements armed as Time Sampling, Edge/Time, or Edge/Event will reset the event counters sometime before each start sample. Overflow-processing routines, for measurements using these arming modes, should expect event counts to increase when comparing a start sample to its corresponding stop sample. However, because of the reset, it is incorrect to compare the current start sample with the stop sample which immediately preceded it. Time counters are not reset, therefore time data will be continuous except for counter rollovers.

Channel C Correction

When making measurements on Channel C, the event counter sees only every fourth input event. To correct for this, multiply event data by four. This must be done after correcting for counter rollovers.

PRODUCING VALID TIME STAMPS

FORMAT 5A

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum, and all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels.

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Event and Time Stamps

TIME INTERVAL RESULTS

Time Interval results are generated by calculating the elapsed time between the start time stamp and the stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for Time Interval measurements is Missed Events. Missed Events are events which are counted between measurements but are not time stamped; they are events missed between current stop and next start. The formula is:

Missed Events$_i$ = [start event stamp$_{i+1}$ – stop event stamp$_i$] – 1

PRF, FREQUENCY PRI, AND PERIOD RESULTS

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events, then taking the appropriate quotient to produce the final result.

$$\textbf{PRF}_i \textbf{ or Frequency}_i = \frac{\text{stop event stamp}_i - \text{start event stamp}_i}{\text{corrected stop time stamp}_i - \text{start time stamp}_i}$$

$$\textbf{PRI}_i \textbf{ or Period}_i = \frac{\text{corrected stop time stamp}_i - \text{start time stamp}_i}{\text{stop event stamp}_i - \text{start event stamp}_i}$$

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating the elapsed time between current start and stop time stamps.

$$\textbf{Measurement Gate Time}_i = \text{corrected stop time stamp}_i - \text{start time stamp}_i$$

Format 5B

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
±Time Interval	A, B	ON	Edge/Event

How Signals are Converted to Binary Data

Example: ±Time Interval A | Edge/Event

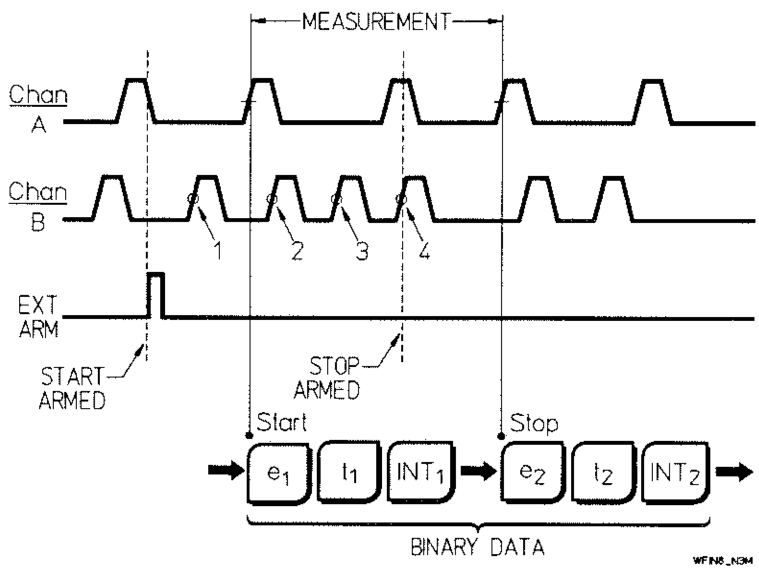


Figure 5B-A. ±Time Interval Measurement on Channel A with Edge/Event Arming Mode Showing Corresponding Binary Data Output

Producing Results

±Time Interval results are generated by calculating elapsed time between start and stop samples.

EXPANDED DATA

Expanded Data for ±Time Interval measurements is Missed Events.

Missed Events are defined as trigger events which are counted between start and stop samples, but which are not time stamped. See *Glossary* in this chapter for more details. The formula is:

Missed Events = (elapsed events between start and stop) – 1.

NOTE

This definition is intentionally different from the definition of Missed Events for all other ±Time Interval and Time Interval measurements. In other ±Time Interval and Time Interval measurements, Missed Events are events which occur between the stop of one measurement, and the start of the next measurement. In this format (5B), Missed Events are those which occur between the start and stop events of an individual measurement. In other words, Missed Events in this format occur within the measurement, rather than between measurements.

Binary Data Output

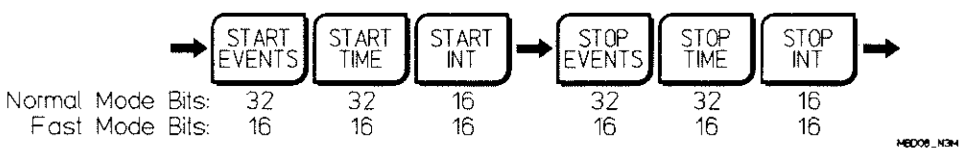


Figure 5B-B. Format 5B Binary Output

Format 5B binary output is shown in *Figure 5B-B* above. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Each sample generates the binary data defined below:

EVENT: binary event count.
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

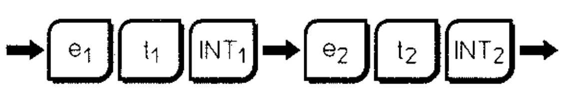


Figure 5B-C: Format 5B Binary Output

The binary data stream shown above was derived from the signals shown in *Figure 5B-A*. To produce valid event stamps, your program must correct the binary event data for counter rollovers. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

In this format the event counters are reset before each start sample. A routine which checks for event counter rollover need only check the start and stop event data of a given measurement. If the stop datum is smaller than the start datum, a counter rollover has occurred and the appropriate overflow correction must be added to the stop datum. After the correction has been made, the routine should continue checking the data at the next start datum. Event Overflow Correction values are given in *Technical Note 1*. Time counters, on the other hand are not reset, so time data will be continuous between counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Since the ±Time Interval function will allow the stop sample to be collected before the start sample, time data does not always increase monotonically. When start and stop are reversed, (stop before start) the stop time count is (correctly) smaller than the start time count. Overflow processing routines must distinguish between start-stop reversals and counter rollovers. Overflow processing routines correct the data only when a counter rollover is detected.

Correction for counter rollovers can be accomplished in two scans of the time data. A scan correcting for rollovers between start and stop is described first; second, a scan is described which corrects for rollovers occurring between measurements.

Correcting for Rollovers Between Start and Stop

Counter rollovers which occur between start and stop are detected when subtraction of start data from stop data produces an out-of-range result. If the subtraction produces a number greater than +Maximum, leave the stop datum uncorrected and add the appropriate Time Overflow Correction to the start datum. If the subtraction produces a number smaller than –Maximum, leave the start datum uncorrected, and add the appropriate Time Overflow Correction to the stop datum. Add the appropriate Time Overflow Correction value to all subsequent data (both start and stop) in either case. Following correction, resume the scan at the next measurement. Time Overflow Correction values are given in *Technical Note 1*. Values for +Maximum and –Maximum are given in *Technical Note 2*.

FORMAT 5B

Correcting for Rollovers Between Measurements

Once the first scan is complete, a second scanning routine detects and corrects for counter rollovers between start-stop pairs. The first scan leaves the time data in this condition: the smallest time value in each pair should be greater than the largest time value in the preceding pair (unless a counter rollover has occurred between pairs). The second scan identifies the smallest time value in each pair, called "true start", and identifies the largest time value in the preceding pair, called "true stop". If true stop is greater than true start, a counter rollover has occurred between pairs. The appropriate Time Overflow Correction should be added to all data subsequent to true stop. Following correction, resume the scan at the next pair. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels.

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Event and Time Stamps

Time Interval results are generated by calculating elapsed time between start time stamp and stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for Time Interval measurements is Missed Events. Missed Events are trigger events which are counted between start and stop samples in the same measurement, but which are not time stamped. The formula is:

Missed Events$_i$ = [stop event stamp$_i$ – start event stamp$_i$] – 1

Format 6

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Time Interval	A→ B B→ A	ON	Time Holdoff Event Holdoff

How Signals are Converted to Binary Data

Example: Time Interval A→ B | Time Holdoff

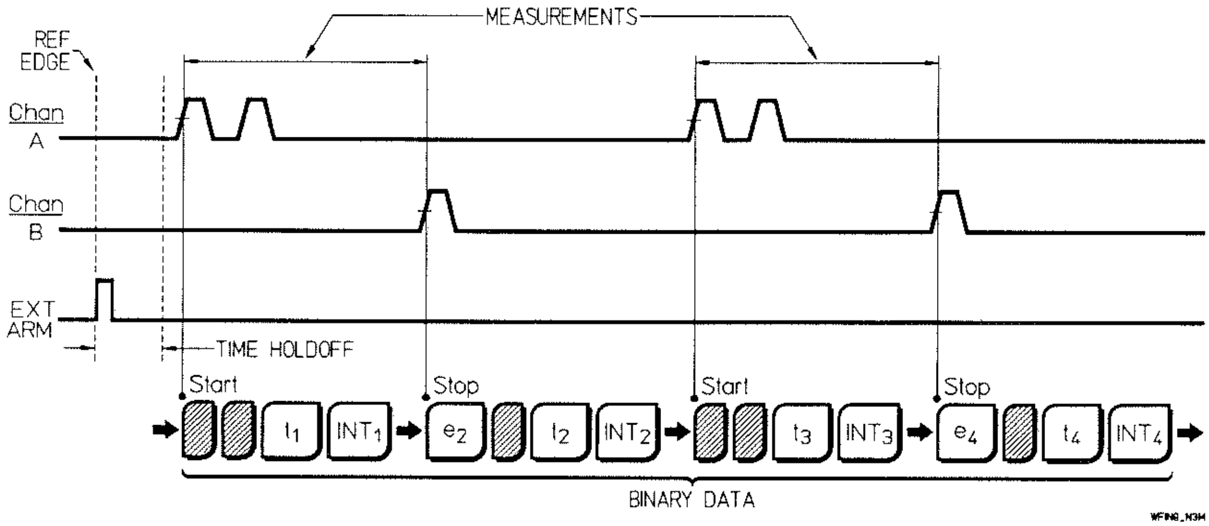


Figure 6-A. Time Interval A→ B Measurement With Time Holdoff Arming Mode Showing Corresponding Binary Data Output.

Producing Results

Time Interval results are generated by calculating the elapsed time between start and stop samples.

EXPANDED DATA

Expanded Data for Time Interval A→ B and Time Interval B→ A is normally Missed Start Channel Events and Missed Stop Channel Events. For this format (6), only Missed Stop Channel Events are available.

Missed Stop Channel Events are defined as stop channel trigger events which are counted between stop samples, but which are not time stamped. See the *Glossary* in this chapter for more details. The formula is:

Missed Stop Channel Events = (elapsed stop channel events between successive stops) – 1

Binary Data Output

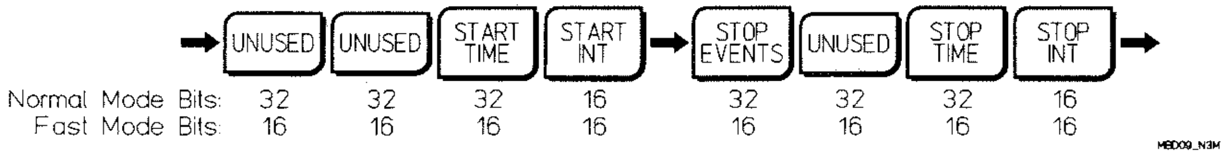


Figure 6-B. Format 6 Binary Output

Format 6 binary output is shown in *Figure 6-B* above. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Each pair of samples (each measurement), generates the set of binary data described below:

UNUSED FIELD
UNUSED FIELD
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

STOP CHANNEL EVENT: binary event count on the stop channel,
UNUSED FIELD
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

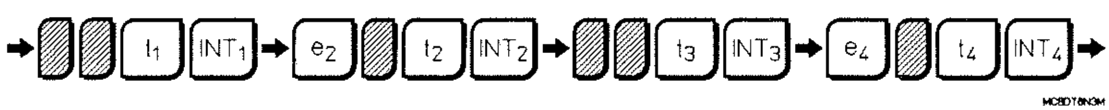


Figure 6-C. Format 6 Binary Output

The binary data stream shown above was derived from the signals shown in *Figure 6-A*. To produce valid event stamps, your program must correct the binary stop channel data for event counter rollovers. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.

2. Incorporate interpolator data.

3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Stop Channel Data for Counter Rollovers

Successive stop channel event counts increase monotonically unless a counter rollover has occurred. Note that stop channel event counts are provided with every other sample. Monotonic increase should be expected for data e2, e4, e6, etc. (see *Figure 6-A*). An overflow-processing routine must scan the stop event data for any stop event datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and to all subsequent stop data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1.*

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum, and to all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3.* Time Overflow Correction values are given in *Technical Note 1.*

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Start and Stop Channels.

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Functions Results from Event and Time Stamps

Time Interval results are generated by calculating elapsed time between start time stamp and stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for Time Interval measurements is Missed Stop Channel Events. Missed Stop Channel Events are trigger events which are counted between stop channel samples, but which are not time stamped. The formula is:

Missed Stop Channel Events$_i$ = [stop event stamp$_{i+1}$ – stop event stamp$_i$] – 1

Format 8

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Time Interval	A, B	ON	Time Holdoff Event Holdoff

How Signals Are Converted to Binary Data

Example: Time Interval B | Time Holdoff

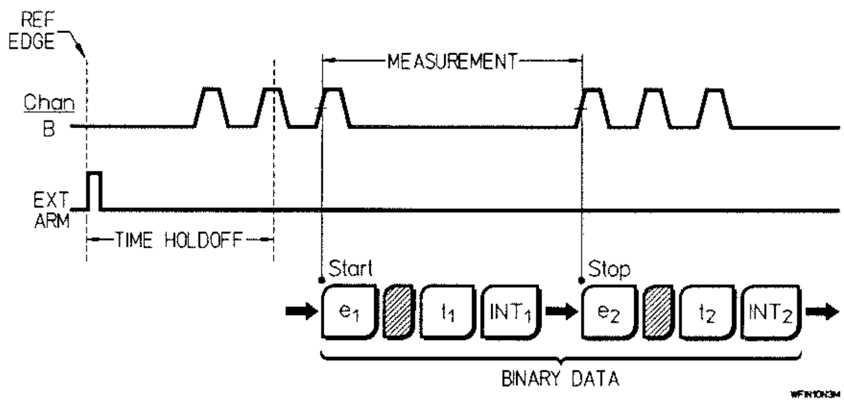


Figure 8-A. Time Interval Measurement on Channel B With Time Holdoff Arming Mode Showing Corresponding Binary Output.

Producing Results

Time Interval results are generated by calculating elapsed time between start and stop samples.

EXPANDED DATA

Expanded Data for Time Interval measurements is Missed Events. Missed Events are defined as trigger events which are counted between measurements, but which are not time stamped. See Glossary in this chapter for more details.

Binary Data Output

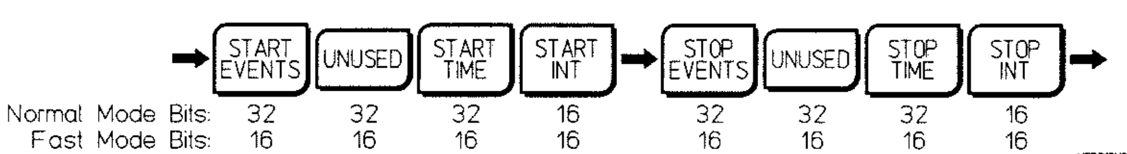


Figure 8-B. Format 8 Binary Output.

Format 8 binary output is shown in *Figure 8-B* above. Each measurement consists of two samples: start and stop.

Each sample generates the binary data defined below:

EVENT: binary event count.
UNUSED FIELD
TIME: time of occurrence for each sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

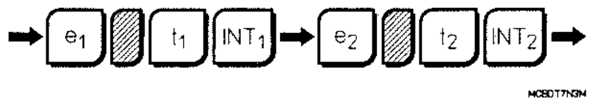


Figure 8-C. Format 8 Binary Output

The binary data stream shown above was derived from the signal shown in *Figure 8-A*. To produce valid event stamps, your program must correct the binary event data for counter rollovers. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Event Data for Counter Rollovers

Successive event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any event datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum, and to all subsequent data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1*.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data, seeking any any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum, and to all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Start and Stop Channels

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application.

For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps

Time Interval results are generated by calculating elapsed time between the start time stamp and stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for Time Interval measurements is Missed Events. Missed Events are trigger events which are counted between measurements, but which are not time stamped; they are events missed between current stop and next start. The formula is:

Missed Events$_i$ = [start event stamp$_{i+1}$ – stop event stamp$_i$] – 1

Format 10A

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Time Interval	A→ B, B→ A	ON	Automatic Edge Holdoff Interval Sampling Random Sampling Edge/Interval Edge/Random Repetitive Edge Sampling

How Signals are Converted to Binary Data

Example: Time Interval A → B | Automatic

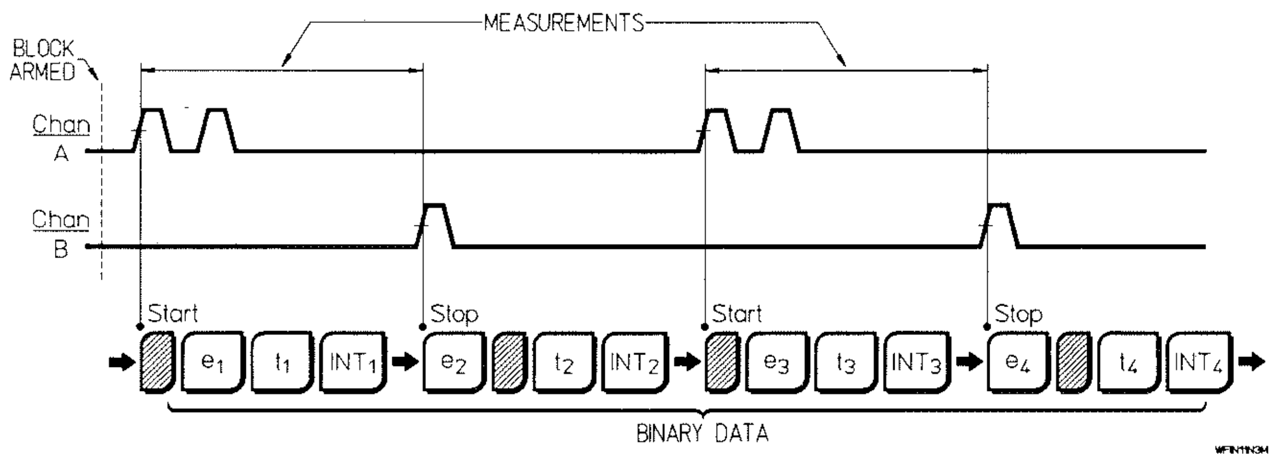


Figure 10A-A. Time Interval Measurement on Channel A → B With Automatic Arming Mode Showing Corresponding Binary Output.

Producing Results

Time Interval results are generated by calculating the elapsed time between the start and stop samples.

EXPANDED DATA

Expanded Data for Time Interval A → B and Time Interval B → A measurements is Missed Start Channel Events and Missed Stop Channel Events.

Missed Start Channel Events are defined as start channel trigger events which are counted between start samples, but which are not time stamped. See *Glossary* for more details. The Formula is:

Missed Events = (events on the start channel between successive starts) – 1

Missed Stop Channel Events are defined as stop channel trigger events which are counted between stop samples, but which are not time stamped. See *Glossary* for more details. The formula is:

Missed Events: (events on the stop channel between successive stops) – 1

Binary Data Output

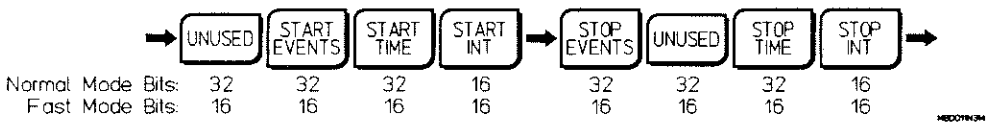


Figure 10A-B. Format 10A Binary Output

Format 10A binary output is shown in *Figure 10A-B*. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Each pair of samples, (each measurement) generates the binary data defined below:

UNUSED FIELD
START CHANNEL EVENT: binary event count on the start channel. For example: Channel B events for Time Interval B → A measurement.
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

STOP CHANNEL EVENT: binary event count on the stop channel. For example: Channel A events for Time Interval B → A measurement.
UNUSED FIELD
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

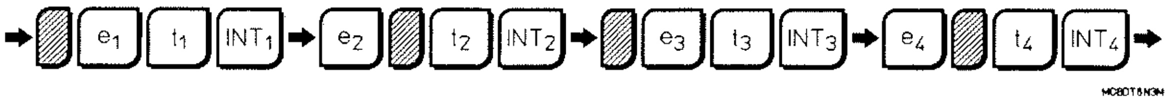


Figure 10A-C: Format 10A Binary Output

The binary data stream shown above was derived from the signals shown in *Figure 10A-A*. To produce valid event stamps, your program must:

1. Correct the start channel binary event data for counter rollovers.
2. Correct the stop channel binary event data for counter rollovers.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between start channel and stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Start Channel Data for Counter Rollovers

Start channel event counts increase monotonically unless a counter rollover has occurred. Note that start channel event counts are provided with every other sample. Monotonic increase should be expected for data e1, e3, e5, etc. See *Figure 10A-A*. An overflow-processing routine must scan the start data for any start datum smaller than its immediate predecessor. On detection of such a datum, the the routine must add the appropriate Event Overflow Correction value to that start datum and to all subsequent start data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1*.

Correcting Stop Channel Data for Counter Rollovers

This correction process parallels that performed for start channel data. Successive stop channel event counts increase monotonically unless a counter rollover has occurred. Note that stop channel event counts are provided with every other sample. Monotonic increase should be expected for e2, e4, e6, etc. See *Figure 10A-A*. The start data overflow processing routine may be used here by providing it with stop data. The routine should then find the rollover points and add the overflow correction to all subsequent data.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and to all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels.

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps

Time Interval results are generated by calculating elapsed time between the start time stamp and stop time stamp.

Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for Time Interval is Missed Events. Missed Events are trigger events which are counted, but which are not time stamped.

Missed Start Channel Events$_i$ = [start event stamp$_{i+1}$ – start event stamp$_i$] – 1

Missed Stop Channel Events$_i$ = [stop event stamp$_{i+1}$ – stop event stamp$_i$] – 1

Format 10B

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
±Time Interval	A→ B, B→ A	ON	Automatic Edge Holdoff Interval Sampling Parity Sampling Repetitive Edge Sampling Repetitive Edge-Parity Sampling Random Sampling Edge/Interval Edge/Parity Edge/Random
Pulse: Width, Offtime, Rise Time, Fall Time, Duty Cycle	A	ON/OFF	Automatic
Phase	A rel B, B rel A	ON/OFF	Automatic Interval Sampling Edge Holdoff Edge/Interval

How Signals are Converted to Binary Data

Example: ±Time Interval B → A \| Automatic

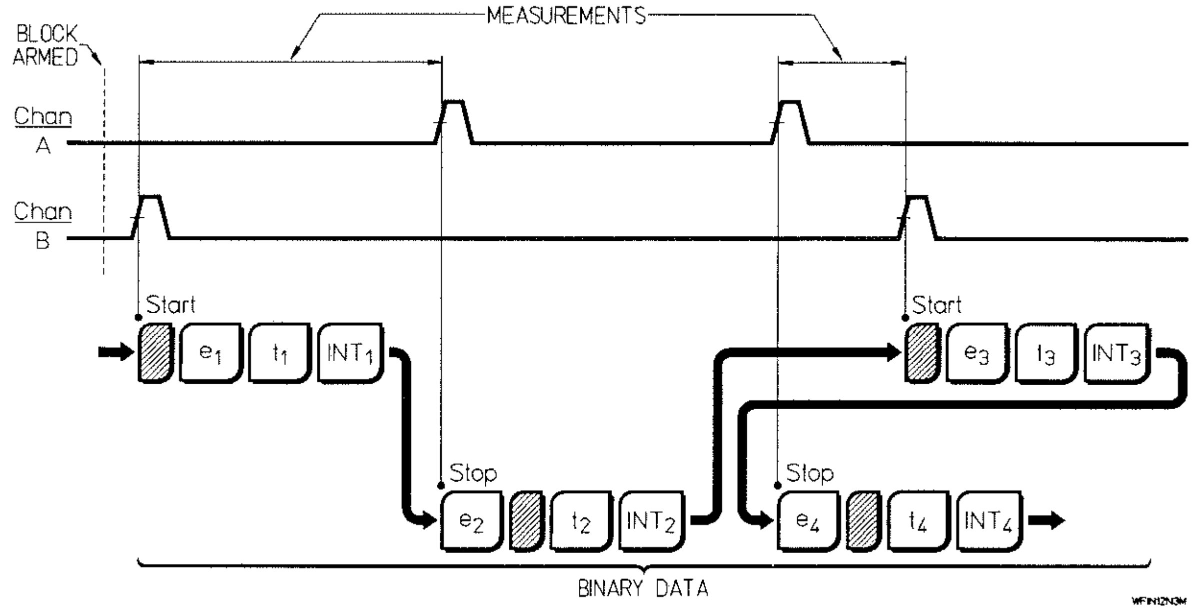


Figure 10B-A. ±Time Interval B → A Measurement on Channel B → A with Automatic Arming Mode Showing Corresponding Binary Output.

Producing Results

±Time Interval results are generated by calculating the elapsed time between start and stop samples. Characteristic of ±Time Interval measurements, start may follow stop, as shown above. In these cases, the measurement result is negative.

EXPANDED DATA

Expanded Data for ±Time Interval A → B or ±Time Interval B → A measurements is Missed Start Channel Events and Missed Stop Channel Events.

Missed Start Channel Events are defined as start channel trigger events which are counted between start samples, but which are not time stamped. The formula is:

Missed Events = (events elapsed on the start channel between successive starts) – 1

Missed Stop Channel Events are defined as stop channel trigger events which are counted between stop samples, but which are not time stamped. The formula is:

Missed Events = (events elapsed on the stop channel between successive stops) – 1

Measurement results for Duty Cycle, Rise Time, Fall Time, Pulse Width, Pulse Offtime, and Phase are captured when the data generated by these measurements is combined with the appropriate algorithms. For each measurement, a diagram and an algorithm is given in the section of this format (10B) entitled *Generating Final Function Results.*

Binary Data Output

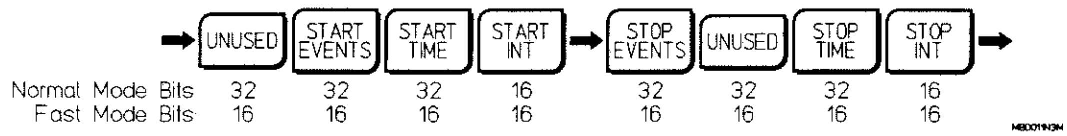


Figure 10B-B. Format 10B Binary Output

FORMAT 10B

Format 10B binary output is shown in *Figure 10B-B* above. Each measurement consists of two samples: start and stop. Measurements in this format require you to determine which input channel generates the start data, and which channel generates the stop data. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Note that the binary output stream will always deliver the start channel datum first, followed by the stop channel datum — even if the start sample was collected after the stop sample. Each pair of samples, (each measurement) generates the set of binary data described below:

UNUSED FIELD
START CHANNEL EVENT: binary event count on the start channel. For example, Channel B events for ±Time Interval B→ A measurement.
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

STOP CHANNEL EVENT: binary event count on the stop channel. For example: Channel A events for ±Time Interval B→ A measurement.
UNUSED FIELD
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

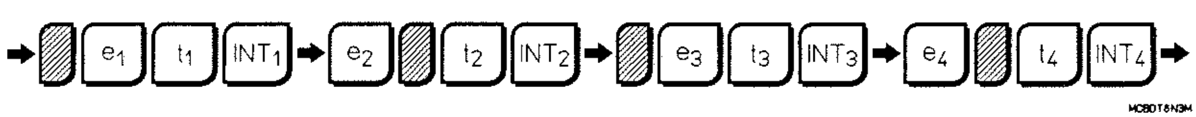


Figure 10B-C. Format 10B Binary Output

FORMAT 10B

The binary data stream was derived from the signals shown in *Figure 10B-A*. To produce valid event stamps, your program must:

1. Correct the start channel binary event data for counter rollovers.
2. Correct the stop channel binary event data for counter rollovers.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between start channel and stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Start Channel Data for Counter Rollovers

Start channel event counts increase monotonically unless a counter rollover has occurred. Note that start channel event counts are provided with every other sample. Monotonic increase should be expected for data e1, e3, e5, etc. See *Figure 10B-A*. An overflow-processing routine must scan the start data for any start event datum smaller than its immediate predecessor. On detection of such an event datum, the routine must add the appropriate Event Overflow Correction value to that start datum, and to all subsequent start data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1*.

Correcting Stop Channel Data for Counter Rollovers

This correction process parallels that performed for start channel data. Stop channel binary event counts increase monotonically unless a counter rollover has occurred. Note that stop channel event counts are provided with every other sample. Monotonic increase should be expected for e2, e4, e6, etc. See *Figure 10B-A*. The start data overflow processing routine may be used here by providing it with stop data. The routine should then find the rollover points and add the overflow correction value to all subsequent data.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Since the ±Time Interval function will allow the stop sample to be collected before the start sample, time data does not always increase monotonically. When start and stop are reversed, (stop before start) the stop time count is (correctly) smaller than the start time count. Overflow processing routines must distinguish between start-stop reversals and counter rollovers. Overflow processing routines correct the data only when a counter rollover is detected.

Correction for counter rollovers can be accomplished in two scans of the time data. A scan correcting for rollovers between start and stop is described first; second, a scan is described which corrects for rollovers occurring between measurements.

Correcting for Rollovers Between Start and Stop

Counter rollovers which occur between start and stop are detected when subtraction of start data from stop data produces an out-of-range result. If the subtraction produces a number greater than +Maximum, leave the stop datum uncorrected and add the appropriate Time Overflow Correction to the start datum. If the subtraction produces a number smaller than –Maximum, leave the start datum uncorrected, and add the appropriate Time Overflow Correction to the stop datum. Add the appropriate Time Overflow Correction value to all subsequent data (both start and stop) in either case. Following correction, resume the scan at the next measurement. Time Overflow Correction values are given in *Technical Note 1*. Values for +Maximum and –Maximum are given in *Technical Note 2*.

Correcting for Overflows Between Measurements

Once the first scan is complete, a second scanning routine detects and corrects for counter rollovers between start-stop pairs. The first scan leaves the time data in this condition: the smallest time value in each pair should be greater than the largest time value in the preceding pair (unless a counter rollover has occurred between pairs). The second scan identifies the smallest time value in each pair, called "true start", and identifies the largest time value in the preceding pair, called "true stop". If true stop is greater than true start, a counter rollover has occurred between pairs. The appropriate Time Overflow Correction should be added to all data subsequent to true stop. Following correction, resume the scan at the next pair. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. The formula is:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results

±Time Interval results are generated by calculating elapsed time between start and stop time stamps.

±Time Interval$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Expanded Data for ±Time Interval A → B or ±Time Interval B → A is Missed Start Channel Events and Missed Stop Channel Events.

Missed Start Channel Events are defined as start channel trigger events which are are counted between start samples but which are not time stamped. The formula is:

Missed Start Channel Events$_i$ = [start event stamp$_{i+1}$ – start event stamp$_i$] – 1

Missed Stop Channel Events are defined as stop channel trigger events which are counted between stop samples but which are not time stamped. The formula is:

Missed Stop Channel Events$_i$ = [stop event stamp$_{i+1}$ – stop event stamp$_i$] – 1

DUTY CYCLE

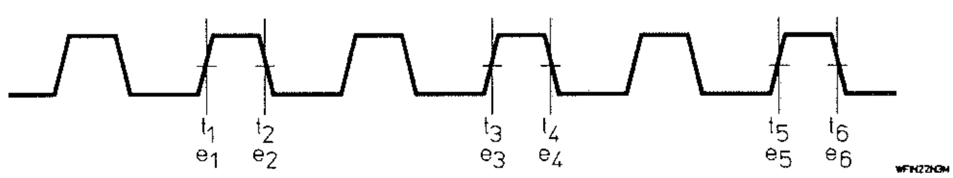


Figure 10B-D. Duty Cycle Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Duty Cycle is given below. In *Figure 10B-D,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Duty Cycle$_i$ $=[t_{2i} - t_{2i-1}] \times \frac{e_{2i+1} - e_{2i-1}}{t_{2i+1} - t_{2i-1}} \times 100\%$

RISE TIME

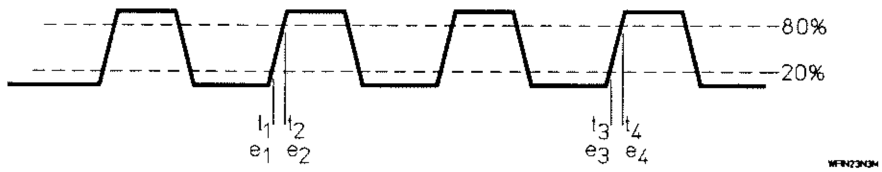


Figure 10B-E. Rise Time Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Rise Time is given below. In *Figure 10B-E,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Rise Time$_i$ $= t_{2i} - t_{2i-1}$

FALL TIME

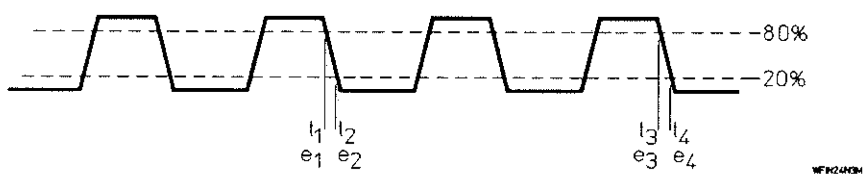


Figure 10B-F. Fall Time Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Fall Time is given below. In *Figure 10B-F,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Fall Time$_i$ $= t_{2i} - t_{2i-1}$

PULSE WIDTH

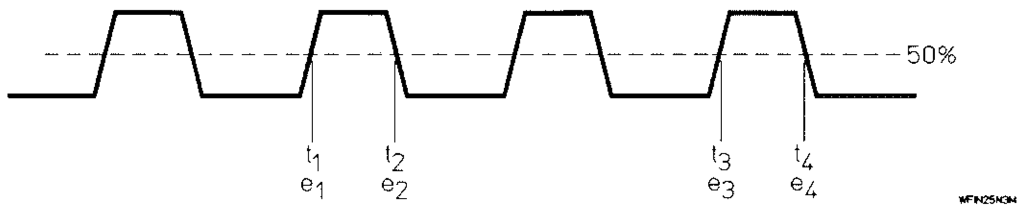


Figure 10B-G. Positive Pulse Width Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Pulse Width is given below. In *Figure 10B-G,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Pulse Width$_i = t_{2i} - t_{2i-1}$

PULSE OFFTIME

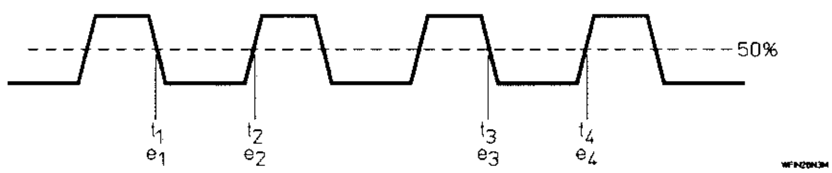


Figure 10B-H. Negative Pulse Width Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Pulse Offtime is given below. In *Figure 10B-H,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Pulse Offtime$_i = t_{2i} - t_{2i-1}$

PHASE

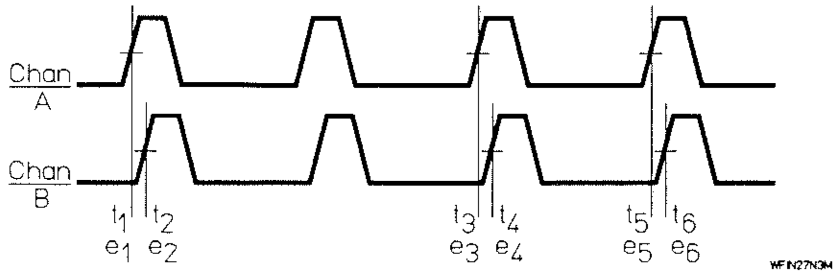


Figure 10B-I. Phase Example Signal Showing Time and Event Stamp Measurement Points

The formula for calculating Phase is given below. In *Figure 10B-I,* "t" and "e" refer to time stamps and event stamps respectively. Time and event stamps must have been corrected for offset and counter rollover as discussed above before performing calculations.

Phase = (fractional phase difference) + (whole number of cycles of phase difference)

$$\textbf{Phase A rel B}_i = \left\{ \frac{t_{2i+2} - t_{2i+1}}{\dfrac{t_{2i+2} - t_{2i}}{e_{2i+2} - e_{2i}}} + \left[(e_{2i+1} - e_1) - (e_{2i+2} - e_2) \right] \right\} \times 360^\circ$$

$$\textbf{Phase B rel A}_i = \left\{ \frac{t_{2i+1} - t_{2i+2}}{\dfrac{t_{2i+1} - t_{2i-1}}{e_{2i+1} - e_{2i-1}}} + \left[(e_{2i+2} - e_2) - (e_{2i+1} - e_1) \right] \right\} \times 360^\circ$$

Format 10C

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
PRF, Frequency, PRI, Period	Dual, Ratio, Sum, and Difference of A, B, and C	ON/OFF	Automatic Interval Sampling Edge Sampling Edge/Interval

How Signals are Converted to Binary Data

Example: Frequency A and B | Automatic

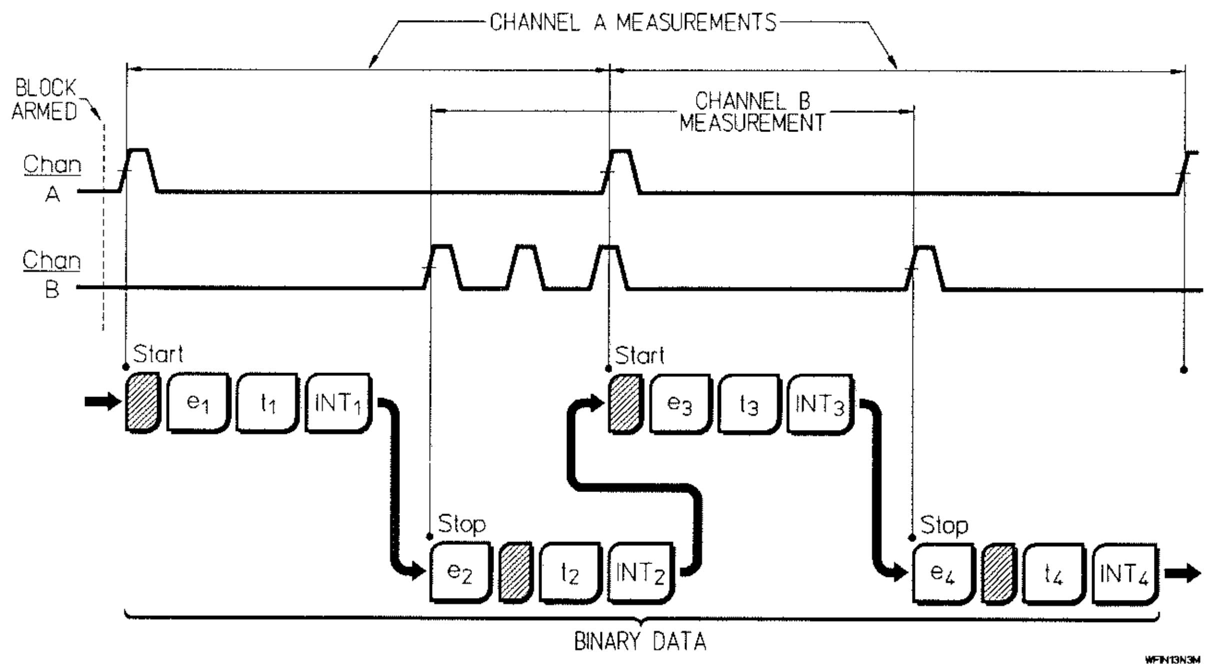


Figure 10C-A. Frequency Measurements on Channels A and B with Automatic Arming Mode Showing Corresponding Binary Data Output

Producing Results

Two-channel Frequency and Period results are generated by extracting two discrete sets of single-channel data from the data stream shown above.

Results are generated for the start channel by calculating both elapsed time and elapsed events between odd-numbered samples (start samples), then taking the appropriate quotient. The formulas are:

$$\text{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\text{PRI or Period} = \frac{\text{elapsed time}}{\text{elapsed events}}$$

Processing for the stop channel is almost identical. Results are generated for the stop channel by using even-numbered samples (stop samples).

The measurement choice may require that these results be combined (for example Frequency A + B). If so, results are combined algebraically, measurement-by-measurement.

EXPANDED DATA

Expanded Data for PRF, Frequency, PRI, and Period measurements is Gate Time. Gate Times for the start channel are generated by calculating elapsed time between odd-numbered samples (start samples). Gate Times for the stop channel are generated by calculating elapsed time between even-numbered samples (stop samples).

Binary Data Output

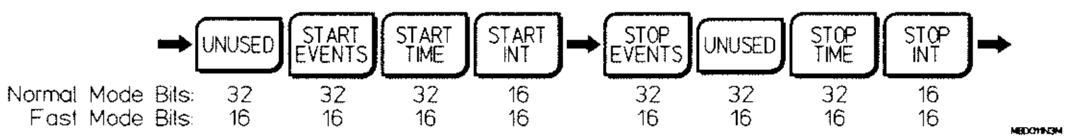


Figure 10C-B. Format 10C Binary Output

Format 10C binary output is shown in *Figure 10C-B* above. Each measurement consists of two samples: start and stop. Input channels associated with start and stop data must be identified. See *Technical Note 4* for information on how to determine the correct start and stop channels.

Each pair of samples generates the binary data defined below:

UNUSED FIELD
START CHANNEL EVENT: binary event count on the start channel.
Example: Channel B events for Frequency B and C measurement.
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

STOP CHANNEL EVENT: binary event count on the stop channel.
Example: Channel C events for Frequency B and C measurement.
UNUSED FIELD
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

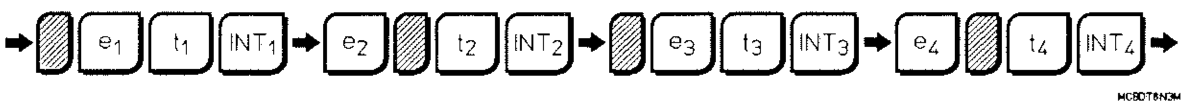


Figure 10C-C. Format 10C Binary Output

The binary data stream shown above was derived from the signals shown in *Figure 10C-A.* To produce valid event stamps, your program must:

1. Correct the binary start channel data for event counter rollovers.
2. Correct the binary stop channel data for event counter rollovers.
3. If measurement was made on Channel C, multiply the event stamp data by four.

To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start channel and the stop channel.

PRODUCING VALID EVENT STAMPS

Correcting Start Channel Event Data for Counter Rollovers

Start channel event counts increase monotonically unless a counter rollover has occurred. Note that start channel event counts occur every other sample. Monotonic increase should be expected for data e1, e3, e5, etc. (see *Figure 10C-A*). An overflow-processing routine must scan the start event data for any start event datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that start datum, and to all subsequent start data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1*.

Correcting Stop Channel Event Data for Counter Rollovers

Correction of stop channel data is virtually identical to that performed for start channel data. Successive stop channel event counts increase monotonically unless a counter rollover has occurred. Note that stop channel event counts are provided with every other sample. Monotonic increase should be expected for e2, e4, e6, etc. (see *Figure 10C-A*). An overflow-processing routine must scan the stop event data for any stop event datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that stop datum, and to all subsequent stop data, then continue with the scan. Event Overflow Correction values are given in *Technical Note 1.*

Channel C Correction

When making measurements on Channel C, the event counter sees only every fourth input event. To correct for this, multiply each event datum by four. This must be done after correcting for counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data For Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. The start sample will always be collected before the stop sample. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum, and to all subsequent data, then continue with the scan. An example routine is given in *Technical Note 3.* Time Overflow Correction values are given in *Technical Note 1.*

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between Start and Stop Channels.

There are several possible signal paths running from signal input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results

PRF, Frequency, PRI, and Period results are generated for each channel by calculating elapsed events and elapsed time between appropriate samples.

Results are generated for the start channel by using the odd-numbered samples, while results are generated for the stop channel by using even-numbered samples.

$$\textbf{Start channel PRF}_i \textbf{ or Frequency}_i = \frac{\text{start event stamp}_{i+1} - \text{start event stamp}_i}{\text{start time stamp}_{i+1} - \text{start time stamp}_i}$$

$$\textbf{Stop channel PRF}_i \textbf{ or Frequency}_i = \frac{\text{stop event stamp}_{i+1} - \text{stop event stamp}_i}{\text{corrected stop time stamp}_{i+1} - \text{corrected stop time stamp}_i}$$

$$\textbf{Start channel PRI}_i \textbf{ or Period}_i = \frac{\text{start time stamp}_{i+1} - \text{start time stamp}_i}{\text{start event stamp}_{i+1} - \text{start event stamp}_i}$$

$$\textbf{Stop channel PRI}_i \textbf{ or Period}_i = \frac{\text{corrected stop time stamp}_{i+1} - \text{corrected stop time stamp}_i}{\text{stop event stamp}_{i+1} - \text{stop event stamp}_i}$$

If two-channel measurements are chosen (A/B, A + B, B – A, etc.) the results computed above must be combined algebraically, measurement-by-measurement, to obtain results.

For example:

Frequency A + B$_i$ = Channel A Frequency$_i$ + Channel B Frequency$_i$

Expanded Data for Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating the elapsed time between appropriate time stamps. Start channel gate times are generated by using the odd-numbered time stamps. Stop channel gate times are generated by using the even-numbered time stamps. The formula is:

Gate Time$_i$ = time stamp$_{i+1}$ – time stamp$_i$

Format 10D

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
PRF, Frequency, PRI, Period	A, B, C	ON/OFF	Time/Time Event/Event

How Signals are Converted to Binary Data

There are two possible binary output configurations for this measurement. By allowing both, the full capabilities of these complex arming modes are supported. In these arming modes, the start sample can either precede or follow the stop sample. If the start sample precedes the stop sample, the binary data is delivered as shown in *Figure 10D-A*. If the stop sample precedes the start sample, the binary data is delivered as shown in *Figure 10D-B*. Note that the variation takes place measurement-by-measurement: the first pair of samples might resemble *Figure 10D-A* while the second pair of samples might resemble *Figure 10D-B*.

Example 1: Frequency A | Event/Event arming, start precedes stop; start armed after 3 Channel A events, stop armed after 2 Channel B events.

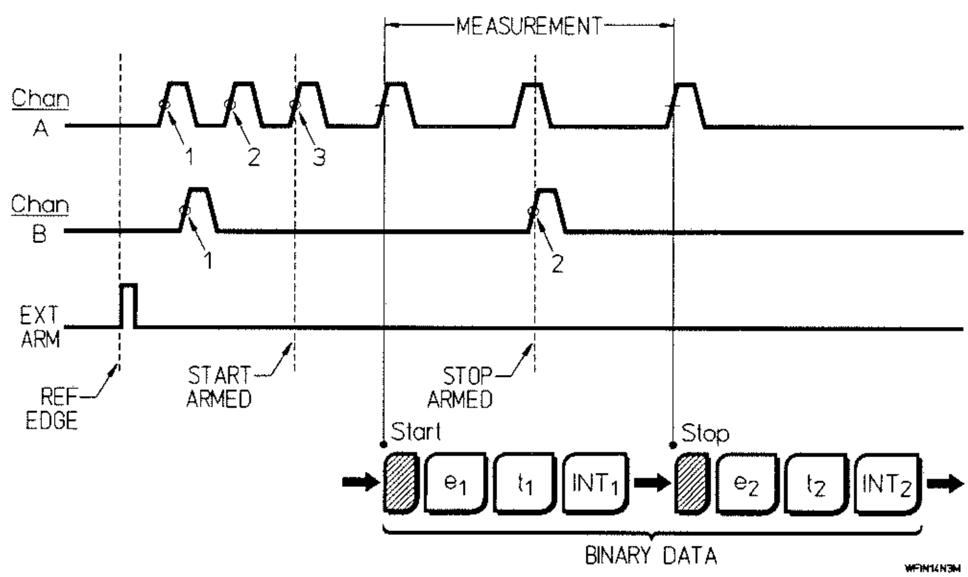


Figure 10D-A. Frequency Measurement on Channel A with Event/Event Arming Mode Showing Corresponding Binary Output.

Example 2: Frequency A | Event/Event arming, stop precedes start; start armed after 3 Channel A events, stop armed after 2 Channel B events.

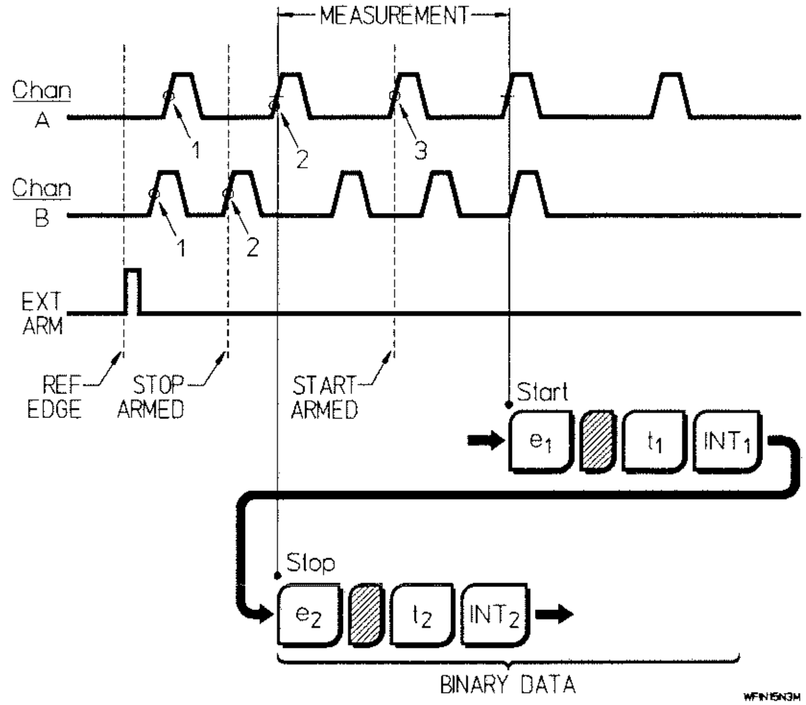


Figure 10D-B. Frequency Measurement on Channel A with Event/Event Arming Mode Showing Corresponding Binary Output

Producing Results

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between start and stop, then taking the appropriate quotient. The formulas are:

$$\textbf{PRF or Frequency} = \frac{\text{elapsed events}}{\text{elapsed time}}$$

$$\textbf{PRI or Period} = \frac{\text{elpased time}}{\text{elapsed events}}$$

Note that if the stop sample precedes the start sample, the values for both elapsed time and elapsed events will be negative, but the final result will be be a positive quantity.

EXPANDED DATA

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between start and stop samples. Note that if the stop sample precedes the start sample, the Measurement Gate Time calculation will be negative unless a provision is made in your algorithm for correction.

Binary Data Output

In this format one of two patterns will occur. If start precedes stop:

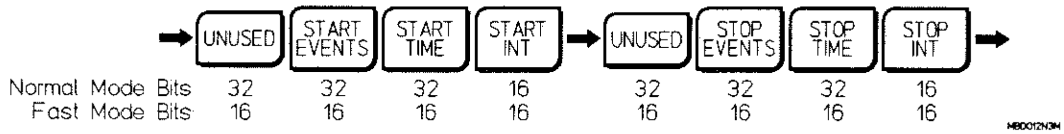


Figure 10D-C. Format 10D Binary Output: Start Before Stop

The second pattern occurs if stop precedes start:

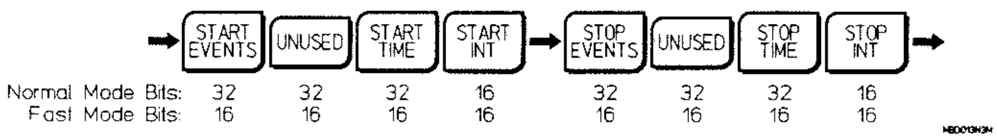


Figure 10D-D. Format 10D Binary Output: Stop Before Start

The two possible formats for binary output are shown above. Each measurement consists of two samples: start and stop.

If start precedes stop, use the format shown in *Figure 10D-C.* Each pair of samples, (each measurement), generates the set of binary data defined below:

UNUSED FIELD
START CHANNEL EVENT: binary event count.
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

UNUSED FIELD
STOP CHANNEL EVENT: binary event count.
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

If stop precedes start, use the format shown in *Figure 10D-D.* Each pair of samples, (each measurement), generates the set of binary data defined below:

START CHANNEL EVENT: binary event count.
UNUSED FIELD
TIME: start sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

STOP CHANNEL EVENT: binary event count.
UNUSED FIELD
TIME: stop sample time of occurrence.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Block Start, Inhibit.

Converting Binary Data to Event Stamps and Time Stamps

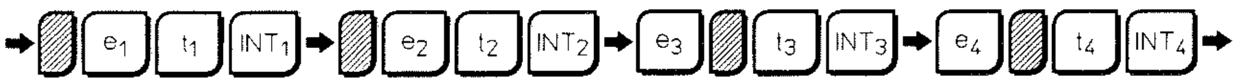


Figure 10D-E. Format 10D Binary Output

The binary data stream was derived from the signal shown in *Figure 10D-A* followed by the signal in *Figure 10D-B*. To produce valid time stamps, your program must:

1. Correct the binary time data for counter rollovers.

2. Identify from the binary time data whether the start sample or stop sample is first in each pair.

3. Incorporate interpolator data.

4. Correct for differences in electrical path length between the start and the stop channel.

To produce valid event stamps, your program must:

1. Identify the binary output pattern for each pair of samples.

2. Correct the binary data for counter rollovers.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Since this function will allow the stop sample to be collected before the start sample, time data does not increase monotonically. When start and stop are reversed, the stop time count is (correctly) smaller than the start time count. Your overflow processing routine must distinguish between start-stop reversals and counter rollovers. Overflow processing routines must correct the data only when a counter rollover is detected.

Correction for counter rollovers can be accomplished in two scans of the time data. The two scan processes are described below. First, a scan which corrects for rollovers between start and stop, then a scan which corrects for rollovers between measurements (start-stop pairs).

Correcting for Rollovers Between Start and Stop

Counter rollovers which occur between start and stop are detected when subtraction of start data from stop data produces an out-of-range result. If a subtraction produces a number greater than +Maximum, leave the stop datum uncorrected and add the appropriate Time Overflow Correction to the start datum. Make the same correction to all subsequent data (both start and stop). Following the corrections, continue subtracting start data from stop data to detect further rollovers.

If a subtraction produces a number smaller than –Maximum, leave the start datum uncorrected, and add the appropriate Time Overflow Correction to the stop datum. Make the same correction to all subsequent data (both start and stop). Following the corrections, resume the scan at the next start datum. Time Overflow Correction values are given in *Technical Note 1*. Values for +Maximum and –Maximum are given in *Technical Note* 2.

Correcting for Rollovers Between Measurements

Once the first scan is complete, a second scanning routine detects and corrects for counter rollovers which occur between measurements (start-stop pairs). The first scan leaves the time data in this condition: the smallest time value in each pair should be greater than the largest time value in the immediately preceding pair (unless a counter rollover has occurred between pairs). The second scan identifies the smaller time value in each pair, called "true start", then identifies the larger time value in the preceding pair, that pair's "true stop". If true stop is greater than true start, a counter rollover has occurred between pairs. The appropriate Time Overflow Correction should be added to all data subsequent to true stop. Following the corrections, resume the scan at the next pair. An example routine is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels

There are several possible signal paths running from signal input correctors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application. For the stop channel time stamp:

corrected stop time stamp = stop time stamp + offset

PRODUCING VALID EVENT STAMPS

An overflow-detection routine must scan the start-stop pairs of event data. For each pair, the event data should increase from the first datum collected to the second datum collected — unless a counter rollover occurred. For example, when the start sample is known to precede the stop sample, the start event count will be smaller than the stop event count — unless a rollover has occurred. Similarly, when the stop sample precedes the start sample, the stop sample event count will be smaller than the start event count — unless a counter rollover has occurred. Note that the event counters are always reset between pairs, so there will be no continuity in event counts between pairs.

The overflow detection routine should examine the last-collected datum in each pair. If the last-collected datum is smaller than the first-collected datum, the routine must add the appropriate Event Overflow Correction value to the last-collected datum. Then the next pair is examined, and the process repeats. Event Overflow Correction values are given in *Technical Note 1.*

Generating Final Function Results

PRF, Frequency, PRI, and Period results are generated by calculating both elapsed time and elapsed events between start and stop samples, then taking the appropriate quotient. The formula is:

$$\textbf{PRF}_i\textbf{ or Frequency results}_i = \frac{\text{stop event stamp}_i - \text{start event stamp}_i}{\text{corrected stop time stamp}_i - \text{start time stamp}_i}$$

$$\textbf{PRI}_i\textbf{ or Period result}_i = \frac{\text{corrected stop time stamp}_i - \text{start time stamp}_i}{\text{stop event stamp}_i - \text{start event stamp}_i}$$

EXPANDED DATA

Expanded Data for PRF, Frequency, PRI, and Period measurements is Measurement Gate Time. Measurement Gate Time is generated by calculating the absolute value of elapsed time between start and stop samples. The formula is:

$$\textbf{Gate Time}_i = \text{ABS}\,[\text{stop time stamp}_i - \text{start time stamp}_i]$$

Format 11

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Totalize	A, B	ON/OFF	Interval Sampling Edge Sampling Edge/Interval Edge/Edge

How Signals are Converted to Binary Data

Example: Totalize Channel A | Interval Sampling

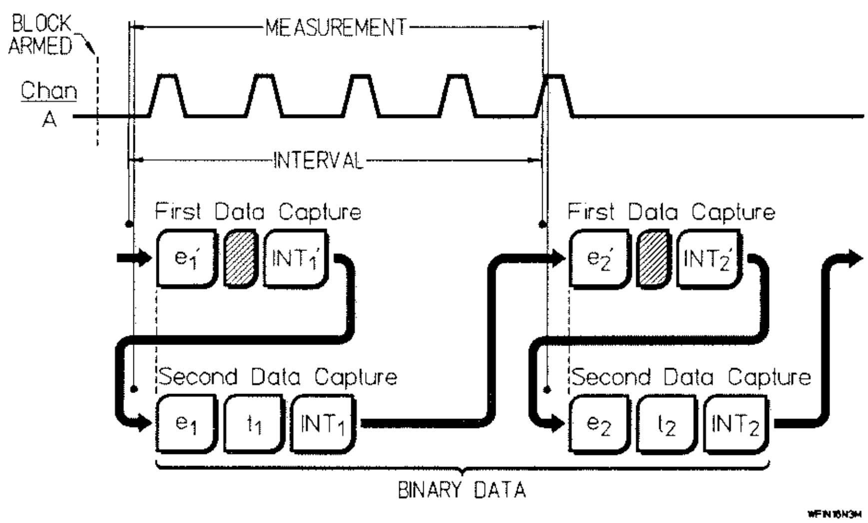


Figure 11-A. Totalize Measurement on Channel A with Interval Sampling Arming Mode Showing Corresponding Binary Output

Producing Results

Totalize results are generated by calculating elapsed events between contiguous valid samples. As diagrammed above, the instrument makes two attempts to capture valid data for each sample. Your program must use the status bits provided to merge the data from each pair of attempts into a single valid sample; then process the set of valid samples.

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Binary Data Output

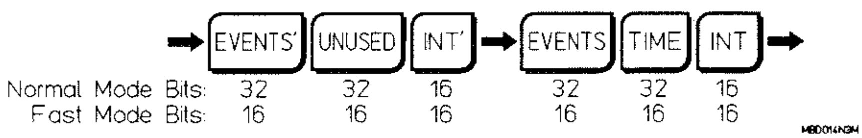


Figure 11-B. Format 11 Binary Output

Format 11 binary output is shown in *Figure 11-B* above. Each sample generates the binary data described below:

EVENT': instrument's first attempt at capture of a valid event datum.
UNUSED FIELD
INT': status bits only.

Valid status bits: Block Start, Inhibit.

EVENT: instrument's second attempt at capture of a valid event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13).

Converting Binary Data to Event Stamps and Time Stamps

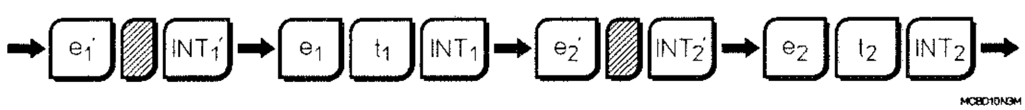


Figure 11-C. Format 11 Binary Output

The binary data stream was derived from the signal shown in *Figure 11-A*. To produce valid event stamps:

1. Identify each datum which represents a valid event count (e' or e).

2. Correct the valid event counts for counter rollovers.

To produce valid time stamps:

1. Correct the binary time data for counter rollovers.

2. Incorporate the interpolator data.

PRODUCING VALID EVENT STAMPS

Identifying the Datum Which Represents the Valid Event Count

The Totalize Status 1 Bit, "TS1", is bit 13 of the Interpolator/Status Field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). TS1 identifies the datum which represents the valid event count.

If TS1 = 0, use e′, the instrument's first attempt at valid event data capture. This is the valid binary event count.

If TS1 = 1, use e, the instrument's second attempt at valid event data capture. Subtract 1 from the integer represented by e. The subtraction results in a valid binary event count.

Correcting Event Data for Counter Rollovers

Valid event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the set of valid event data, seeking any event datum which is smaller than its immediate predecessor. On detection of such an event datum, the routine must add the appropriate Event Overflow Correction value to that datum and all subsequent data then continue the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum smaller than its immediate predecessor. On detection of such datum, the routine must add the appropriate Time Overflow Correction value to that datum and to all subsequent time data, then continue the scan. An example is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp: = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Generating Final Function Results from Time Stamps and Event Stamps

Totalize results are generated by calculating elapsed events between valid event stamps:

Totalize result$_i$ = event stamp$_{i+1}$ – event stamp$_i$

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Times are generated by calculating the elapsed time between valid time stamps:

Gate Time$_i$ = time stamp$_{i+1}$ – time stamp$_i$

Format 12

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Totalize	A, B	ON/OFF	Time/Interval

How Signals are Converted to Binary Data

Example: Totalize A | Time/Interval Arming

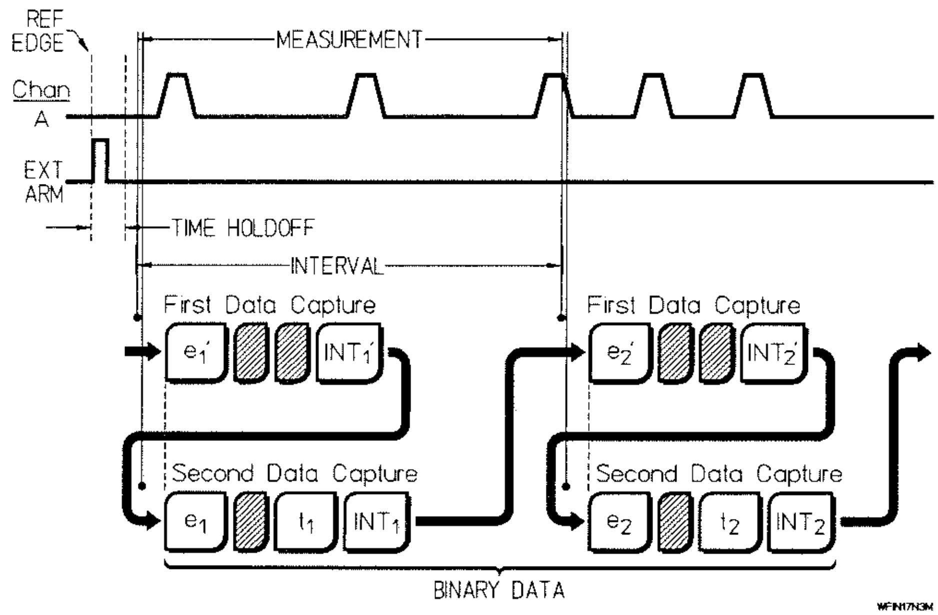


Fig 12-A. Totalize Measurement on Channel A with Time/Interval Arming Mode Showing Corresponding Binary Output.

Producing Results

Totalize results are generated by calculating elapsed events between contiguous valid samples. As diagrammed above, the instrument makes two attempts to capture valid data for each sample. Your program must use the status bits provided to merge the data from each pair of attempts into a single valid sample; then process the set of valid samples.

EXPANDED DATA

Expanded Data for Totalize measurement is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Binary Data Output

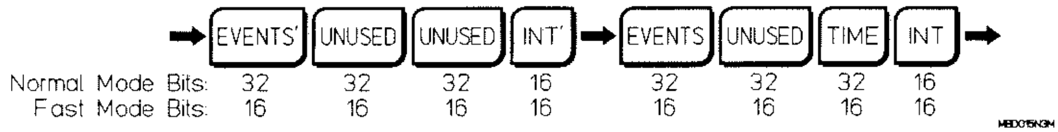


Figure 12-B. Format 12 Binary Output

Format 12 binary output is shown in *Figure 12-B* above. Each sample generates the binary data described below:

EVENT′: instrument's first attempt at capture of a valid event datum.
UNUSED FIELD
UNUSED FIELD
INT′: status bits only

Valid status bits: Block Start, Inhibit.

EVENT: instrument's second attempt at capture of a valid event datum.
UNUSED FIELD
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 2 (bit 14).

Converting Binary Data to Event Stamps and Time Stamps

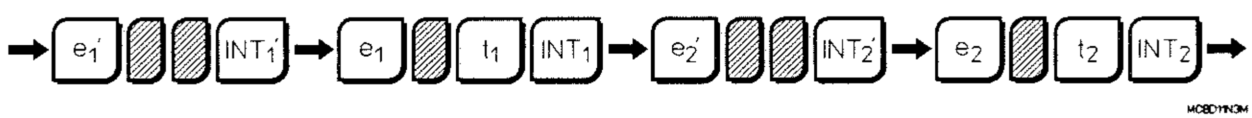


Figure 12-C. Format 12 Binary Output

The binary data stream was derived from the signal shown in *Figure 11-A*. To produce valid event stamps:

1. Identify each datum which represents a valid event count (e′ or e).

2. Correct the valid event counts for counter rollovers.

To produce valid time stamps:

1. Correct the binary time data for counter rollovers.

2. Incorporate the interpolator data.

PRODUCING VALID EVENT STAMPS

Identifying the Datum Which Represents the Valid Event Count

The Totalize Status 2 Bit, "TS2", is bit 14 of the Interpolator/Status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). TS2 identifies the datum which represents the valid event count.

If TS2 = 0, use e′, the instrument's first attempt at valid event data capture. This is the valid binary event count.

If TS2 = 1, use e, the instrument's second attempt at valid event data capture. Subtract 1 from the integer represented by e. The subtraction results in the valid binary event count.

Correcting Event Data for Counter Rollovers

Valid event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and all subsequent data, then continue the scan. An example is given given in *Technical Note* 3. Event Overflow Correction values are given in *Technical Note 1.*

PRODUCING VALID TIME STAMPS

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent time data, then continue the scan. An example is given in *Technical Note* 3. Time Overflow Correction values are given in *Technical Note 1.*

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time and interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Generating Final Function Results From Time Stamps and Event Stamps

Totalize results are generated by calculating elapsed events between valid event stamps:

Totalize result$_i$ = event stamp$_{i+1}$ – event stamp$_i$

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Times are generated by calculating the elapsed time between valid time stamps:

Gate Time$_i$ = time stamp$_{i+1}$ – time stamp$_i$

Format 13

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Totalize	Dual, Ratio, Sum and Difference of A and B	ON/OFF	Interval Sampling Edge Sampling Edge/Interval Edge/Edge

How Signals are Converted to Binary Data

Example: Totalize A and B | Interval Sampling

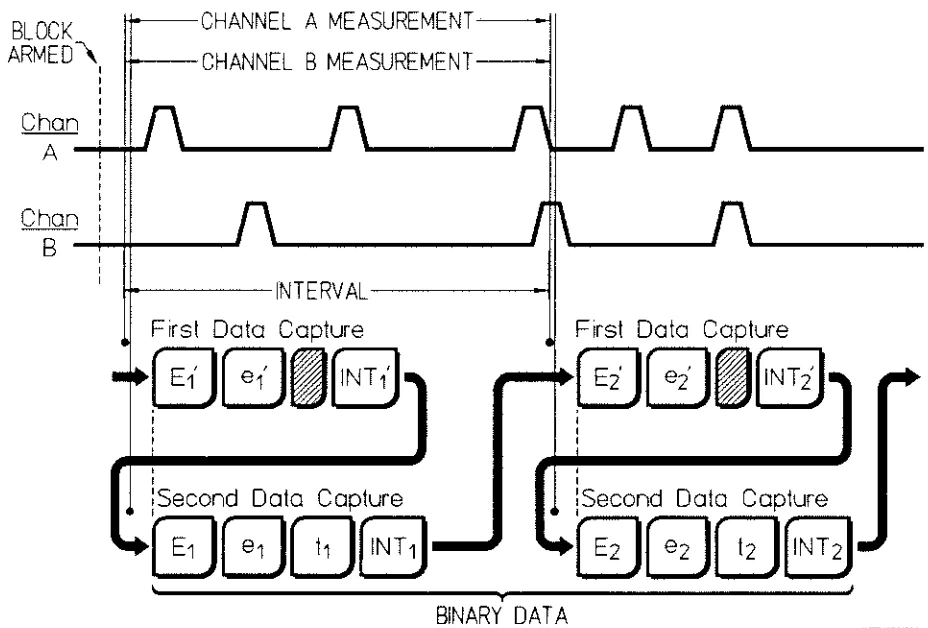


Figure 13-A. Totalize Measurement on Channel A and B with Interval Sampling Arming Mode Showing Corresponding Binary Output

Producing Results

Totalize results are generated by calculating elapsed events between contiguous valid samples. As diagrammed above, the instrument makes two attempts to capture valid data for each sample. Your program must use the status bits provided to merge the data from each pair of attempts into a single valid sample; then process the set of valid samples.

EXPANDED DATA

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between samples.

Binary Data Output

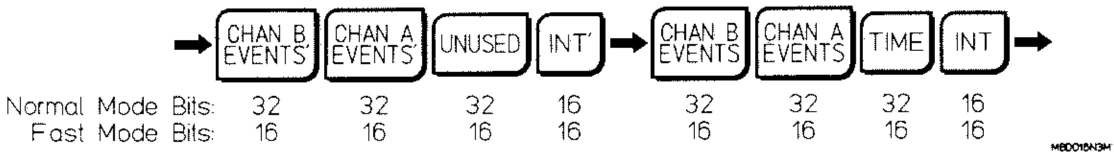


Figure 13-B. Format 13 Binary Output

Format 13 binary output is shown in *Figure 13-B* above. Each sample generates the binary data described below:

EVENT': instrument's first attempt at capture of a valid Channel B event datum.
EVENT': instrument's first attempt at capture of a valid Channel A event datum.
UNUSED FIELD
INT': status data only.

Valid status bits: Block Start, Inhibit.

EVENT: instrument's second attempt at capture of a valid Channel B event datum.
EVENT: instrument's second attempt at capture of a valid Channel A event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13), Totalize Status 2 (bit 14).

Converting Binary Data to Event Stamps and Time Stamps

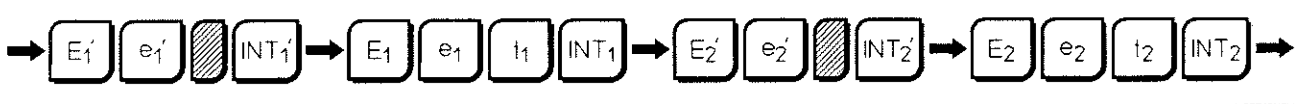


Figure 13-C. Format 13 Binary Output

The binary data stream was derived from the signals shown in *Figure 13-A*. To produce valid event stamps:

1. Identify each datum which represents a valid Channel A event count (e' or e).

2. Correct the valid Channel A event counts for counter rollovers.

3. Identify each datum which represents a valid Channel B event count (E' or E).

4. Correct the valid Channel B event counts for counter rollovers.

To produce valid time stamps:

1. Correct the binary time data for counter rollovers.
2. Incorporate the interpolator data.

PRODUCING VALID EVENT STAMPS

Identifying the Datum Which Represents the Valid Channel A Event Count

The Totalize Status 1 Bit, "TS1", is bit 13 of the interpolator/status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). During two-channel Totalize measurements, TS1 identifies the datum which represents the valid Channel A event count.

If TS1 = 0, use e′, the instrument's first attempt at valid Channel A event data capture. This is the valid binary event count.

If TS1 = 1, use e, the instrument's second attempt at valid Channel A event data capture. Subtract 1 from the integer represented by e. The subtraction results in a valid binary event count.

Identifying the Datum Which Represents the Valid Channel B Event Count

The Totalize Status 2 Bit, "TS2", is bit 14 of the interpolator/status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). During two-channel Totalize measurements, TS2 identifies the datum which represents the valid Channel B event count.

If TS2 = 0, use E′, the instrument's first attempt at valid Channel B event capture. This is the valid binary event count.

If TS2 = 1, use E, the instrument's second attempt at valid Channel B event capture. Subtract 1 from the integer represented by E. The subtraction results in a valid binary event count.

Correcting Event Data for Counter Rollovers

Each sample should now include both a valid Channel A event datum and a valid Channel B event datum. The Channel A event data should be processed independent of Channel B event data. Consider each channel's event data as an independent data stream. Each set of data can now be processed for counter rollover.

Valid event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and all subsequent data, then continue the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

PRODUCING VALID TIME STAMPS

Time and interpolator data is delivered with the second data capture for each sample. To produce valid time stamps:

1. Correct the binary data for counter rollovers.

2. Incorporate interpolator data.

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and all subsequent time data, then continue the scan. An example is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Generating Final Function Results From Time Stamps and Event Stamps

Totalize results are generated for each channel by calculating elapsed events between valid event stamps:

Channel A $\textbf{result}_i$ = Channel A event stamp_{i+1} – Channel A event stamp_i

Channel B $\textbf{result}_i$ = Channel B event stamp_{i+1} – Channel B event stamp_i

The measurement may require that these independent results be combined (for example, A + B,). If this is required, results are combined algebraically, measurement-by-measurement. For example:

Totalize A + B $\textbf{result}_i$ = Channel A result_i + Channel B result_i

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Times are generated by calculating elapsed time between valid time stamps.

Gate $\textbf{Time}_i$ = time stamp_{i+1} – time stamp_i

Format 14

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Totalize	A, B	ON/OFF	Externally Gated

How Signals are Converted to Binary Data

Example: Totalize A | Externally Gated

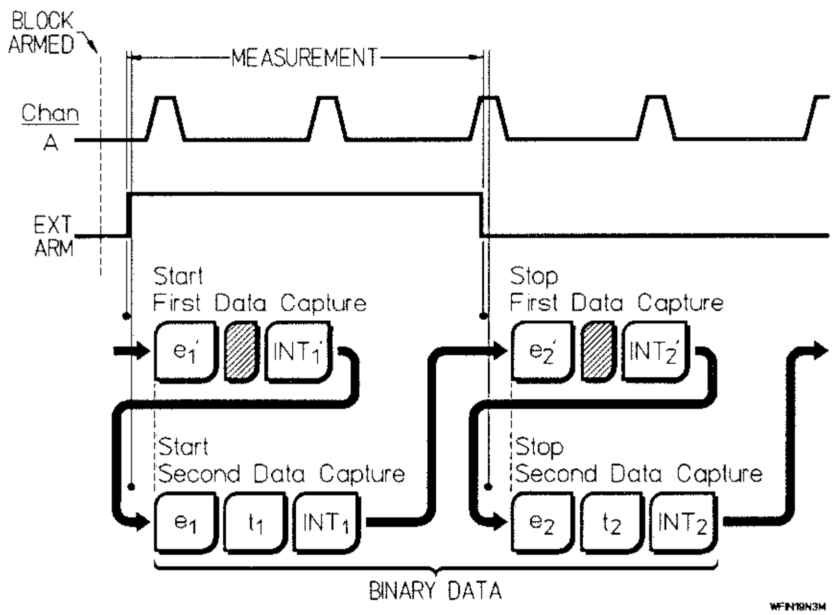


Figure 14-A. Totalize Measurement on Channel A with Externally Gated Arming Mode Showing Corresponding Binary Output

Producing Results

Totalize results are generated by calculating elapsed events between a valid start sample and a valid stop sample. As shown above in *Figure 14-A,* the instrument makes two attempts to capture valid data for each sample. Your program must use the status bits provided to merge the data from each pair of attempts into a single valid sample. Then processes the set of valid samples.

EXPANDED DATA

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between start and stop samples.

Binary Data Output

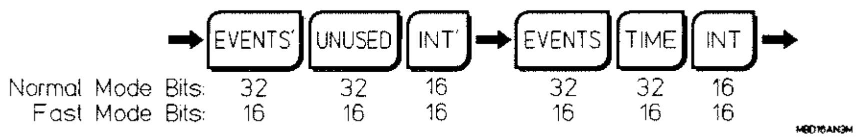


Figure 14-B. Format 14 Binary Output

Format 14 binary output is shown in *Figure 14-B* above. Each sample generates the binary data described below:

START EVENT': instrument's first attempt at capture of a valid event datum.
UNUSED FIELD
INT': status bits only.

Valid status bits: Block Start, Inhibit.

START EVENT: instrument's second attempt at capture of a valid event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13).

STOP EVENT': instrument's first attempt at capture of a valid event datum.
UNUSED FIELD
INT': status bits only.

Valid status bits: Block Start, Inhibit.

STOP EVENT: instrument's second attempt at capture of a valid event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13).

Converting Binary Data to Event Stamps and Time Stamps

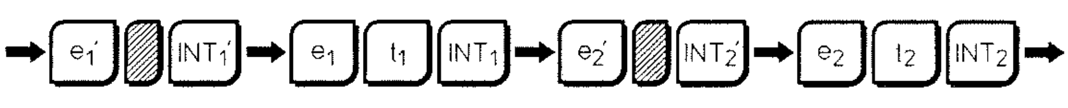


Figure 14-C. Format 14 Binary Output

The binary data stream was derived from the signal shown in *Figure 14-A*. To produce valid event stamps:

1. Identify each datum which represents a valid event count (e′ or e).

2. Correct the valid event count for counter rollovers.

To produce valid time stamps:

1. Correct the binary time data for counter rollovers.

2. Incorporate the interpolator data.

3. Correct for differences in electrical path length between the start and stop channels.

PRODUCING VALID EVENT STAMPS

Identifying the Datum Which Represents the Valid Event Count

The Totalize Status 1 Bit, "TS1", is bit 13 of the Interpolator/Status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). TS1 identifies the datum which represents the valid event count.

If TS1 = 0, use e′, the instrument's first attempt at valid event data capture. This is the valid binary event count.

If TS1 = 1, use e, the instrument's second attempt at valid event data capture. Subtract 1 from the integer represented by e. The subtraction results in a valid binary event count.

Correcting Event Data for Counter Rollovers

Valid event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and all subsequent data, then continue the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

PRODUCING VALID TIME STAMPS

Correcting Binary Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to that datum and to all subsequent time data, then continue the scan. An example is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp: = [time datum] $\times$[2 ns] – [interpolator datum] $\times$[0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels.

There are several possible signal paths running from input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters, including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application.

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps and Event Stamps

Totalize results are generated by calculating elapsed events between valid start event stamps and valid stop event stamps:

Totalize result$_i$ = stop event stamp$_i$ – start event stamp$_i$

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Times are generated by calculating elapsed time between valid start time stamps and valid stop time stamps.

Gate Time$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

Format 15

MEASUREMENT	CHANNEL(S)	EXPANDED DATA	ARMING MODES
Totalize	Dual, Ratio, Sum and Difference of A and B	ON/OFF	Externally Gated

How Signals are Converted to Binary Data

Example: Totalize A and B | Externally Gated

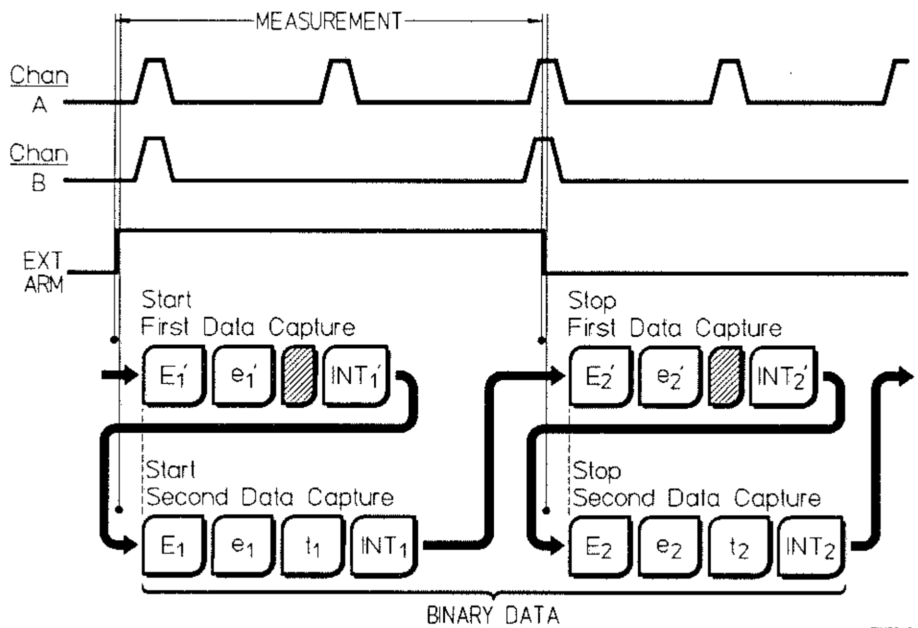


Figure 15-A. Totalize Measurement on Channels A and B with Externally Gated Arming Mode, Showing Corresponding Binary Output

Producing Results

Totalize results are generated by calculating elapsed events between a valid start sample and a valid stop sample. As shown above in *Figure 15-A,* the HP 5373A makes two attempts to capture valid data for each sample. Your program must use the status bits provided to merge the data from each pair of attempts into a single valid sample. Then process the set of valid samples.

EXPANDED DATA

Expanded Data for Totalize measurements is Measurement Gate Time. Gate Time is generated by calculating elapsed time between start and stop samples.

Binary Data Output

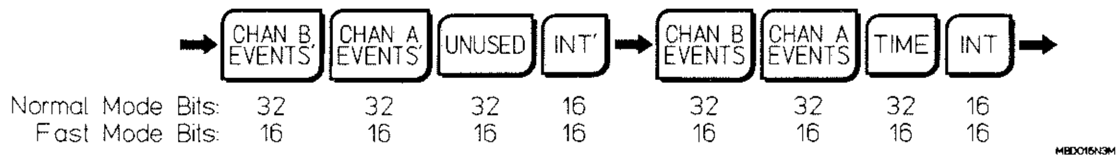


Figure 15-B. Format 15 Binary Output

Format 15 binary output is shown in *Figure 15-B* above. Each sample generates the binary data described below:

START EVENT': instrument's first attempt at capture of a valid Channel B event datum.
START EVENT': instrument's first attempt at capture of a valid Channel A event datum.
UNUSED FIELD
INT': status bits only.

Valid status bits: Block Start, Inhibit.

START EVENT: instrument's second attempt at capture of a valid Channel B event datum.
START EVENT: instrument's second attempt at capture of a valid Channel A event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13), Totalize Status 2 (bit 14).

STOP EVENT': instrument's first attempt at capture of a valid Channel B event datum.
STOP EVENT': instrument's first attempt at capture of a valid Channel A event datum.
UNUSED FIELD
INT': status bits only.

Valid status bits: Block Start, Inhibit.

STOP EVENT: instrument's second attempt at capture of a valid Channel B event datum.
STOP EVENT: instrument's second attempt at capture of a valid Channel A event datum.
TIME: time of occurrence for sample.
INT: interpolator data: provided to increase time resolution.

Valid status bits: Totalize Status 1 (bit 13), Totalize Status 2 (bit 14).

Converting Binary Data to Event Stamps and Time Stamps

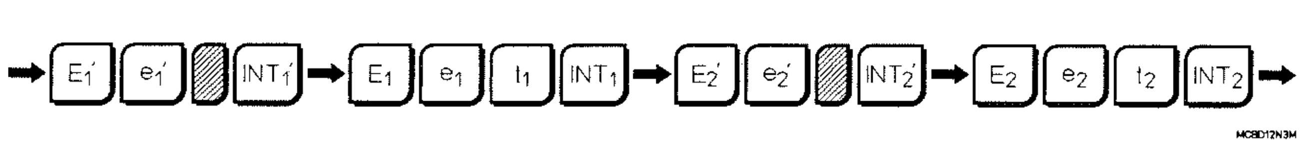


Figure 15-C. Format 15 Binary Output

The binary data stream was derived from the signals shown in *Figure 15-A*. To produce valid event stamps:

1. Identify each datum which represents a valid Channel A event count (e′ or e).
2. Correct the valid Channel A event counts for counter rollovers.
3. Identify each datum which represents a valid Channel B event count (E′ or E).
4. Correct the valid Channel B event counts for counter rollovers.

PRODUCING VALID EVENT STAMPS

Identifying the Datum Which Represents the Valid Channel A Event Count

The Totalize Status 1 Bit, "TS1", is bit 13 of the Interpolator/Status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). During two-channel Totalize measurements, TS1 identifies the datum which represents the valid Channel A event count.

If TS1 = 0, use e′, the instrument's first attempt at valid Channel A event data capture. This is the valid binary event count.

If TS1 = 1, use e, the instrument's second attempt at valid Channel A event capture. Subtract 1 from the integer represented by e. The subtraction results in a valid binary event count.

Identifying the Datum Which Represents the Valid Channel B Event Count

The Totalize Status 2 Bit, "TS2", is bit 14 of the interpolator/status field (see *Interpolator/Status Field* under *Special Topics* at the front of this chapter). During two-channel Totalize measurements, TS2 identifies the datum which represents the valid Channel B event count.

If TS2 = 0, use E′, the instrument's first attempt at valid Channel B event data capture. This is the valid event count.

If TS2 = 1, use E, the instrument's second attempt at valid Channel B event capture. Subtract 1 from the integer represented by E. The subtraction results in a valid binary event count.

Correcting Event Data for Counter Rollovers

Each sample should now include both a valid Channel A event datum and a valid Channel B event datum. The Channel A event data should be processed independent of the Channel B event data. Consider each channel's event data as an independent data stream. Each set of data can now be processed for counter rollover.

Valid event counts increase monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the event data for any datum which is smaller than its immediate predecessor. On detection of of such a datum, the routine must add the appropriate Event Overflow Correction value to that datum and all event subsequent data, then continue the scan. An example is given in *Technical Note 3*. Event Overflow Correction values are given in *Technical Note 1*.

PRODUCING VALID TIME STAMPS

Time and interpolator data is delivered with the second data capture for each sample. To produce valid time stamps, your program must:

1. Correct the binary data for counter rollovers.
2. Incorporate interpolator data.
3. Correct for differences in electrical path length between the start and stop channels.

Correcting Time Data for Counter Rollovers

Binary time data increases monotonically unless a counter rollover has occurred. An overflow-processing routine must scan the time data for any datum which is smaller than its immediate predecessor. On detection of such a datum, the routine must add the appropriate Time Overflow Correction value to the datum and all subsequent time data, then continue the scan. An example is given in *Technical Note 3*. Time Overflow Correction values are given in *Technical Note 1*.

Incorporating Interpolator Data

The algorithm for producing time stamps from binary time data and binary interpolator data is:

time stamp = [time datum] × [2 ns] – [interpolator datum] × [0.1 ns]

Correcting for Differences in Electrical Path Length Between the Start and Stop Channels.

There are several possible signal paths running from input connectors on the HP 5373A front panel to measurement hardware inside the instrument. Signal routing depends on the selection of several parameters including Common Input Mode and Attenuation. To adjust for this variation, add a correction value, called an offset, to each stop channel time stamp. Start channel time stamps require no correction. Use *Technical Note 4* to determine the correct value of offset for your application.

corrected stop time stamp = stop time stamp + offset

Generating Final Function Results From Time Stamps and Event Stamps

Totalize results are generated for each channel by calculating elapsed events between valid start event stamps and valid stop event stamps:

Channel A result$_i$ = stop event stamp$_i$ – start event stamp$_i$

Channel B result$_i$ = stop event stamp$_i$ – start event stamp$_i$

The measurement may require that these independent results be combined (for example, A + B). If this is required, results are combined algebraically, measurement-by-measurement. For example:

Totalize A + B result$_i$ = Channel A result$_i$ + Channel B result$_i$

Expanded Data for Totalize measurements is Measurement Gate Times. Gate Times are generated by calculating elapsed time between start time stamps and stop time stamps.

Gate Time$_i$ = corrected stop time stamp$_i$ – start time stamp$_i$

TECHNICAL NOTE 1

NOTE

Each format section contains algorithms which correct for counter rollover. These algorithms are simplified to more clearly describe the rollover correction process. When you write your data collection and reduction program, you may wish to devise more efficient algorithms. For example, an overflow correction routine will execute faster if the accumulated number of rollovers is multiplied by the Overflow Correction value and added to each datum before it is tested for rollover.

Event Overflow Correction Values

	Normal Mode (32-bit data fields)	Fast Mode (16-bit data fields)
Required Overflow Correction for event data	Add 4,294,967,296 (2^{32}) to all event data after rollover.	Add 65,536 (2^{16}) to all event data after rollover.

Time Overflow Correction Values

	Normal Mode (32-bit data fields)	Fast Mode (16-bit data fields)
Required Overflow Correction for time data	Add 4,294,967,296 (2^{32}) to all time data after rollover.	Add 65,536 (2^{16}) to all time data after rollover.

TECHNICAL NOTE 2

+Maximum and –Maximum Values

	Normal Mode (32-bit data fields)	Fast Mode (16-bit data fields)
+Maximum value	2,147,483,647 ($2^{31} - 1$)	32,767 ($2^{15} - 1$)
–Maximum value	–2,147,483,647	–32,767

TECHNICAL NOTE 3

Rollover Processing Routines

This Technical Note contains example rollover processing routines for each Binary Format. The examples can be used as a guide to devising your own rollover processing routines to process rollovers in both event and time data.

A counter rollover occurs when the 32 bit time or event counter reaches its maximum count ($2^{32}-1$) and is incremented by one. At that point the counter returns to zero and begins counting until it again reaches ($2^{32}-1$) . Because the counters in the 5373A are freerunning, this can happen at any time in the measurement process. To convert time and event data into valid time stamps and event stamps your program must detect and compensate for counter rollovers. Time and event data should always increase. If a specific time or event datum is smaller than the preceding time or event datum, a counter rollover must have occurred. Counter rollover processing consists of finding each time or event datum that is less than its immediately preceding time or event datum, then adding (2^{32}) to all following time or event data.

NOTE

For ±TI measurements the above explanation does not apply. For information on how to process counter rollovers for ±Time Interval measurements see the examples for Format 4A processing.

NOTE

The following examples are designed to show the general program flow for processing counter rollovers. The code is not intended for direct use.

In all examples the binary data has been converted to decimal data.

Counter Rollover Processing:

FORMATS 1A, 2A

The rollover processing routine for time Data is:

N = the number of measurements
(There are N+1 samples for this format).

Data Structure Notes:

1. One dimensional array.

2. Time data stored in successive array locations in chronological order (i.e., order in which data was received).

```
FOR i = 1 TO N
  IF time(i+1) < time(i) THEN    !is next sample less than current sample
   FOR j = (i+1) TO (N+1)        !for all following samples
     time(j) = time(j) + 2^32    !add overflow value
   NEXT j
  ENDIF
NEXT i
```

The rollover processing routine for event data is:

N = the number of measurements
(There are N+1 samples for this format).

Data Structure Notes:

1. One dimensional array.

2. Time data stored in successive array locations in chronological order (i.e., order in which data was received).

```
FOR i = 1 TO N
  IF event(i+1) < event(i) THEN    !is next sample less than current sample
   FOR j = (i+1) TO (N+1)          !for all following samples
     event(j) = event(j) + 2^32    !add overflow value
   NEXT j
  ENDIF
NEXT i
```

NOTE

Event Data is not available for Format 1A.

FORMATS 1B, 2B, AND 3

The rollover processing routine for time data is:

N = the number of measurements
(There are N+1 samples for this format).

```
FOR i = 0 TO N                        !Time(0) corresponds to the time stamp
                                      !of the Block Arm
  IF time(i+1) < time(i) THEN         !is next sample less than current sample
   FOR j = (i+1) TO (N+1)             !for all following samples
     time(j) = time(j) + 2^32         !add overflow value
   NEXT j
  ENDIF
NEXT i
```

The rollover processing routine for event data is:

N = the number of measurements
(There are N+2 samples for this format).

```
FOR i = 1 TO N                          !Event(0) corresponds to the time stamp
                                        !of the Block Arm and is not valid
  IF event(i+1) < event(i) THEN         !is next sample less than current sample
   FOR j = (i+1) TO (N+1)               !for all following samples
     event(j) = event(j) + 2^32         !add overflow value
   NEXT j
  ENDIF
NEXT i
```

NOTE

Event Data is not available in Format 1B.

FORMAT 2A

See Format 1A

FORMAT 2B

See Format 1B

FORMAT 3

See Format 1B

FORMAT 4A

The rollover processing routine for time data is:

N = the number of measurements
(There are 2N samples for this format).

Data Structure Notes:

1. One dimensional array.
2. Odd indices reference Start Samples.
3. Even indices reference Stop Samples.

```
FOR i = 1 TO (2*N-1)
  IF time(i+1) < time(i) THEN   !is next sample less than current sample
   FOR j = (i+1) TO (2*N)        !for all following samples
     time(j) = time(j) + 2^32   !add overflow value
   NEXT j
  ENDIF
NEXT i
```

The rollover processing routine for event data is:

N = the number of measurements
(There are 2N samples for this format).

Data Sructure Notes:

1. One dimensional array.
2. Odd indices reference Start Samples.
3. Even indices reference Stop Samples.

```
FOR i = 1 TO (2*N-1)
  IF event(i+1) < event(i) THEN   !is next sample less than current sample
   FOR j = (i+1) TO (2*N)          !for all following samples
     event(j) = event(j) + 2^32   !add overflow value
   NEXT j
  ENDIF
NEXT i
```

NOTE

Event Data is not available for Format 4A.

FORMAT 4B FOR ±TIME INTERVAL MEASUREMENTS

Processing counter rollovers for ±TI measurements must be handled differently because negative time intervals are allowed (in other words a stop event is allowed to occur before a start event). It is important to understand that the start channel data is always physically transferred before the stop channel data. However, the time and event data from the start channel may have a larger value than the time and event data from the stop channel. While taking ±TI measurements, it is most convenient to process rollovers by grouping time data into start/stop pairs. Rollovers can occur between start/stop pairs, or within start/stop pairs. To check for these two rollover possibilities two scans of the data are necessary.

To correct for rollovers within start/stop pairs:

Because stop is allowed to happen before the start in ±TI measurements, the rollover processing routine must distinguish between counter rollovers and start-stop reversals. To do this, the processing routine must check the difference between the start time and the stop time within a start/stop pair. If the difference between start and stop is greater than $(2^{31}-1)$ then a rollover must have occurred. To account for a rollover within a start/stop pair follow the procedure listed below.

- If (stop-start) > $(2^{31}-1)$ then the stop must have come before the start and a rollover must have occurred between them. In this case, the processing routine must add 2^{32} to the start datum and all subsequent data (excluding the current stop datum).

- If (stop-start) < – (2^{31}–1) then the start must have come before the stop and a rollover must have occurred between them. In this case, the processing routine must add 2^{32} to the stop and all subsequent data (excluding the current start datum).

To correct for rollovers between start/stop pairs:

Once the first data scan is complete, a second routine detects and corrects for counter rollovers between start/stop pairs. The smallest time value in each pair should be greater than the largest time value in the preceding pair (unless a counter rollover has occurred between pairs). The second scan identifies the smallest time value in each pair, "True start", and identifies the largest time value in the preceding pair, "True stop". If true stop is greater than true start, a counter rollover has occurred between stop/start pairs. 2^{32} must be added to all data subsequent to true stop. The following example demonstrates this procedure.

The rollover processing routine for the time data is:

N = the number of desired results
There are 2N samples for this format.

This example uses a one-dimensional array to hold all the binary time data. Data in even indices, (2*i), is the start data. Data in the odd indices, (2*i-1), is the stop data.

```
FOR i = 1 TO (N)        !is there an overflow after start, and prior to stop
  IF (time(2*i) - time(2*i-1)) < -(2^31-1) THEN
   time(2*i) = time(2*i) + 2^32 !correct the stop
   FOR j = (2*i+1) TO (2*N)     !correct all subsequent data
     time(j) = time(j) + 2^32
   NEXT j
  ELSE
    IF (time(2*i) - time(2*i-1) > (2^31-1) THEN
      time(2*i-1) = time(2*i-1) + 2^32   !correct the start
     FOR j = (2*i+1) TO (2*N)           !correct all subsequent data
       time(j) = time(j) + 2^32
     NEXT j
   ENDIF
 ENDIF
NEXT i                       !is there an overflow between start/stop pairs
FOR i = 1 TO (N-1) !is stop less than start
  IF (time(2*i) < time(2*i-1)) THEN ! use time(2*i-1)
    IF (time(2*i+2) < time(2*i+1)) THEN ! use time(2*i+2)
       {compare smallest sample of second pair with greatest of first}
       {second pair stop with first pair start}
      IF time(2*i+2) < time(2*i-1) THEN !if less than, add overflow to all
        FOR j = (2*i+1) TO (2*N)        !the following samples
         time(j) = time(j) + 2^32
        NEXT j
      ENDIF
    ELSE !stop > or = to start, use time(2*i+1)
       {compare smallest sample of second pair with greatest of first}
       {second pair start with first pair start}
      IF time(2*i+1) < time(2*i-1) THEN !if less than, add overflow value
```

```
        FOR j = (2*i+1) TO (2*N)        !to all following samples
          time(j) = time(j) + 2^32
        NEXT j
       ENDIF
  ENDIF
  ELSE                                 !stop> or = to the start, use time(2*i)
   {check next pair for smaller sample, is stop less than start}
   IF (time(2*i+2) < time(2*i+1)) THEN !use time(2*i+2)
      { compare smallest sample of second pair with greatest of first}
      {second pair stop with first pair stop}
      IF time(2*i+2) < time(2*i) THEN  !if less than, add overflow value
        FOR j = (2*i+1) TO (2*N)       !to all following samples
          time(j) = time(j) + 2^32
        NEXT j
      ENDIF
   ELSE                               !stop > or = to start, use time(2*i+1)
      {compare smallest sample of second pair with greatest of first}
      {second pair stop with first pair stop}
     IF time(2*i+1) < time(2*i) THEN
        {if less than, add overflow to all following samples}
       FOR j = (2*i+1) TO (2*N)
         time(j) = time(j) + 2^32     ! add overflow value
       NEXT j
      ENDIF
   ENDIF
  ENDIF
NEXT i
```

For Format 4B, there is no **event data.**

FORMAT 5A

See Format 2A

FORMAT 5B

See Format 4B

FORMAT 6

The rollover processing routine for the time data is:

N = the number of measurements
(There are 2N samples for this format).

Data Structure Notes:

1. One dimensional array.
2. Odd indices reference Start Samples.
3. Even indices reference Stop Samples.

```
FOR i = 1 TO (2*N-1)
  IF time(i+1) < time(i) THEN   !is next sample less than current sample
   FOR j = (i+1) TO (2*N)        !for all following samples
     time(j) = time(j) + 2^32   !add overflow value
   NEXT j
  ENDIF
NEXT i
```

The rollover processing routine for the events data is:

N = the number of measurements
(There are N samples for this format).

> *NOTE*
>
> *Although this routine is written for 2N samples, only the stop data is valid in this mode, and the odd numbered samples are ignored.*

Data Structure Notes:

1. One dimensional array.
2. Odd indices are unused.
3. Even Indices reference Stop Samples.

```
FOR i = 1 TO (N-1)
  IF event(2*i+2)< event(2*i) THEN  !is next sample less than current sample
   FOR j = (i+1) TO (N)            !for all following samples
     event(2*j) = event(2*j) + 2^32!add overflow value
   NEXT j
  ENDIF
NEXT i
```

FORMAT 8

See Format 4A

FORMAT 10A

See Format 4A

FORMAT 10B

See Format 4B

FORMAT 10C

See Format 4A

FORMATS 11, 12, AND 13

Rollover processing in Totalize mode also checks to see if the time and event data increase monotonically. If a time or event datum is less than a previous datum then a counter rollover must have occurred. Care must be taken, however, to determine which time and event data is valid. In Totalize mode the 5373A attempts to capture event data twice. Thus, event data must be checked to see whether the first attempt or the second attempt at data capture is valid. This is done by checking the appropriate status bit in the Interpolator/Status Field. If the status bit is a zero, use the first attempt at data capture; if the status bit is one, use the second attempt at data capture and subtract 1 from the binary number represented by the second attempt at data capture. The valid time data always refers to the second attempt at data capture.

NOTE

Format 11 uses TS1 as the status bit. Format 12 uses TS2 as the status bit. Format 13 uses TS1 as the channel A status bit and TS2 as the channel B status bit.

This program assumes use of Format 11. Therefore the status bit used is TS1. Rollover routines for Formats 12 and 13 should be changed to use the appropriate status bits.

The algorithm for the events data is:

N=the number of desired results
There are 2(N+1) samples for this format.

This example uses two one-dimensional arrays to hold binary event data and status data.

The array which holds the event data is structured as follows: Data in the odd indices are the first attempt at data capture. Data in the even indices are the second attempt at data capture.

The array which holds the status data is structured as follows:

Data in the odd indices are unused. Data in the even indices hold status data for corresponding samples in the event data array.

```
FOR i = 2 to 2*(N+1) STEP 2          !This subtracts
  event(i) = event(i) - 1            !1 from every
NEXT i                               !second attempt
                                       !at data capture.
FOR i = 1 TO N
  IF TS1 (2*i) THEN !valid current sample is second read attempt
    IF TS1 (2*i+2) THEN !valid next sample is that pair's second read attempt
        {check second read and second read}
      IF event(2*i+2) < event(2*i) THEN !next sample < than current sample
        FOR j = (2*i+2) TO 2(N+1)      !for all following samples
          event(j) = event(j) + 2^32  !add overflow value to all samples
```

```
        NEXT j
      ENDIF
    ELSE !valid next sample is that pairs first read attempt
        {check second read and first read}
     IF event(2*i+1) < event(2*i) THEN !next sample < than current sample
       FOR j = (2*i+1) TO 2(N+1)       !for all following samples
         event(j) = event(j) + 2^32 !add overflow value to all samples
       NEXT j
     ENDIF
 ELSE !valid current sample is first read attempt
   IF TS1 (2*i+2) THEN !valid next sample is that pair's second read attempt
        {check first read and second read}
     IF event(2*i+2) < event(2*i-1) THEN !next sample < than current sample
       FOR j = (2*i+2) TO 2(N+1)         !for all following samples
         event(j) = event(j) + 2^32     !add overflow value to all samples
       NEXT j
    ENDIF
  ELSE !valid next sample is that pair's first read attempt
         {check first read and first read}
      IF event(2*i+1) < event(2*i-1) THEN !next sample < than current
        FOR j = (2*i+1) TO 2(N+1)        !for all following samples
          event(j) = event(j) + 2^32   !add overflow value to all samples
        NEXT j
      ENDIF
  ENDIF
NEXT i
```

The algorithm for the time data is:

N = the number of desired results
There are 2(N+1) samples for this format.

NOTE

Only use the second attempt at data capture.

This example uses a one-dimensional array to hold all the binary time data. Data in the odd indices is unused. Data in the even indices contain time data for corresponding samples in the event data array.

```
FOR i = 1 TO N
  IF time(2*i+2) < time(2*i) THEN !is next sample less than current sample
    FOR j = (i+1) TO (N+1)      !for all following samples
        {add overflow value to all second read samples}
      time(2*j) = time(2*j) + 2^32
    NEXT j
  ENDIF
NEXT i
```

FORMATS: 14 AND 15

The algorithm for the event data is:

N = the number of desired results
There are 4N samples for this format.

NOTE

Format 14 uses TS1 as the STATUS bit. Format 15 uses TS1 as the Channel A STATUS bit and TS2 as the Channel B STATUS bit.

This program assumes use of Format 14. Therefore the status bit that is used is TS1. Rollover routines for Format 15 should be changed to use the appropriate status bits.

```
FOR i = 2 to 4*N STEP 2             !This subtracts
 event(i) = event(i)-1              !1 from every
NEXT i                              !second attempt
                                      !at data capture.
FOR i = 1 TO 2*N-1
 IF TS1 (2*i) THEN  !TS1 set
   IF TS1 (2*i+2) THEN  !TS1 set
        {check second read and second read}
    IF event(2*i+2) < event(2*i) THEN !next sample < than current sample
      FOR j = (2*i+2) TO 4*N          !for all following samples
        event(j) = event(j) + 2^32  !add overflow value to all samples
      NEXT j
    ENDIF
   ELSE   !TS1 clear
       {check second read and first read}
   IF event(2*i+1) < event(2*i) THEN !next sample < than current sample
    FOR j =  (2*i+1) TO 4*N           !for all following samples
      event(j) = event(j) +  2^32   !add overflow value to all samples
    NEXT j
 ENDIF
ELSE !TS1 clear
  IF TS1 (2*i+2) THEN !TS1 set
       {check first read and second read}
    IF event(2*i+2) < event(2*i-1) THEN !next sample < than current
      FOR j = (2*i+2) TO 4*N      !for all following samples
        event(j) = event(j) + 2^32 !add overflow value to all samples
      NEXT j
    ENDIF
  ELSE !TS1 clear
       {check first read and first read}
    IF event(2*i+1) < event(2*i-1) THEN !next sample < than current
      FOR j = (2*i+1) TO 4*N      !for all following samples
```

```
        event(j) = event(j) + 2^32 !add overflow value to all samples
      NEXT j
    ENDIF
ENDIF
NEXT i
```

The algorithm for the time data is:

N=the number of desired results
There are 4N samples for this format.

NOTE

Only use the second attempt at data capture.

```
FOR i = 1 TO 2*N-1
  IF time(2*i+2) < time(2*i) THEN !is next sample less than current sample
    FOR j = (i+1) TO (2*N)       !for all following samples
        {add overflow value to all second read samples}
      time(2*j) = time(2*j) + 2^32
    NEXT j
  ENDIF
NEXT i
```

TECHNICAL NOTE 4

How To Determine Offset Values and Start/Stop Channels

For certain measurements, an offset value must be added to the Time Data in order to correct for differences in electrical path length between measurement channels. The offset value required depends on the HP 5373A measurement setup. This Technical Note describes how to calculate offset values.

The following Formats require offset values be added to Time Data:
1B,2B,3,4A,4B,5A,5B,6,8,10A,10B,10C,14,15

OFFSET DETERMINATION FOR FORMATS 4A, 4B, 5A, 5B, 6, 8, 10A, 10B, 10C, 14, 15

To determine the offset value to be added to the Binary Time Data for a particular measurement follow these steps.

1. Determine which channel is the start channel, see *START/STOP* table at the end of this Technical Note.
2. Determine the total delay on the measurement's start channel by following the *OFFSET FLOWCHART* for that particular channel. The *OFFSET FLOWCHART* is located next to the *START/STOP* table at the end of this Technical Note.

   For example: For a Time Interval B→ A measurement, Channel B is the start channel. Use the *OFFSET FLOWCHART* to sum all the delays for Channel B.
3. Determine which channel is the stop channel, see *START/STOP* table.
4. Determine the total delay on the measurement's stop channel by following the *OFFSET FLOWCHART* for that channel.

   For example: For a Time Interval B→ A measurement, Channel A is the stop channel. Use the *OFFSET FLOWCHART* to sum all the delays for Channel A.
5. Subtract the total delay for the stop channel from the total delay for the start channel . This difference will be the offset value to be added to the stop time data. The formula is:

   Offset = (Start channel delay) – (Stop channel delay)

Example 1:

Measurement:
Time Interval A→ B
Separate Mode
Channel A - X2.5 Attn.
Channel B - X1 Attn.

Following the *OFFSET FLOWCHART*, the delay in Channel A is:
9,900 ps + 0 ps + 400 ps + 600 ps = 10,900 ps

Following the *OFFSET FLOWCHART*, the delay in Channel B is:
10,100 ps + 0 ps + 0 ps + 0 ps = 10,100 ps

The offset to be added to the stop time data is:
Channel A delay – Channel B delay = 10,900 ps – 10,100 ps = 800 ps.

Example 2:

Measurement set up:
Time Interval B→ A
Common Mode
Channel A - X2.5 Attn.
Channel B - X2.5 Attn.

Following the *OFFSET FLOWCHART*, the delay in Channel B is:
10,100 ps + 800 ps + 400 ps + 600 ps = 11,900 ps

Following the *OFFSET FLOWCHART*, the delay in Channel A is:
9,900 ps + 0 ps + 400 ps + 0 ps = 10,300 ps

The offset that should be added to the stop time data is:
Channel B delay – Channel A delay = 11,900 ps – 10,300 ps = 1600 ps.

OFFSET DETERMINATION FOR FORMATS 1B, 3, 2B

Because these measurements time stamp the Block Arm, the offset value refers to the delay between the block arming channel and the measurement channel. The offset for these measurements is the delay between the measurement channel and the channel that arms the block of measurements. The offset value should be subtracted from the first time datum (this first time datum represents the time stamp of the Block Arm).

To calculate this offset use follow these steps:

1. Determine the total delay on the block arming channel by following the *OFFSET FLOWCHART* for that channel.

2. Determine the total delay on the measurement channel by following the *OFFSET FLOWCHART* for that channel.

3. Subtract the total delay for the measurement channel from the total delay for the block arming channel. This difference will be the offset value to be subtracted from the first time datum (this first time datum represents the time stamp of the Block Arm). The formula is:

   Offset = (Block Arming channel) – (measurement channel)

Example 3:

Measurement:
Continuous Time Interval
Channel A X2.5 Attn.
arm block of measurements on positive edge of External Arm

Following the *OFFSET FLOWCHART,* the delay in the External Arm is:
14,000 ps + 600 ps = 14,600 ps

Following the *OFFSET FLOWCHART,* the delay in Channel A is:
9,900 ps + 0 ps + 400 ps + 0 ps = 10,300 ps

The offset that should be subtracted from the first Time Datum is:
External Arm delay – Channel A delay = 14,600 ps – 10,300 ps = 4,300 ps.

Example 4:

Measurement:
Continuous Time Interval on Channel B
Channel A X2.5 Attn.
Channel B X2.5 Attn.
arm block of measurements on positive edge of Channel A

Following the *OFFSET FLOWCHART,* the delay in Channel A is:
9,900 ps + 0 ps + 400 ps + 600 ps = 10,900 ps

Following the *OFFSET FLOWCHART,* the delay in Channel B is:
10,100 ps + 0 ps + 400 ps +0 ps = 10,500 ps

The offset that should be subtracted from the first time datum is:
Channel A delay – Channel B delay = 10,900 ps – 10,500 ps = 400 ps.

Start/Stop Table

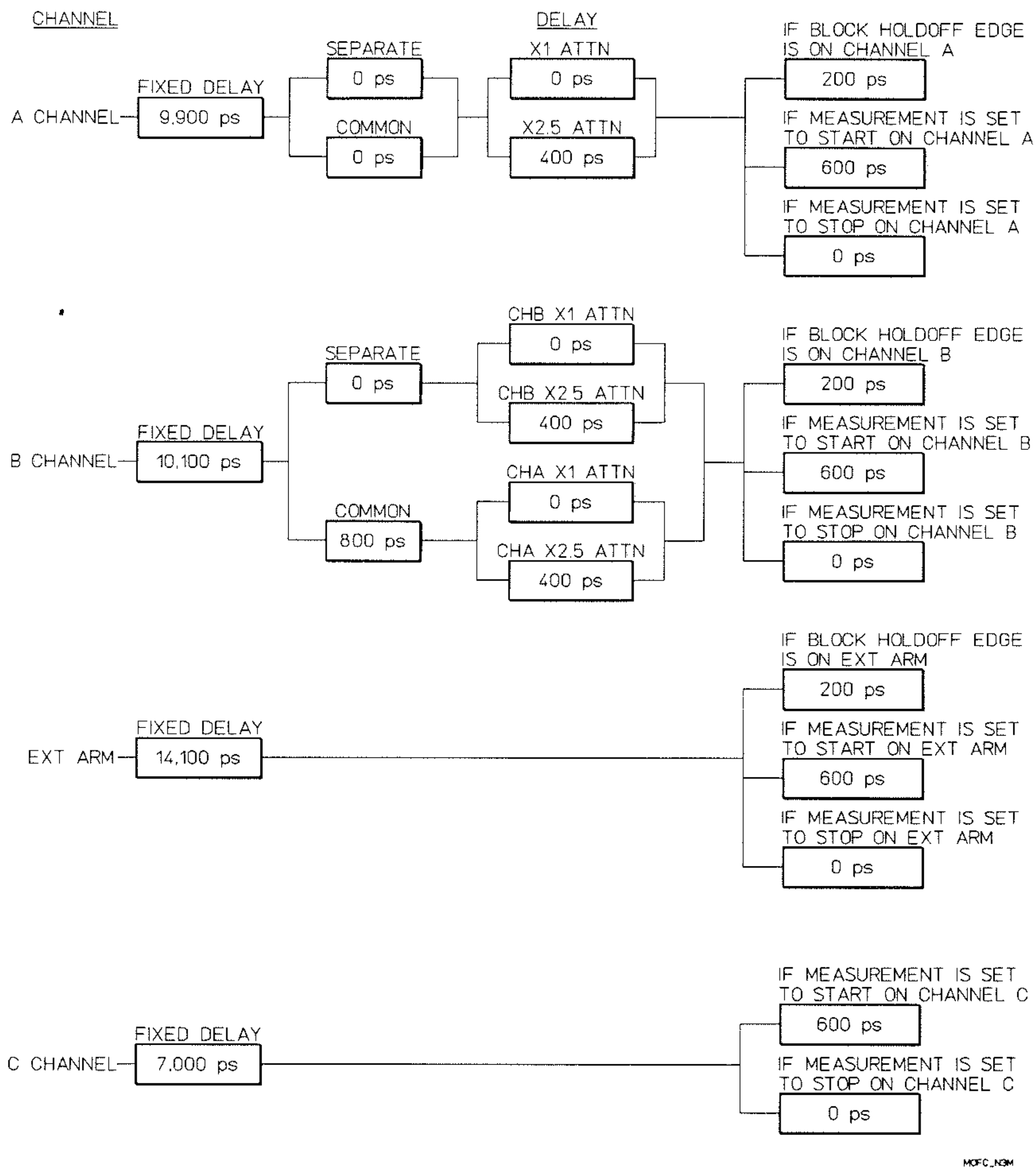
This table enables you to determine which channel (A,B, or C) is the start channel, and which is the stop channel for any measurement. Use this table along with the *OFFSET FLOWCHART* to determine the offset value to be added to your time data.

MEASUREMENT	CHANNEL SELECTION	START SAMPLE CHANNEL	STOP SAMPLE CHANNEL
Duty Cycle	A	A RISING EDGE	B (COMMON) FALLING EDGE
Non-Continuous or Externally Gated PRF, Frequency, PRI, or Period *(see note below)*	A B C	A B C A A B	A B C B C C
PRF, Frequency, PRI, or Period	A&B,A/B,B/A,A+B,A-B,B-A A&C,A/C,C/A,A+C,C-A,A-C B&C,B/C,C/B,B+C,B-C,C-B	A A B	B C C
Pulse Width	A	A RISING EDGE	B (COMMON) FALLING EDGE
Pulse Offtime	A	A FALLING EDGE	B (COMMON) RISING EDGE
Phase	A REL B B REL A	A A	B B
Rise Time	A	A 20% POINT	B(COMMON) 80% POINT
Fall Time	A	A 80% POINT	B(COMMON) 20% POINT
Time Interval	A B A→B B→A	A B A B	A B B A
±Time Interval	A B A→B B→A	A B A B	A B B A

Note: non-continuous PRF, Frequency, PRI, and Period measurements refer to PRF, Frequency, PRI, and Period measurements which use the following arming modes: Time Sampling, Edge/Time, Edge/Event, Time/Time, Event/Event.

OFFSET FLOW CHART

NOTE: FOR EXTERNALLY GATED TOTALIZE MEASUREMENTS THE OFFSET VALUE IS ALWAYS 600 ps.



4 PROGRAMMING RULES AND GUIDANCE

4

PROGRAMMING RULES AND GUIDANCE

INTRODUCTION

Programming rules pertain to the command structure, syntax, mnemonics, program flow and other high-level concepts that apply to writing code that the HP 5373A can recognize and execute. This chapter describes such rules and gives guidance to help you write programs.

CONVENTIONS

Some HP-IB command names are different from the corresponding front-panel function names. In this manual, both versions are given. Distinctions are clarified in the text of this chapter and in accompanying illustrations and tables.

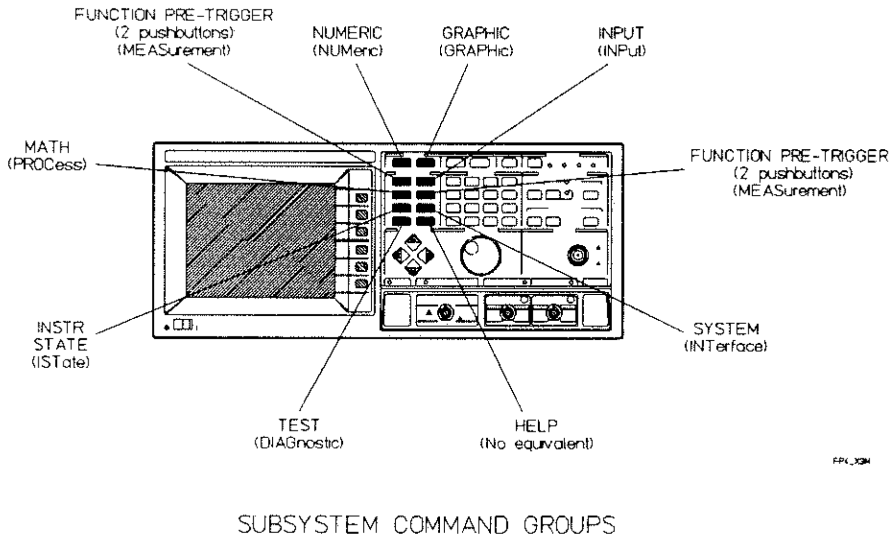
In this manual, front-panel control names are spelled out and printed in all capital letters. If a front-panel name is abbreviated on the instrument, that is the way it is spelled, if the reference is to the front-panel designation. For example INSTR STATE stands for Instrument State on the instrument. The applicable HP-IB command convention generally spells out the longform (ISTate) of the corresponding subsystem.

COMMAND STRUCTURE AND ELEMENTS

The HP 5373A front-panel layout gathers functionally related pushbutton controls in groups, for example:

- RESULTS
- HP-IB STATUS
- MENU SELECTION
- DATA ENTRY
- INSTRUMENT CONTROL

Likewise, The HP 5373A hierarchical command structure, *Figure 4-1*, gathers functionally related HP-IB commands in groups. The command groups approximately follow the instrument menus. If you are not already familiar with the



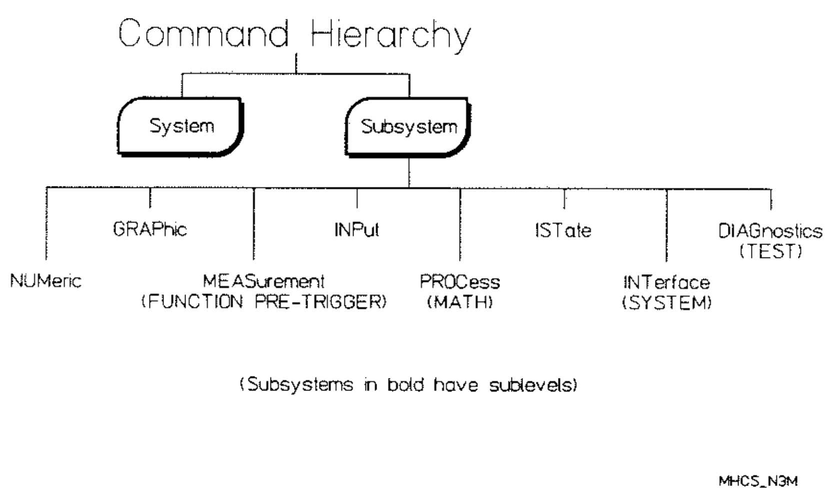


Figure 4-1. Hierarchical Command Structure

HP 5373A front-panel controls and menus, refer to the Operating Manual for a complete explanation of these topics.

Command Hierarchy

The command hierarchy contains these elements

- system commands
- subsystem commands
  - sublevel commands (only under the following subsystems)

    GRAPhic
    MEASurement (FUNCTION/PRE-TRIGGER)
    INPut
    PROCess (MATH)
    ISTate

System commands are general-purpose high-level commands. On the other hand, subsystem commands correspond to front-panel functional controls that relate to specific activities, such as making measurements or displaying results.

System commands and subsystem commands are at the same level in the hierarchy. Beneath a subsystem classification are the respective subsystem commands. In turn, sublevel commands fall under five of the eight subsystems.

As an organizational aid, the Command Reference (Chapter 5) is tabbed so you can readily turn to the functional groups illustrated. Alternately, to quickly find a command listed in the Command Reference, refer to the Table of Contents, Index, or *Table 5-1*.

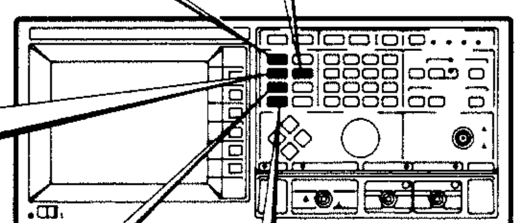
Matching Front-Panel Features to HP-IB Commands

Figure 4-2 illustrates how front-panel display choices relate to corresponding HP-IB commands. For each functional group a two-column table maps the relationship.

FUNCTION (MEASurement)	MEAS;
Measurement (Frequency, Time Interval, Totalize, etc.)	FUNCtion <function>
Channel (Measurement channel)	SOURce <channel>
Number of blocks	BLOCk <number>
Number of measurements/block	MSIZe or SSIZe <number>
Arming Mode	ARMing <arming mode>
Block Holdoff/Start	STARt
	CHANnel <A, B, or EXTernal>
	DCHannel <A or B>
	DELay <number>
	SLOPe <POSitive or NEGative>
Sample Arm/Stop	SAMPle
	CHANnel <A, B, or EXTernal>
	D CHannel <A, B, or OSCillator>
	DELay <number>
	SLOPe <POSitive or NEGative>

PRE-TRIGGER (MEASurement)	MEAS;
Pre-trigger mode	PMODe <TIME, TINTerval, EXTernal, or X>
Pre-trigger amount	PAMount <number> or <number>% or <number> sample
Pre-trigger Units	PUNits? (reply is "%" or "SAMP")
Pre-trigger Location	PLOCation? (returns pre-trigger index)
Pre-trigger external slope	PXSLope <POSitive or NEGative>
Time Interval Detect status	DETect <OFF, ABOVe,>, <,INSide,> <, OUTSide, <> >
Detector Lower threshold	DLOWer <number>
Detector Middle threshold	DMIDdle <number>
Detector Upper threshold	DUPPer <number>
Measurement Inhibit status	INHibit <ON or OFF>
Inhibit Level	ILEVel <TTL, GND, GROund, or ECL>
Inhibit Sense	ISENse <ABOVe or BELow>

MATH (PROCess)	PROC;
Channel	SOURce <channel>
Referece	SREFerence (Set Reference)
	CREFerence (Clear Reference)
Statistics	STATistics <ON or OFF>
Math	MATH <ON or OFF>
Offset	OFFSet <number>
Normalize	NORMalize <number>
Scale	SCALe <number>
Limits	LIMit <ON or OFF>
High	HLIMit <number>
Low	LLIMit <number>

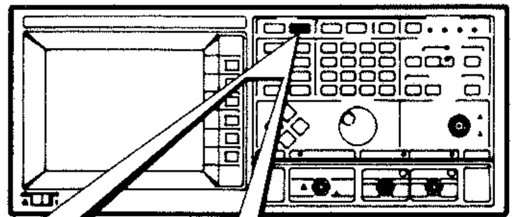


INSTR STATE (Instrument STate)	IST;
Write Protect	REGister <number>
	PROTect <ON or OFF>
Erase Register Data	ERASe <number>

TEST (DIAGnostic)	DIAG;
Run	TEST <number>
Pause	PAUSe
Continue	CONTinue (Continues execution of a test suspended by the PAUSe command)
Until Fail	UFAil <ON or OFF>
STOP	STOP

Figure 4-2. Relationship of Command Structure to Front Panel Controls (1 of 3)

GRAPHIC (GRAPhic)	GRAP;
To select particular graph for display	GDISplay <HISTogram or TVARiation or ETIMe>
Show Graph	SGRaph <MAIN or MEMory or BOTH>
Copy Main graph to Memory	COPY
Select active marker color	SMARker <BLACk or WHITe>
Active marker orientation	MORient <VERTical or HORizontal>
To move active vertical marker left	MLEFt <number>
To move active vertical marker right	MRIGht <number>
To move active horizontal marker up	MUP <number>
To move active horizontal marker down	MDOWn <number>
Marker → Max	MMAXimum
Marker → Min	MMINimum
Move inactive marker to position of active marker	MMOVe
Marker Display Mode	MDMode <MARKer or DELTa or STATistics or MODulation>
Marker Next Mode	MNEXt <DATA or PIXel>
Screen scroll graph (scroll using arrow keys)	SSCRoll <LEFT or RIGHt>
Zoom in, out or return to full scale	ZOOM <IN or OUT or FULL>
Outline status	OUTLine <ON or OFF>
Grid status	GRID <ON or OFF>
Y-axis scaling mode	YSCale <LOG or LINear>
Update mode (selects display mode for multiple-pass measurements).	UPDate <WHILe or AFTer>
Connect on or off	CDATa <ON or OFF>
View channel for dual-channel measurements (PRF, Frequency, PRI, Period, Totalize A and B, B and C, or A and C)	VCHannel <A or B or C>



GRAPHIC (GRAPhic) (continued)	GRAP;
Histogram graph (HIST)	GRAP;HIST;
X-axis Manual scaling status	XMSCale <ON or OFF>
Y-axis Manual scaling status	YMSCale <ON or OFF>
X-axis Auto Range Hold	XARHold
X-axis Marker Range Hold	XMRHold
Y-axis Auto Range Hold	YARHold
Y-axis Marker Range Hold	YMRHold
X Minimum value	XMINimum <number>
Bin Width value	BWIDth <number>
Y Maximum value	YMAXimum <number>
Time Variation graph (TVAR)	GRAP;TVAR;
X-axis Manual scaling status	XMSCale <ON or OFF>
Y-axis Manual scaling status	YMSCale <ON or OFF>
X-axis Auto Range Hold	XARHold
X-axis Marker Range Hold	XMRHold
Y-axis Auto Range Hold	YARHold
Y-axis Marker Range Hold	YMRHold
X Minimum value	XMINimum <number>
X Maximum value	XMAXimum <number>
Y Minimum value	YMINimum <number>
Y Maximum value	YMAXimum <number>
Event Timing graph (ETIM)	GRAP;ETIM;
Time/X-Axis:	
X-axis Manual scaling status	XMSCale <ON or OFF>
X-axis Auto Range Hold	XARHold
X-axis Marker Range Hold	XMRHold
X Minimum value	XMINimum <number>
X Maximum value	XMAXimum <number>

Figure 4-2. Relationship of Command Structure to Front Panel Control (2 of 3)

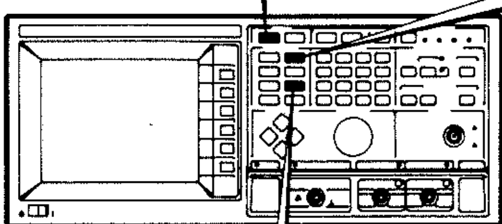
NUMERIC
(NUMeric) **NUM;**

Result Displays	
Result	DISP,NUMeric
Statistics	DISP,STATistics
Result/Statistics	DISP,SPLit
Limit Status	DISP,LIMit
Bold	DISP,BOLD
Gate Data (Expanded data)	EXPand <ON or OFF>
Scroll keys*	SCRoll <DOWN or UP>

*These are the Up/Down Cursor/Scroll front panel keys.

INPUT
(INPut) **INP;**

Input Channels (Separate/Common)	MODe <SEParate or COMMon>
Trigger Event:	
Channel	SOURce <channel>
Slope	SLOPe <POSitive or NEGative>
Mode	TRIGger <MANual or RAUTo or SAUTo>
Level	LEVel <voltage number> (Manual Trigger or External Arm)
	RLEVel <number> (Auto Trigger)
Bias	BIAS <ECL, GROund, or GND>
Attenuation	ATTenuation <X1 or X2>
Hysteresis	HYSTeresis <MINimum or MAXimum>



SYSTEM
(INTerface) **INT;**

HP-IB Configuration:	
Talk Only Print source	PSOurce <DISPlay or MEASurement> (Print Source)
Result Format	OUTPut <ASCii or BINary or FPOint>
Response Timeout	MTSTatus <ON or OFF> (Measurement Timeout Status)
	MTValue <number> (Measurement Timeout Value)
System Clock:	
DATE	DATe <year, month, day>
TIME	TODay <hour, minute, seconds> (Time of Day)

Figure 4-2. Relationship of Command Structure to Front Panel Control (3 of 3)

Commands and Queries

There are two command types: those that issue instructions to the instrument to perform specific operations, and queries that interrogate the instrument for a current state, or request the return of data. In this respect, a query is just a specific kind of command that performs a distinct function. Within the command structure is a subset of commands that you can transform into respective queries simply by appending a question mark. On the other hand, some commands exist only in the query form: no parameter entry or non-query mode exists for these commands. Query-only examples are ERR?, XMARker?, and *ESR? Each Command Reference entry (Chapter 5) identifies the classification in the main entry of the listing that appears in the Table of Contents (command, command/query or query only).

Queries are not allowed while the instrument is in the Binary output mode. Any responses from the instrument while in the Binary output mode are in binary format. Accordingly, queries requiring a string response will appear incorrect to the controlling program.

Duplication of Command Names

Some subsystem command names are duplicated from subsystem to subsystem. For example, the subsystem command SOURce occurs in the INPut, MEASurement, and PROCess subsystems. You must keep these duplications in mind when programming the HP 5373A.

Likewise, some subsystem sublevel command groups have duplicate command names. For example XMINimum, XMAXimum, YMINimum, and YMAXimum are duplicated in several GRAPhic subsystem sublevels.

System Commands

The system commands are:

ABORt	MMODe?	*ESE?
BEND?	ODATa?	*ESR?
BTIMe?	OHIStogram?	*HSE
CANCel	PLOCation?	*HSE?
CLEar	PLOT	*HSR?
DATe	POD?	*IDN?
DATe?	PRESet	*OPC
DLENgth?	PRINt	*OPC?
DMSetup	REMote	*OPT?
DSP	RESTart	*PSC
DSP?	SETup	*PSC?
ERRor?	SETup?	*RCL
HOVer?	SMODe	*RST
HUNder?	SMODe?	*SAV
HWIThin?	SUBS?	*SRE
IDATa?	TODay	*SRE?
KEY	TODay?	*STB?
KEY?	WINTerval?	*TRG
LOCal	WTSend	*TST?
MENu	WTSend?	
MENu?	*CLS	
MMODe	*ESE	

IEEE COMMON COMMANDS

System commands are divided into two subgroups: HP 5373A (instrument specific) commands and IEEE common commands. The IEEE common commands include an asterisk (*) as the first character in their command mnemonic. They act just like instrument specific commands; the only difference is they lack longform command mnemonics. The IEEE common commands are defined by IEEE-488 standards and function identically in all instruments that conform to the standard. (That is, the commands are device-independent.)

HP 5373A SPECIFIC

Instrument specific commands are all those commands that do not have an asterisk in the command mnemonic. They consist of commands, command/queries, and query only types.

Subsystem Commands

Here are the subsystem command groups. (Terms enclosed in parentheses are the corresponding menu names for those subsystems. Most subsystem and menu names are alike)

NUMeric: (NUMERIC)	Controls display of numeric results, with options for scrolling and large character fonts.
GRAPhic: (GRAPHIC)	Turns data into a graphic display: Histogram, Time Variation, or Event Timing graph
MEASurement: (FUNCTION/ PRE-TRIGGER)	Controls measurement, arming, and gating modes. Also controls Pre-trigger functions.
INPut: (INPUT)	Controls input signal conditioning parameters, trigger levels, and slope.
PROCess: (MATH)	Controls measurement post-processing options, such as math statistical functions and limit testing.
ISTate: (INSTR STATE)	Displays status information about each Save/Recall register.
INTerface: (SYSTEM)	Controls the configuration of the HP-IB interface. Do not confuse this set of commands with the general purpose high level system commands.
DIAGnostic: (TEST)	Provides access to selected diagnostic tests.

Sublevel Commands

Five subsystems have commands that in turn have sublevel commands:

- GRAPhic
- INPut
- ISTate
- MEASurement (FUNCTION/PRE-TRIGGER)
- PROCess (MATH)

Sublevel commands relate to respective subsystem functions. The subsystem sublevel commands can have modifiers in the same way that system level commands can take arguments.

For example, the Time Variation sublevel has an XMAXimum sublevel command that controls the graph X-axis. An example command string to set this value looks like this.

```
OUTPUT 703;"GRAP;TVAR;XMAX,2.0"
```

NAVIGATING THE HIERARCHY

To select a subsystem command you first issue a subsystem selector, which is simply the functional name of the subsystem group (for instance MEASurement). Then, you can issue an applicable subsystem command. You can also choose a subsystem sublevel command in the same program line. For example:

```
OUTPUT 703;"MEAS;FUNC,FREQ;SOUR,A"
```

The Active Subsystem or Sublevel Scheme

Once a given subsystem or sublevel is selected, that level remains active until your program designates another subsystem or sublevel. This active subsystem scheme makes an efficient command structure because you can continue to issue commands in an active subsystem (or sublevel) without reiterating a subsystem or sublevel call. A drawback is that some name duplication occurs across subsystems. This means you must keep track of which subsystem is current to ensure the result you want. (The SUBSystem? query can help in this regard.) The principle of selecting an active set of commands only applies to subsystem and sublevel commands; system commands are a different case.

What Happens When You Issue a System Command

Your program can issue a system command at any time, without regard to the active subsystem selection. For example, if your program selects the PROCess subsystem level and you then want to issue a PRINt command from the system command group, you can do so without regard to the active subsystem. The PROCess subsystem stays active and your program can continue to issue PROCess commands after the PRINt command executes. In other words, System commands have no effect on the active subsystem or sublevel.

DIFFERENCES BETWEEN HP-IB AND FRONT-PANEL CONTROL

There is not an exact one-to-one correspondence between HP-IB commands and front-panel controls. For example, there is no need to have a "Key" pushbutton on the front panel, you merely push the desired key. But a KEY command is required to simulate a key press using the HP-IB control scheme. Nevertheless, mentally mapping the front-panel control scheme to the HP-IB command structure provides a useful way to remember which commands control which operating features.

In some cases differences arise because of design constraints, or other factors. In this regard, the HP-IB Pre-trigger commands present a special case. Although front-panel Pre-trigger commands are grouped under a corresponding PRE-TRIGGER pushbutton, the corresponding HP-IB commands are selected via the MEASurement (FUNCTION/ PRE-TRIGGER) subsystem selector. This is the only exception to the one-to-one tracking of subsystem selection and front-panel controls. In this manual, the Pre-trigger commands are merged alphabetically in the MEASurement (FUNCTION/ PRE-TRIGGER) tabbed section of the Command Reference (Chapter 5).

COMMAND MNEMONICS

The HP 5373A accepts shortform and longform programming commands and queries. The HP 5373A parser accepts upper or lower case letters. In the manual text, upper case letters denote the command shortform; the complete spelling is the longform. Shortform and longform versions are derived using the rules covered below.

Choosing Either Shortform or Longform

Use the shortform if conserving memory space is a consideration. Doing so results in a more efficient program. The efficiencies are in two areas: your program occupies less total program memory, and it runs slightly faster since fewer characters are transmitted to the HP 5373A and the instrument has fewer characters to parse. Alternately, you can make your programs easier to read and understand by using longform commands.

How Longform Commands Are Derived

Longform command mnemonics are generally formed according to the following rules:

If a function description is one word, the longform is the entire word.

For example,

DISPLAY for the Display function
RESTART for the Restart function.

If a function description is more than one word, the longform is the first letter of each word except for the last word, which is simply appended without modification.

For example,

MSIZE for Measurement Size
CTINTERVAL for Continuous Time Interval.

Exceptions to the rules above are:

AMPModulation, for Amplitude Modulation
OFFTime, for Pulse Offtime
POW, for Envelope Power
PRF, for Pulse Repetition Frequency
PRI, for Pulse Repetition Interval

How Shortform Commands Are Derived

A shortform mnemonic always has three or four characters and is the minimum command designation accepted by the HP 5373A. Shortform commands are derived by truncating the longform command according to the following rules:

If the longform is four characters, no truncation is necessary unless the last character is a vowel. In this case, the fourth character is dropped (for example, DATE abbreviates to DAT) and is written in the documentation as DATe.

If the longform has more than four characters, and if the fourth character is a vowel, then truncate the longform to three characters. Otherwise, truncate to four characters.

For example:

HISTOGRAM truncates to HIST (HISTogram)
CHANNEL truncates to CHAN (CHANnel)
ERROR? truncates to ERR? (ERRor?)
ARMING truncates to ARM (ARMing)

Alternate Command Mnemonics

The HP 5373A accepts alternate forms of certain shortform command mnemonics to accommodate industry-accepted standards or variations. For example, you can use RIS, RISE, or RTIM to select the Rise Time function. All alternate shortform command mnemonics are listed in *Table 4-1.*

Table 4-1. Alternate Command Mnemonics

Function or Parameter	Mnemonics Common
PRF (Pulse Repetition Frequency) or Frequency	PRF or FREQuency
PRI (Pulse Repetition Interval) or Period	PRI or PERiod
Duty Cycle	DUTY OR DCYC (DCYCle)
Fall Time	FALL or FTIM (FALLtime, FTIMe)
Rise Time	RIS, RISE, or RTIM (RISEtime, RTIMe)
Time Interval	TIM, TIME, or TINT (TINTerval)
Measurement size/sample size (these functions are identical)	MSIZ or SSIZ (MSIZe, SSIZe)
Ground	GRO or GND (GROund)
Root Mean Square	RMS or RMSQ (RMSQuare)

MAXIMUM COMMAND LINE LENGTH

The HP 5373A accepts a maximum command line of 80 characters, including delimiters. If more than 80 characters are sent, the extra characters are truncated and an error message appears on the status line of the display. This 80-line limit does not include the terminating (e.g. CR/LF) characters.

If you are using an HP-85B computer as a controller, a false error occurs when sending command lines of 64 to 79 characters. If a line length between 64 and 79 characters is sent, the HP 5373A will still process all commands up to the 79 character length. You can ignore this false error message if the command line length is equal to or less than 79 characters.

COMMAND FORM AND SEQUENCE

All commands are of the form COMMAND <space> <modifier> or COMMAND <,> <modifier>.

The command parser accepts command sequences in different program lines as long as the overall sequence is correct. For example, the following program segment:

```
10 OUTPUT 703;"PRES;MEAS;FUNC,FREQ;SOUR,A"
20 OUTPUT 703;"ARM,AUT;INP;MOD,SEP"
```

is equivalent to the following sequence:

```
10 OUTPUT 703;"PRES"          ! Presets the instrument
20 OUTPUT 703;"MEAS"          ! Selects Measurement
                              subsystem
30 OUTPUT 703;"FUNC,FREQ"     ! Sets the function to
                              Frequency
40 OUTPUT 703;"SOUR,A"        ! Selects Channel A a
                              measurement source
                              channel
50 OUTPUT 703;"ARM,AUT"       ! Selects Automatic
                              arming
60 OUTPUT 703;"INP"           ! Selects Input subsystem
70 OUTPUT 703;"MOD,SEP"       ! Selects Separate mode
                              for Channel A and
                              Channel B
```

COMMAND FORMAT

The following paragraphs describe:

- Alpha and numeric argument formats
- Delimiters

Alpha and Numeric Arguments

Except for GRAPhic subsysem commands, a command that requires "ON" and "OFF" parameters may use either alpha or numeric arguments. The alpha argument "OFF" can be represented by "0" and the argument "ON" can be represented by "1". Note that (except for the GRAPhic subsystem) queries of binary parameters are returned as "1" for values of "ON" and "0" for values of "OFF" (the numeric representations are defined by IEEE standards as the "shortforms" for the alpha arguments "ON" and "OFF").

Parameter Formats

The parameter field, following a command, can be alpha or numeric. A parameter field must be one of three types:

Numeric — An integer, floating point, or exponential value. When sending an exponent, the characters <E> or <e> must be used to delimit the mantissa. The syntax follows the integer (NR1), decimal (NR2), or exponential (NR3) formats illustrated in *Figure* 4-3.

Alpha — ASCII strings that start with an alpha character and are followed by any printable character except <space>,<;>, <,>, <#>, <">, or <_>. An apostrophe <'> is allowed. Examples of alpha arguments are "OFF" and "ON".

ASCII String — Any collection of ASCII characters delimited by quotation marks <">, except that embedded quotation marks are not allowed.

Block — A block of binary data defined as:

```
<#> <length> <length bytes> <DAB> <DAB>...
```

where <length> is a single byte that contains the number of <length bytes>. The number specified by <length bytes> represents the number of data bytes <DAB> that follow. In the following example, the number 6 means that 6 bytes are used to identify the length bytes field and 000123 means that 123 bytes follow:

```
#6000123........
```

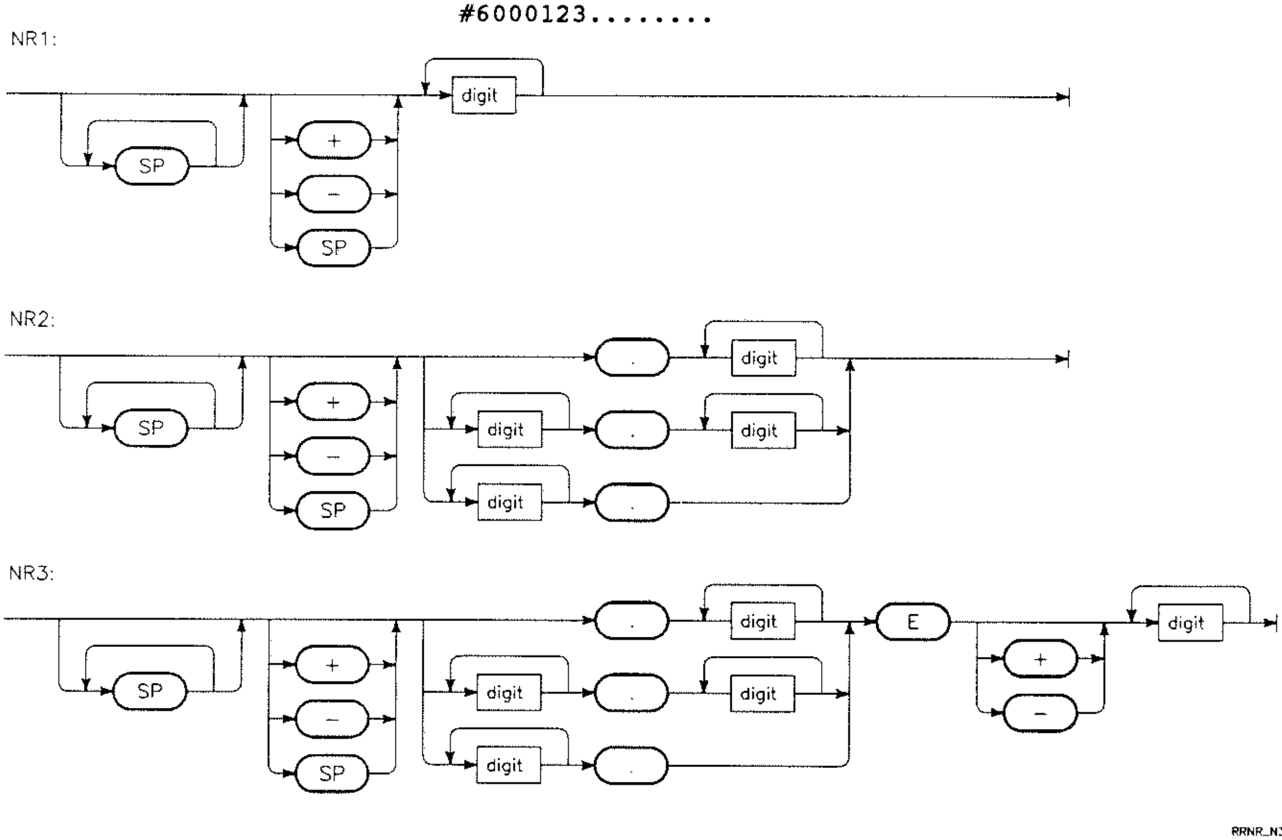


Figure 4-3. Numeric Data Formats

GENERAL FORMATTING RULES

The general rules for command formatting are:

- All program messages (except binary types) are interpreted as standard 7-bit ASCII code. A program message is a string of one or more program commands followed by an End-of-String (EOS) message.
- No distinction is made between upper and lower case letters.
- Except for GRAPhic subsystem commands, for a command requiring "ON" and "OFF" as arguments, the instrument will respond to "1" and "0" equally well.
- Command headers and multiple arguments must be explicitly delimited.

SYNTAX DIAGRAMS EXPLAINED

Figure 4-4 is a top level command syntax diagram. This diagram relates the entire subsystem and sublevel command structure and graphically reveals how you can string commands together, using delimiters as appropriate. Use the syntax diagrams to work out program flow and visually distinguish subsystem commands and sublevel commands.

Interpreting the Top Level Syntax Diagrams

The key to interpreting the top level diagram (and other syntax diagrams printed elsewhere) is to understand the looping flow implied by the arrow heads. You can only "move" in the direction of an arrow. For example, if you enter from the left (at the box labeled Subsystem Selectors) you can choose one of the eight subsystems, select a command, drop down and select a sublevel command (if applicable), come back to the top and stop. Or, continue by inserting a separator (at the top of the diagram) and then loop back and select another subsystem.

Subsystem commands and sublevel commands diagrammed in this illustration are completely collapsed representations of what you see diagrammed at the start of the respective Command Reference entries (Chapter 5). The Command Reference diagrams are expanded to show every command and sublevel command and give applicable command arguments and modifiers. Queries are also diagrammed.

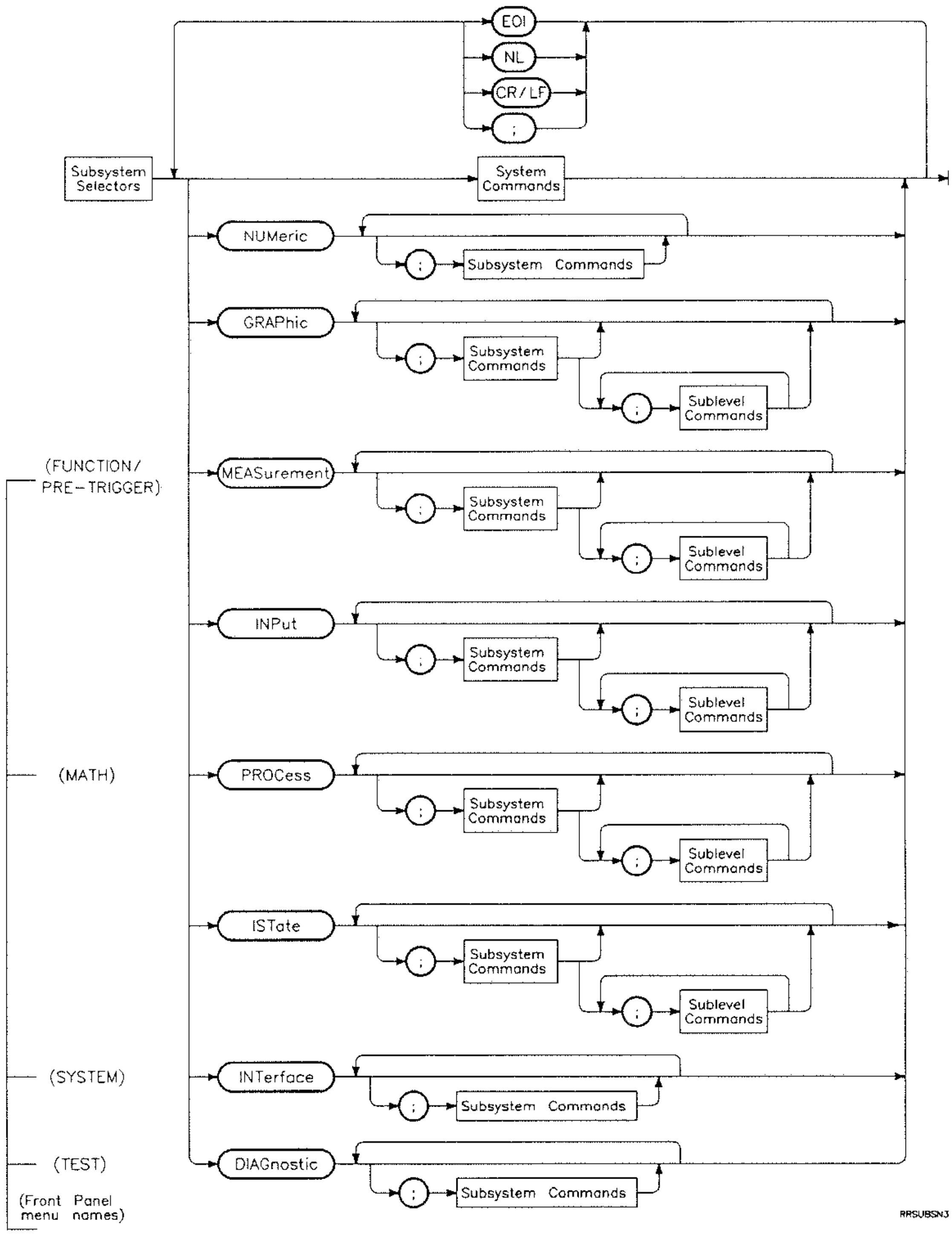


Figure 4-4. Top Level Command Syntax Diagram

Respective syntax diagrams appear at the beginning of each subsystem group. A syntax diagram includes the subcommands and parameters for each command. Some syntax diagrams require more than one page. If so, continuation pages are provided as needed. Sublevels are diagrammed separately, if there are more than a few sublevel commands.

Message Delimiters (separators)

The use of delimiters in a program message is governed by the following rules:

- Separate multiple arguments for a program command using commas <,> or spaces <SP>.
- Delimit program commands with a semicolon <;>.
- Use a new line <NL>, carriage return/line feed <CR/LF>, or EOI (End Or Identify signal) as the EOS message. Terminate each program message either by a <NL>, <CR/LF>, or by asserting the EOI signal with the last byte in the message.

One or more carriage returns <CR> immediately preceding a new line <NL> character are ignored by the instrument. A good way to understand how to appropriately use separators is to study example programs. Appendix E in this manual lists several example programs written in HP BASIC.

PROCESSING CONSIDERATIONS

Program commands that comprise a data message string are not executed until the entire string is sent. The commands in the string are then interpreted and processed in the order received. Commands preceding an error in a multi-command message are executed up to the point where the error occurred.

PROGRAMMING TIPS

Two ways to go about programming the instrument are to use the front-panel control method as a guide to selecting applicable commands or to use the Teach/Learn method.

Going From Front Panel To Program

One way to write a program is to first devise a corresponding measurement using the front-panel operating features. Note the steps you take; then, write your program following the same flow you used to make the manual measurement, except write the corresponding commands.

Teach/Learn Explained

A program in Appendix E demonstrates how to use the "SETup?" query and the "SETup" command to save and recall front-panel setups, a technique known as the Teach/Learn method. As the name implies, Teach/Learn is a two step process. First the Teach/Learn program stores a desired front-panel setup in a string variable. Later, you select a stored string variable (the learned settings) to retrieve the corresponding front-panel setup. You can create an unlimited number of such variables to manage diverse measurement setups.

5

COMMAND REFERENCE

INTRODUCTION

This reference section lists and describes HP 5373A commands. The reference is prefaced by introductory material that defines the audience and explains how the section is organized. The listing scheme is explained. Finally, the introduction to the section concludes with a master list of commands organized able information. Use this master list to find a command alphabetically (without regard for subsystem or sublevel).

WHO THIS SECTION IS FOR

The presentation of material here assumes a given level of expertise and knowledge. To fully benefit from the information in the command listings, you should first read and understand the foregoing sections on HP-IB commands and programming rules and guidance.

HOW THIS SECTION IS ORGANIZED

Within this section, tabbed divisions correspond to subsystem levels. System commands come first (they are all general-purpose commands). Then, the ordering is by corresponding front panel function layout, from top-to-bottom, right-to-left, starting with the NUMeric subsystem commands. Here is the scheme.

- System
- NUMeric
- GRAPhic
- MEASurement (FUNCTION/PRE-TRIGGER)
- INPut
- PROCess (MATH)
- ISTate
- INTerface (SYSTEM)
- DIAGnostic (TEST)

Pre-trigger commands are alphabetically integrated with MEASurement subsystem commands even though there is a separate front panel pushbutton dedicated to pre-trigger functions. In the HP-IB scheme, pre-trigger commands are viewed as an extension of the MEASurement subsystem.

Commands within a subsystem tabbed section are usually organized alphabetically, except where functional grouping provides a more logical reference. The IEEE-488 common commands in the system section are all organized alphabetically and appear after the first group of HP 5373A specific System commands.

WHAT EACH LISTING INCLUDES

Each command description includes

- a heading and brief command function description (this heading appears in the table of contents),
- the shortform mnemonic, including the full functional descriptor from which the mnemonics are derived (in [brackets]),
- the longform mnemonic,
- a description of the command function and operation,
- required parameters and their ranges (if applicable),
- one or more example lines of code.

In a few cases, the function descriptor entry is clarified by additional terms that are not part of the command mnemonics. Accordingly, the explanatory terms are printed in parentheses within the brackets. For example:

Shortform: MATH [MATH (modifiers)]
Shortform: REST [RESTart (measurement)]

The terms *modifiers* and *measurement* respectively clarify MATH and RESTart.

The example code is written in HP Series 200/300 BASIC. All examples assume that the instrument address is set to three and the interface select code is set to seven.

MASTER LIST OF COMMANDS

Table 5-1 lists all commands in alphabetical order and references the respective sublevel. If you can remember a command name and want to know which subsystem the command belongs in, search the table. If you want to see all the commands in a given subsystem or sublevel, organized alphabetically, refer to the table of contents.

Table 5-1. Master List of Commands (Alphabetical Order)

COMMAND	WHERE USED	COMMAND	WHERE USED	COMMAND	WHERE USED
ABORt	System	DISPlay	NUM	HYSTeresis	INP;SOUR
ARMing	MEAS	DISPlay?	NUM	HYSTeresis?	INP;SOUR
ARMing?	MEAS	DLENgth?	System		
ATTenuation	INP;SOUR	DLOWer	MEAS	IDATa?	System
ATTenuation?	INP;SOUR	DLOWer?	MEAS	ILEVel	MEAS
AVARiance?	PROC;SOUR	DMIDdle	MEAS	ILEVel?	MEAS
		DMIDdle?	MEAS	INHibit	MEAS
BEND?	System	DMSetup	System	INHibit?	MEAS
BIAS	INP;SOUR	DSP	System	ISENse	MEAS
BIAS?	INP;SOUR	DSP?	System	ISENse?	MEAS
BLOCk	MEAS	DUPPer	MEAS		
BLOCk?	MEAS	DUPPer?	MEAS	KEY	System
BTIMe?	System			KEY?	System
BWIDth	GRAP;WALL	ERASe	IST		
BWIDth	GRAP;WHIST	ERRor?	System	LEVel	INP;SOUR
BWIDth	GRAP;WMUL	ETIMe	GRAP	LEVel?	INP;SOUR
BWIDth	GRAP;HIST	EVT1?	GRAP;ETIM	LIMit	PROC;SOUR
BWIDth?	GRAP;HIST	EVT1?	GRAP;TVAR	LIMit?	PROC;SOUR
		EVT2?	GRAP;ETIM	LLIMit	PROC;SOUR
CANCel	System	EVT2?	GRAP;TVAR	LLIMit?	PROC;SOUR
CARRier	PROC	EXPand	NUM	LOCal	System
CARRier?	PROC	EXPand?	NUM		
CATTen	INP;SOUR			MATH	PROC;SOUR
CATTen?	INP;SOUR	FUNCtion	MEAS	MATH?	PROC;SOUR
CDATa	GRAP	FUNCtion?	MEAS	MAXimum?	PROC;SOUR
CDATa?	GRAP			MCENter?	GRAP
CFRequency	PROC	GDISplay	GRAP	MDMode	GRAP
CFRequency?	PROC	GDISplay?	GRAP	MDMode?	GRAP
CHANnel	MEAS;SAMP	GRID	GRAP	MDOWn	GRAP
CHANnel	MEAS;STAR	GRID?	GRAP	MEAN?	PROC;SOUR
CHANnel?	MEAS;SAMP			MEMory	GRAP
CHANnel?	MEAS;STAR	HBINwidth	MEAS	MENu	System
CLISt?	System	HBINwidth?	MEAS	MENu?	System
CLEar	System	HBLock	MEAS	MGRaph	GRAP
CONTinue	DIAG	HBLock?	MEAS	MINimum?	PROC;SOUR
COPY	GRAP	HCENter	MEAS	MLEFt	GRAP
CPERiod	GRAP	HCENter?	MEAS	MMAXimum	GRAP
CPERiod?	GRAP	HCONtinue	GRAP	MMINimum	GRAP
CREFerence	PROC;SOUR	HFORmat	INT	MMODe	System
CSLope	PROC	HFORmat?	INT	MMODe?	System
CSLope?	PROC	HISTogram	GRAP	MMOVe	GRAP
CSTart	PROC	HLIMit	PROC;SOUR	MNEXt	GRAP
CSTart?	PROC	HLIMit?	PROC;SOUR	MNEXt?	GRAP
		HMAXimum?	GRAP	MNUMber?	GRAP
DATe	System	HMEan?	GRAP	MODe	INP
DATe?	System	HMINimum?	GRAP	MODe?	INP
DCHannel	MEAS;SAMP	HOVer?	System	MORient	GRAP
DCHannel	MEAS;STAR	HPAuse	GRAP	MORient?	GRAP
DCHannel?	MEAS;SAMP	HSDev?	GRAP	MRATe?	GRAP
DCHannel?	MEAS;STAR	HSPan	MEAS	MRIGht	GRAP
DELay	MEAS;SAMP	HSPan?	MEAS	MSEGment?	MEAS
DELay	MEAS;STAR	HSTart	MEAS	MSIZe	MEAS
DELay?	MEAS;SAMP	HSTart?	MEAS	MSIZe?	MEAS
DELay?	MEAS;STAR	HUNDer?	System	MTSTatus	INT
DETect	MEAS	HWIThin?	MEAS	MTSTatus?	INT
DETect?	MEAS	HWIThin?	System	MTValue	INT

Table 5-1. Master List of Commands (Continued)

COMMAND	WHERE USED
MTValue?	INT
MUP	GRAP
NORMalize	PROC;SOUR
NORMalize?	PROC;SOUR
ODATa?	System
OFFSet	PROC;SOUR
OFFSet?	PROC;SOUR
OHIStogram?	System
OUTLine	GRAP
OUTLine?	GRAP
OUTPut	INT
OUTPut?	INT
PAMount	MEAS
PAMount?	MEAS
PAUSe	DIAG
PBLock	MEAS
PBLock?	MEAS
PCOMpute	PROC
PDEViation?	GRAP
PLOCation?	System
PLOT	System
PMODe	MEAS
PMODe?	MEAS
POD?	System
PRESet	System
PRETrigger	MEAS
PRETrigger?	MEAS
PRINt	System
PROTect	IST;REG
PROTect?	IST;REG
PSOurce	INT
PSOurce?	INT
PUNits?	MEAS
PXSLope	MEAS
PXSLope?	MEAS
RAVariance?	PROC;SOUR
REGister	IST
REGister?	IST
REMote	System
RESTart	System
RLEVel	INP;SOUR
RLEVel?	INP;SOUR
RMSQuare?	PROC;SOUR
RMSquare?	PROC;SOUR
SAMPle	MEAS
SCALe	PROC;SOUR
SCALe?	PROC;SOUR
SCRoll	NUM
SDEViation?	PROC;SOUR
SELect	GRAP
SELect?	GRAP
SETup	System
SETup?	System
SGRaph	GRAP
SGRaph?	GRAP
SLOPe	INP;SOUR
SLOPe	MEAS;SAMP
SLOPe	MEAS;STAR
SLOPe?	INP;SOUR
SLOPe?	MEAS;SAMP

COMMAND	WHERE USED
SLOPe?	MEAS;STAR
SMARker	GRAP
SMARker?	GRAP
SMODe	System
SMODe?	System
SOURce	INP
SOURce	MEAS
SOURce	PROC
SOURce?	INP
SOURce?	MEAS
SREFerence	PROC;SOUR
SREFerence?	PROC;SOUR
SSCRoll	GRAP
SSIZe	MEAS
SSIZe?	MEAS
STARt	MEAS
STATistics	PROC;SOUR
STATistics?	PROC;SOUR
STOP	DIAG
SUBS?	System
TEST	DIAG
TEST?	DIAG
TODay	System
TODay?	System
TRIGger	INP;SOUR
TRIGger?	INP;SOUR
TVARiation	GRAP
TVResult	MEAS
TVResult?	MEAS
UFAil	DIAG
UFAil?	DIAG
UPDate	GRAP
UPDate?	GRAP
VARiance?	PROC;SOUR
VCHannel	GRAP
VCHannel?	GRAP
WINTerval?	System
WTSend	System
WTSend?	System
XARHold	GRAP;ETIM
XARHold	GRAP;HIST
XARHold	GRAP;TVAR
XMAXimum	GRAP;ETIM
XMAXimum	GRAP;TVAR
XMAXimum?	GRAP;ETIM
XMAXimum?	GRAP;MEM
XMAXimum?	GRAP;MGR
XMAXimum?	GRAP;TVAR
XMINimum	GRAP;ETIM
XMINimum	GRAP;HIST
XMINimum	GRAP;TVAR
XMINimum?	GRAP;ETIM
XMINimum?	GRAP;HIST
XMINimum?	GRAP;MEM
XMINimum?	GRAP;MGR
XMINimum?	GRAP;TVAR
XMRHold	GRAP;ETIM
XMRHold	GRAP;HIST
XMRHold	GRAP;TVAR
XMSCale	GRAP;ETIM

COMMAND	WHERE USED
XMSCale	GRAP;HIST
XMSCale	GRAP;TVAR
XMSCale?	GRAP;ETIM
XMSCale?	GRAP;HIST
XMSCale?	GRAP;TVAR
XVALue?	GRAP
YARHold	GRAP;HIST
YARHold	GRAP;TVAR
YMAXimum	GRAP;HIST
YMAXimum	GRAP;TVAR
YMAXimum?	GRAP;HIST
YMAXimum?	GRAP;MEM
YMAXimum?	GRAP;MGR
YMAXimum?	GRAP;TVAR
YMINimum	GRAP;TVAR
YMINimum?	GRAP;MEM
YMINimum?	GRAP;MGR
YMINimum?	GRAP;TVAR
YMRHold	GRAP;HIST
YMRHold	GRAP;TVAR
YMSCale	GRAP;HIST
YMSCale	GRAP;TVAR
YMSCale?	GRAP;HIST
YMSCale?	GRAP;TVAR
YSCale	GRAP
YSCale?	GRAP
YVALue?	GRAP
ZOOM	GRAP
*CLS	System
*ESE	System
*ESE?	System
*ESR?	System
*HSE	System
*HSE?	System
*HSR?	System
*IDN?	System
*OPC	System
*OPC?	System
*OPT?	System
*PSC	System
*PSC?	System
*RCL	System
*RST	System
*SAV	System
*SRE	System
*SRE?	System
*STB?	System
*TRG	System
*TST?	System

System

SYSTEM COMMANDS

System commands control general instrument functions. Your program can issue them at any time. That is, you can issue a system command no matter which subsystem is designated the active subsystem. System commands do not change the subsystem selection; when the system command executes, the HP 5373A returns to the current subsystem.

System commands are divided into two subgroups: HP 5373A-specific commands and IEEE common commands. The IEEE common commands are preceded by an asterisk (*) that is part of the command mnemonic. The common commands are identical to instrument specific system commands, except that they lack longform command mnemonics. For ease of reference, all of the IEEE common commands are grouped together after all of the HP 5373A-specific commands; system and common subgroups respectively appear in alphabetical order. *Figure 5-1* illustrates the system command syntax.

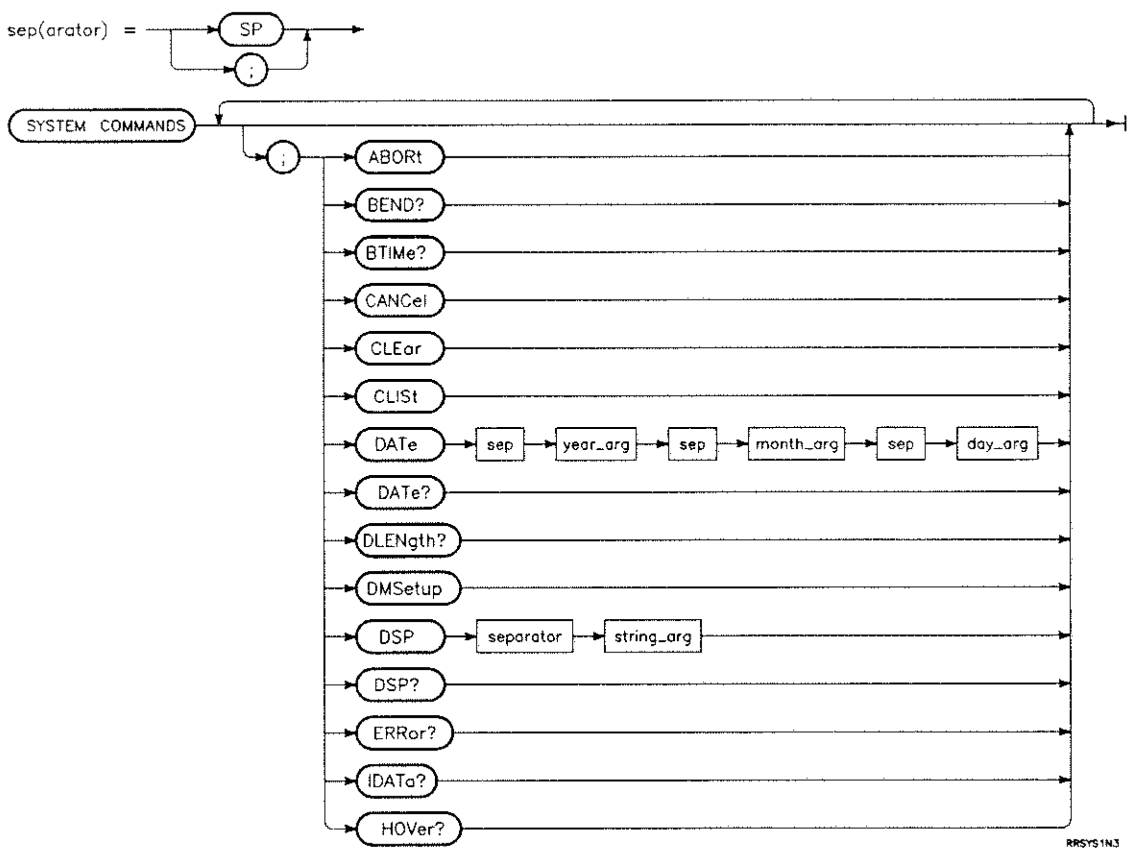


Figure 5-1. System Command Syntax Diagram (1 of 3)

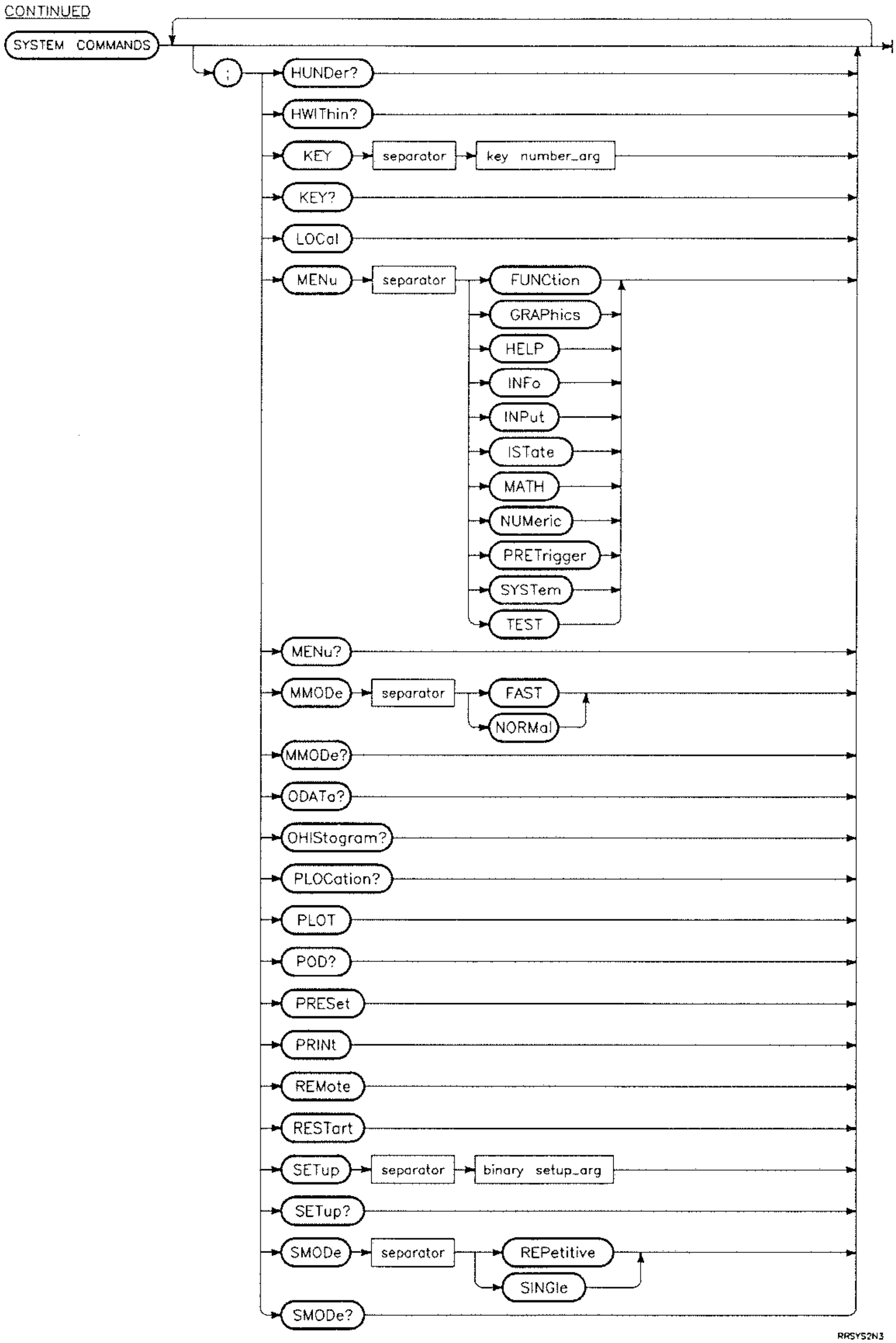


Figure 5-1. System Command Syntax Diagram (2 of 3)

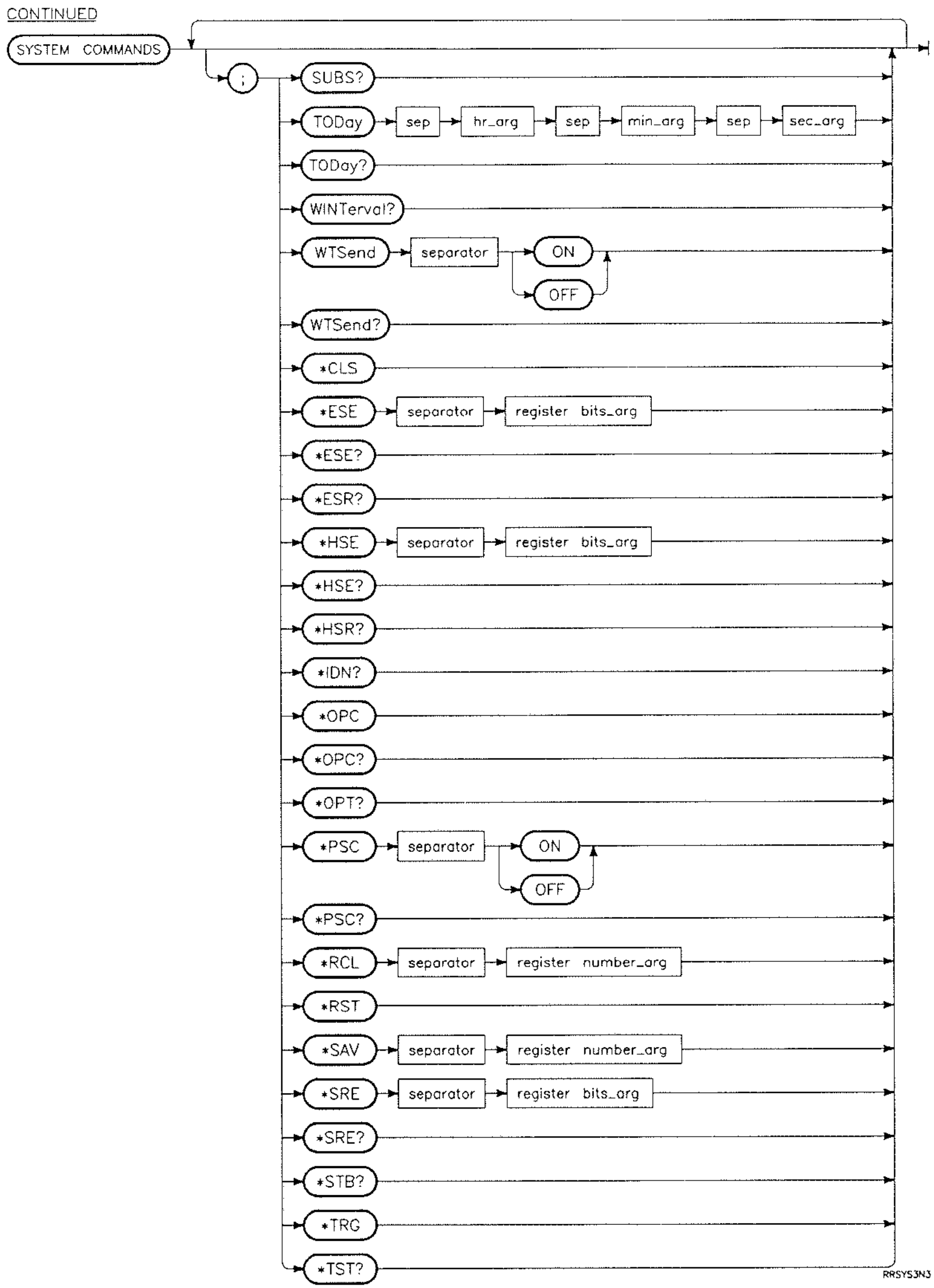


Figure 5-1. System Command Syntax Diagram (3 of 3)

ABOR Abort (command)

Shortform: ABOR [ABORt]
Longform: ABORT

The ABORt command stops a measurement before the measurement cycle is completed. After an ABORt command, you can normally retrieve data by making the HP 5373A a Talker. The effect of the ABORt command depends on your measurement setup. See Appendix E, Programming Examples, for an example program.

Example: OUTPUT 703;"ABOR" - Stops the current measurement process.

BEND? Output Block End List (query only)

Shortform: BEND? [BLock END]
Longform: BEND?

Use the BEND? query to obtain information about the position of the last measurement in each block. While this position is predictable for most types of measurements, there are situations where the block end position (or positions) are not predictable. An example is a Multiple Block setup using Pre-trigger termination. The end of block position could be anywhere.

When there is more than one block end in the acquisition, a list containing the number of blocks and end of block positions is returned. For example, if 5 blocks of 3 measurements were acquired (not Block Averaged), BEND? would return the values 5, 3, 6, 9, 12, 15.

BEND? outputs length information followed by the measurement indices where block ends occurred. For ASCII, the length information is the number of values that follow. For floating point output format, the length information is the number of bytes that follow (this will be 8 times the number of values).

Example: OUTPUT 703;"BEND?" - Queries for the block end list.

BTIM? Output Block Timestamp List (query only)

Shortform: BTIM? [Block TIMestamp]
Longform: BTIME?

Use the BTIMe? query to obtain a list of time intervals from the block arming edge to the first measurement edge of each block. A table of instrument setups that support timing of this interval is provided in *Appendix D, Arming Modes.* For multiple block measurements, which support timing of this interval, a list of time interval values, from the block arming edge (for that block) to the first measurement edge in the block, is provided.

BTIM? outputs length information followed by pairs of values, which indicate the index of the measurement that follows the time interval and the interval value itself. For ASCII, the length information is the number of values that follow. For floating point output format, the length information is the number of bytes that follow (8 times the number of values).

As an example, consider a single block frequency measurement performed using edge holdoff arming. The time from the holdoff edge to the first sample edge is obtained using BTIM? The BTIM? content will be 2 (indicating 2 values to follow), 1 (indicating that next value occurred prior to sample 1), <time value>.

Example: OUTPUT 703;"BTIM?" - Requests the output block timestamp list.

CANC Cancel Hardcopy (command)

Shortform: CANC [CANCel (hardcopy)]
Longform: CANCEL

The CANCel command cancels the hardcopy output initiated by the Print (PRINt) or Plot (PLOT) commands. When the HP 5373A receives the CANCel command, the current printer or plotter output is immediately stopped. Refer to the PRINt and PLOT command descriptions.

Example: The following example shows how to use the CANCel command with an HP 9000 Series 200/300 computer as the controller. Note that sending "OUTPUT 703;CANC" does not work because the HP 5373A is in Talk Only mode while printing or plotting and cannot respond without first being set to Listen mode by the controller.

```
10 SEND 7;UNT MTA LISTEN 3 DATA "CANC",CHR$(13),CHR$(10)
20 SEND 7;UNT TALK 3
30 WRITEIO 7,23;11
40 END
```

CLE Clear (command)

Shortform: CLE [CLEar]
Longform: CLEAR

The CLEar command performs an operation similar to a Device Clear<DCL> or Selected Device Clear <SDC>. In response to either the CLEar,<DCL>, or <SDC> message, the HP 5373A will:

- Clear the input and output buffers.
- Discard all deferred commands and queries.
- Terminate any measurement or acquisition process.

Note that the <DCL> and <SDC> messages (but not the CLEar command) also clear a static failure condition and clear the Event Status and Hardware Status registers.

Example: OUTPUT 703;"CLE" - Terminates measurements, clears input and output buffers, and discards all deferred commands and queries.

CLIS? Carrier Frequency List (query only)

Shortform: CLIS?
Longform: CLIST?

CLIS? returns the carrier frequency value for each block when performing a Phase Deviation (MEAS; FUNC PDEV) or Time Deviation (MEAS; FUNC TDEV) measurement. If the carrier mode is maual (PROC; CARR MAN), the returned value(s) will be the user-entered value (see PROC; CFR command). If the carrier mode is AUTO (PROC; CARR AUTO), the returned value(s) will be computed by the HP 5373A, based upon the data in each block.

CLIS? outputs length information, followed by the carrier value for each consecutive block. For ASCII, the length information is the number of values that follow. For floating point, the length information is the number of bytes that follow (8 times the number of values).

As an example, consider a phase deviation measurement using 2 blocks of 100 measurements (non-averaged). CLIS? would return 2 (indicating 2 values follow), <carrier for block 1>, <carrier for block 2>.

DAT System Clock Date (command/query)

Shortform: DAT [(system clock) DATe]
Longform: DATE

The DATe command sets the system clock date. The format for setting the date is:

DAT, yyyy,mm,dd

where yyyy is the year number (19yy), mm is the month number (1 through 12), and dd is the day number (1 through 31).

The DATe? query returns a string in the format dd mmm yyyy (where mmm is the three-letter month abbreviation rather than a number).

Example: OUTPUT 703;"DAT,1989,8,19" - Sets the system clock date to August 19, 1989.

Example: OUTPUT 703;"DAT?" - Queries for the current system clock date.

DLEN? Data Length (query only)

Shortform: DLEN [Data LENgth]
Longform: DLENGTH

Use the DLENgth? query to identify the number of measurement results that are provided when an ODATa? query is used (or just an enter or transfer follows measurement completion). This query is especially useful prior to ASCII data readout, since there is no length information provided with the measurement output. This result is also useful for pre-triggered or aborted measurements, where the actual number of results may be different than those predicted by computing the Block Size × Block Count product.

Specifically, if the data output is any Non-Hardware histogram function, the data length value is the actual number of measurements that are output. If there are several values read with each measurement (for example Gate Time), you need to take this multiple value output into account.

Another way of looking at the value provided by DLENgth? is that it is the number you would use to terminate a FOR I=1 TO N reading loop, regardless of how many values are entered per loop pass. If the data output is histogram data (any Histogram function), DLENgth? provides the number of bins that are output. This bin value is dependent upon the selected Histogram format (refer to HFORmat command). If you selected COMPressed format, DLENgth? returns the number of non-zero bins, since this is the number of measurement pairs (bin index, bin value) that are output. If you select FULL format, DLENgth? returns the value 2000, since every bin is output. If no measurement data is available at the time DLEN? is executed, a zero will be returned.

Example: OUTPUT 703;"DLEN?" - Queries for the data length.

DMS Default Measurement Setup (command)

Shortform: DMS [Default Measurement Setup]
Longform: DMSETUP

The DMSetup command sets various setup values to default conditions and is equivalent to consecutively pressing the front panel SHIFT and PRESET keys. The default values set by the DMSetup command set a configuration for the current Measurement function that is appropriate for that function. That is, defaults are set for such values as Source channel, Sample Size, Arming (plus associated Arming parameters), as well as Input channel specifications (Trigger Modes, or Levels).

The DMSetup approach to setup is less powerful that PRESET. Accordingly, you may find using DMSetup more useful.

When using DMSetup, general conditions (applying to all measurement types) and specific conditions (applying only to certain measurement types) are set. Appendix B in this manual lists default measurement setups.

Example: OUTPUT 703;"DMS" - Sets default conditions for current measurement function.

DSP Display (command/query)

Shortform: DSP [DiSPlay]
Longform: DSP

The DSP command writes a quoted string, not including quotes, to the status line at the top of each menu screen.

The DSP? query returns the string last written to the status line. The returned string may be one that was written with the DSP command or an internally generated advisory message.

Note that two sets of quotation marks (""...."") are required for sending the display string within the command string.

Examples: OUTPUT 703;"DSP" ""Hello"" " - Displays "Hello" on the status line of the display.

Examples: OUTPUT 703;"DSP?" - Queries the status line of the display.

ERR? Error (query only)

Shortform: ERR? [ERRor?]
Longform: ERROR?

This query returns the next error number from the error queue. The HP 5373A has an error queue that is 16 errors deep and operates on a first-in first-out basis. Successively sending the ERRor? query returns error numbers in the order that they occurred until the queue is empty. Any further queries then return "0" until another error is detected. See Appendix C for a list of error numbers and a description of each error.

Example: OUTPUT 703;"ERR?" - Queries the HP 5373A for the next error number in the error queue.

HOV? Histogram Measurements Over Limit (query only)

Shortform: HOVer? [Histogram measurements OVer (the upper histogram limit)]
Longform: HOVER

The HOVer? query returns the number of histogram measurements that were over the upper histogram limit. A –1 is returned if no Histogram data is available.

Example: OUTPUT 703;"HOV?" - Queries for the number of measurements that were over the upper histogram limit.

HUND? Histogram Measurements Under Limit (query only)

Shortform: HUNDer? [Histogram (measurements) UNDer (the lower histogram limit)]
Longform: HUNDER

The HUNDer? query returns the number of histogram measurements that were under the the lower histogram limit. A 0 is returned if no Histogram data is available.

Example: OUTPUT 703;"HUN?" - Queries for the number of measurements that were under the lower histogram limit.

HWIT? Histogram Measurements Within Limit (query only)

Shortform: HWIThin? [Histogram measurements WIThin (the histogram limits)]
Longform: HWITHIN

The HWIThin? query returns the number of histogram measurements that were within the upper and lower histogram limits. A –1 is returned if no Histogram data is available.

Example: OUTPUT 703;"HWIT?" - Queries for the number of measurements that were within the upper and lower histogram limits.

IDAT? Inhibit Data (query only)

Shortform: IDAT? [Inhibit DATa]
Longform: IDATa?

The IDATa? query returns the inhibit data list so you can locate where inhibits occurred during a measurement. You can read IDATa? in ASCII or Floating Point output formats. Formats follow the standard formats for these modes as described in Chapter 2. For more information about the Inhibit feature refer to the applicable section of the Operating Manual.

IDATa? outputs length information followed by the measurement indices where inhibits occurred. For ASCII, the length information is the number of values that follow. For floating point output format, the length of information is the number of bytes that follow (this will be 8 times the number of values).

As an example, consider a measurement with inhibitted data in locations 5 and 11. The IDAT? content will be 2 (indicating 2 values to follow), 5, 11.

Example: OUTPUT 703;"IDAT?" - Queries for the inhibited data list.

KEY Key Simulation (command/query)

Shortform: KEY [KEY (simulation)]
Longform: KEY

The KEY command simulates pressing a front panel key. Keys may be "pressed" in any order. Invalid keys are ignored.

When using this command, ensure the instrument is in the desired state before sending a KEY command. The key codes and their front-panel function equivalents are listed in *Table 5-2.*

The KEY? query returns the Key Numbers (see *Table 5-2*) of the last keys pressed (including those "pressed" via a KEY command), up to a maximum of 20 numbers. If more than 20 keys have been pressed, only the last 20 key presses are returned; those before the last 20 are lost. If no keys have been pressed (such as after power-up, or if a PRESet is performed), the KEY? query returns "NONE". The KEY? query does not report pressing of the front-panel "Local" key; however, KEY? will report that key's number ("65") if it was "pressed" via a KEY command.

Table 5-2. Key Number Assignments

Key Number	Key Function	Key Number	Key Function
1-23, 30-31	not used	53	EXP
24	softkey 1	54	BACKSPACE
25	softkey 2	55	LAST VALUE
26	softkey 3	56	UP cursor
27	softkey 4	57	RIGHT cursor
28	softkey 5	58	DOWN cursor
29	softkey 6	59	LEFT cursor
32	FUNCTION menu	60	NUMERIC menu
33	MATH menu	61	GRAPHICS menu
34	INSTRUMENT STATE menu	62	STATUS menu
35	TEST menu	63	PRINT
36	INPUT menu	64	SHIFT
37	PRE-TRIGGER menu	65	LOCAL
38	SYSTEM menu	66	RESTART
39	HELP menu	67	SINGLE/REPET
40-49	digits 0-9	68	MANUAL ARM (see note)
50	. (Decimal point)	69	SAVE
51	+/– (plus/minus)	70	RECALL
52	ENTER	71	PRESET

Examples: OUTPUT 703;"KEY,39" - "Presses" the HELP menu key.

OUTPUT 703;"KEY?" - Queries for the last 20 keys that were pressed.

NOTE

The MANUAL ARM key provides two functions:

Provides manual control of the measurement gate for Totalize measurements using MANUAL arming mode. This is the only measurement and arming mode for which this key performs a manual arm function.

Aborts a measurement in progress. The ABORt function is performed for all measurement and arming modes other than Totalize measurements using MANUAL arming mode. This function is identical to using the ABORt command.

LOC Local (command)

Shortform: LOC [LOCal]
Longform: LOCAL

The LOCal command returns the instrument to Local (front panel) operation. The LOCal command performs a similar operation to the Clear Lockout/Set Local HP-IB message. This command is provided for controllers with limited HP-IB control capability. The HP-IB Clear Lockout/Set Local message is the preferred method of switching the instrument from Remote to Local and clearing the Local Lockout.

Example: OUTPUT 703;"LOCAL" - Switches the HP 5373A from Remote to Local and clears the Local Lockout condition.

NOTE

This command should be used in conjunction with the REMOTE command (OUTPUT 703;"REM").

MEN
Select Menu
(command/query)

Shortform: MEN [(select) MENu]
Longform: MENU

Use the MENu command to select one of 11 different menus for display on the screen. The MENu command parameters and their corresponding screen menus are:

Parameter	Screen Menu Displayed
PRETrigger	Pre-trigger
FUNCtion	Function
GRAPhics	Graphics
HELP	Help
INFo	Status
INPut	Input
Instrument STate	Instrument State
MATH	Math
NUMeric	Numeric
SYSTem	System
TEST	Diagnostics

The MEN? query returns the name of the currently displayed menu screen. If "Display Blank" is the current screen, MEN? will return "error" (also, Error 150 will be reported).

Examples: OUTPUT 703;"MEN,GRAP" - Displays the Graphics screen.

OUTPUT 703;"MEN?" - Queries for the currently displayed menu screen.

MMOD
Measurement Format
(command/query)

Shortform: MMOD [Measurement MODe]
Longform: MMODe

The MMODe command sets the sample mode to either FAST or NORMal. FAST sample mode provides a higher maximum measurement rate, but has more restrictive numeric limits for some parameters (compared to NORMal sampling mode). Refer to the Operating Manual for an explanation of the limits.

When FAST is selected, the amount of data output when in BINary output mode is reduced. (Using binary output is explained in Chapter 3, *Binary Output* in the *Special Topics* section.) The benefit of using FAST is that data transfer times are reduced compared with NORMal. Select BINary output mode using the OUTPut command.

The MMODe? query returns the current Measurement mode.

NOTE

The Totalize function always uses NORMal sampling mode. However, if the MMODe state is FAST when Totalize is selected, the MMODe state is not forced to NORMal. This way, you do not have to remember to re-program the MMODe state when the function is later changed to something other than Totalize. If MMODe is queried (MMODe?) when the function is Totalize, the response is UNAV (unavailable). This response is an indication that the actual state of MMODe is not meaningful for the Totalize function.

Parameters: {FAST | NORMal}

Examples: OUTPUT 703;"MEAS;MMOD,FAST" - Sets the measurement mode to FAST.

OUTPUT 703;"MEAS;MMOD?" - Queries for the currently selected measurement mode.

ODATa? Output Measurement Data Results (query only)

Shortform: ODATa? [Output (measurement) DATa? (result)]

Longform: ODATa?

The ODATa? query requests measurement data output. The type of data you receive is dependent upon the selected measurement function. For Histogram functions (HCTinterval, HPMTinterval, HTIMe) the output is histogram data (refer to the OHIStogram command for more information about histogram output.) For all other functions, the output is the list of measurement results, as described in Chapter 2.

The ODATa? query is useful for retrieving measurement results any time after the measurement has completed. In many situations, ODATa? is unnecessary because the HP 5373A automatically makes the measurement results available upon measurement completion. The value of ODATa? is that it provides a user-directed method for requesting measurement results.

Use ODATa? with ASCII or Floating Point output formats. If no results are available when ODATa? executes, a zero is returned (#6000000 for floating point, ASCII 0 for ASCII).

Example: OUTPUT 703;"ODATa?" - Queries for measurement data.

OHIS? Output Histogram Result (query only)

Shortform: OHIS? [Output HIStogram (result)]
Longform: OHIStogram?

The OHIStogram? query returns the histogram result over HP-IB. It can be used to obtain hardware- or software-produced histogram results. Either ASCII or floating point output formats may be used. The content of the OHIS? query is dependent on the selected histogram format (see HPFORmat command in Interface section of this chapter).

Example: OUTPUT 703;"OHIS?" - Queries for the histogram result.

PLOCation? Pre-trigger Location (query only)

Shortform: PLOCation? [Pre-trigger LOCation]
Longform: PLOCATION

Use the PLOCation? query to identify the index position of the Pre-trigger, if one exists. The interpretation is that the index returned is the first sample following the detection of the Pre-trigger. If the measurement cannot produce a Pre-trigger (for example, termination occurs by measurement count), a zero is returned. If a Pre-trigger occurred, but the 5373A could not identify the position, a value of –1 is returned.

Example: OUTPUT 703;"PLOC?" - Queries for the index position of the pre-trigger.

PLOT Plot Screen (command)

Shortform: PLOT [PLOT (screen)]
Longform: PLOT

The PLOT command copies the currently displayed graph to any attached HP plotter that supports HP-GL (Hewlett-Packard Graphics Language). This command is available only for Graphics screens, and is equivalent to pressing the PLOT GRAPH (SHIFT, PRINT) front panel keys. The CANCel (hardcopy) command can be used to abort the current plot output.

To use the PLOT command, select DISPlay data as the print source (refer to the PSOurce command description).

Example: The following example demonstrates a method for using the PLOT command:

```
10 SEND 7;MTA LISTEN 3 DATA "PLOT",CHR$(13),CHR$(10)
20 SEND 7;UNL
30 SEND 7;LISTEN 1
40 SEND 7;TALK 3
50 WRITEIO 7,23;11
60 END
```

POD?
Pods Installed
(query only)

Shortform: POD? [PODs (installed)]
Longform: POD?

The POD? query returns a string indicating which input pods or rear panel options are currently installed (if any). The string returned is:

"HPnnnnnA,HPnnnnnA"

where nnnnn is the Hewlett-Packard model number: 53702, 54001, 54002, or 54003. The first name in the string is the Channel A pod, the second name is the Channel B pod. If a respective pod slot is empty, "NONE" is returned in the appropriate string position.

Example: OUTPUT 703;"POD?" - Queries for the currently installed input pods.

PRES
Preset Instrument
(command)

Shortform: PRES [PRESet (instrument)]
Longform: PRESET

The PRESet command resets the instrument to default settings. This command performs the same function as the *RST command.

Refer to *Table 5-3* for a list of the HP 5373A preset conditions. Note that the PRESet command clears the key queue, but does not clear the error queue.

Example: OUTPUT 703;"PRES" - Presets the HP 5373A to default conditions.

Table 5-3. Preset Conditions

FUNCTION, MODE, OR VALUE	PRESET STATE
Measurement Function	PRF (if HP 53702A Envelope Detector Pod is installed in CHANNEL A) or Frequency
Measurement Channel	A
Block Size	100
Block Count	1
HW Histogram Start	0 S
HW Histogram Binwidth	200 pS
HW Histogram Span	400 nS
HW Histogram Center	200 nS
HW Histogram Block Arm	Fast Arm
Arming	Automatic
Start channel	A
Start channel slope	positive
Stop channel	A
Stop channel slope	positive
Start delay events	1
Start delay time	200 ns
Start delay channel	A
Stop delay events	1
Stop delay interval	10 μs
Stop delay time	1 s
Stop delay cycles	16
Stop delay channel	A
Measurement mode	Normal
Segmentation	Off
TV Averaging	On
Inhibit	Off
Inhibit Threshold	TTL
Inhibit Sense	Above
TI Detect region	Above
TI Detect lower value	0 ns
TI Detect middle value	0 ns
TI Detect upper value	0 ns
Pre-trigger	Off
Graphics Presets	
Displayed Graph	Histogram
Active marker selection	Vertical, Black
X-axis Manual Scale status	Off
Y-axis Manual Scale status	Off
Update status	While (update between blocks)
Connect status	Off
Y scale status (Histogram)	Linear

Table 5-3. Preset Conditions (Continued)

FUNCTION, MODE, OR VALUE	PRESET STATE
Outline status (Histogram)	On
Grid status	Off
View channel (for dual-channel measurements)	A
Marker Display Mode	Marker
Marker Next Mode	Pixel
Marker Min tracking status	Inactive
Marker Max tracking status	Inactive
Graphics menu level	Main
X and Y manual scaling parameters set to:	
Histogram	
Xmin	0
Binwidth	200 ps
Ymax	100
Time Variation	
Xmin	0
Xmax	0
Ymin	0
Ymax	0
Event Time	
Xmin	0
Xmax	0

PRIN Print Screen (command)

Shortform: PRIN [PRINt (screen)]
Longform: PRINT

The PRINt command copies either measurement results or a bit-map of the currently displayed menu to any attached HP printer with graphics capability. This command is available for all menu screens, and is equivalent to pressing the front panel PRINT key. The CANCel (hardcopy) command is available to abort the print output.

To use the PRINt command, use the PSOurce (Print SOurce) command to select DISPlay for a copy of the screen menu, or MEASurement for a copy of the measurement results.

Example: The following example shows a method for using the PRINt command:

```
10 SEND 7;MTA LISTEN 3 DATA "PRIN",CHR$(13),CHR$(10)
20 SEND 7;UNL
30 SEND 7;LISTEN 1
40 SEND 7;TALK 3
50 WRITEIO 7,23;11
60 END
```

REM Remote (command)

Shortform: REM [REMote]
Longform: REMOTE

The REMote command sets the HP 5373A to the Remote mode and sets the Local Lockout. The REMote command performs an operation similar to the HP-IB Remote message followed by the HP-IB Local Lockout message. This command is provided for use by controllers that have a limited HP-IB capability. If available, the HP-IB Remote and Local Lockout messages are the preferred method of switching the HP 5373A from Local to Remote and setting Local Lockout. If the REN (remote enable) control line is true, the HP 5373A REMote command will have no effect.

Example: OUTPUT 703;"REM" - Switches the HP 5373A from Local to Remote and sets Local Lockout.

NOTE

This command should be used in conjunction with the LOCAL command (OUTPUT 703;"LOC").

REST Restart Measurement (command)

Shortform: REST [RESTart (measurement)]
Longform: RESTART

The RESTart command performs the same function as the front panel RESTART key. When received, this command restarts the measurement process and clears cumulative results and error messages (the error queue is cleared). The RESTart command will not clear the key queue and will have no effect on a previously entered data value.

Example: OUTPUT 703;"REST" - Tells the HP 5373A to restart a measurement.

SET Instrument Setup (command/query)

Shortform: SET [(instrument) SETup]
Longform: SETUP

The SETup command and SETup? query are used to send and receive a learn string from the instrument. The SETup? query returns a learn string containing the instrument setup, in block data format, to the controller (the returned sequence of bytes must be saved in an array). The returned string uses the same format as required by the SETup command, thus no modifications need be made to the string between the time it is received after the SETup? query and the time it is sent back to the instrument using the SETup command.

Sending the learn string to the instrument with the SETup command is similar to recalling an instrument setup using the *RCL command. Note that the SETup? and SETup commands should only be used by experienced programmers. The *SAV and *RCL commands are easier to use for saving and recalling instrument setups.

The SETup command restores the settings defined by the array of bytes returned by the SETup? query. The SETup command must be followed by a carriage return <CR> and line feed <LF> to signal the instrument that the next data to follow on the bus is the byte sequence. The setup byte sequence must immediately follow; an error occurs if something else (another command or some other form of data) is sent.

The setup data is sent as a binary block in the following form:

```
<#><non-zero digit><length_word> <DAB>...<DAB>^END
```

The <non-zero digit> is a single ASCII numeral specifying the number of words in <length_word>.

The <length_word> is an unsigned binary integer that is x digits long (where x is specified by the <non-zero digit>) representing the number of Data Bytes (DAB).The SET? query causes the HP 5373A to transmit the current measurement setup to the external controller in the form of a binary block of data. The binary block is in the form described above, except that the <non-zero digit> is always the number 6 and the <length_word> is always 6 digits long.

Examples: The program example (*Figure* 5-2) demonstrates how to use the SETup? query and SETup command to save and restore an instrument setup. Also see *Teach/Learn* Program Example in Appendix E.

```
10   !Example of learn mode using the SET? query and the SET command
20   !
30   !The SET? query reads information from the 5373A that specifies
40   !all instrument settings for a particular measurement.
50   !The SET command sends the same information that was received
60   !by the SET? query to the instrument--thereby "remembering" the
70   !instrument setups.
80   !
90   !THE "SET?" AND "SET" COMMANDS SHOULD ONLY BE USED BY THE
100  !EXPERIENCED HP-IB PROGRAMMER. THE "SAVE" AND "RECALL"
110  !COMMANDS ARE EASIER TO USE AND MAY SERVE YOUR NEEDS.
120  !
130  OPTION BASE 1
140  Isc=7
150  Analyzer=703
160  CLEAR Isc
170  OUTPUT Analyzer;"CLE;PRES"
180  PRINT "MANUALLY SET THE 5373A TO THE DESIRED SETTINGS,"
190  PRINT "PRESS CONTINUE WHEN FINISHED.................."
200  PAUSE
210  ASSIGN @Analyzer TO 703;FORMAT OFF
220  ASSIGN @Setup_buffer TO BUFFER [500]
230  OUTPUT Analyzer;"SET?"
240  ENTER Analyzer USING "%,5A";A$
260  TRANSFER @Analyzer TO @Setup_buffer;END,WAIT
261  STATUS @Setup_buffer,4;Bytes_received
270  PRINT "MANUALLY CHANGE THE 5373A TO DIFFERENT SETTINGS,"
280  PRINT "PRESS CONTINUE AND CHECK FOR CORRECT SETTINGS."
290  PAUSE
300  OUTPUT Analayzer;"SET"
310  Header$="#3"&VAL$(Bytes_received)
320  CONTROL @Setup_buffer,5;1
330  OUTPUT Analyzer USING "#,5A";Header$
340  TRANSFER @Setup_buffer TO @Analyzer
350  PRINT "SET COMMAND FINISHED"
360  END
```

Figure 5-2. Program Example (Using the SET? Query)

SMOD
Sample Mode
(command/query)

Shortform: SMOD [Sample MODe]
Longform: SMODE

The SMODe command is used to determine how often the measurement is displayed. This command is equivalent to the front panel SINGLE/REPET key function. When you select the REPetitive sample mode the instrument makes measurements as quickly as possible. Selecting the SINGle sample makes the instrument hold off a measurement indefinitely, until triggered. In this case, the instrument displays the previous measurement and halts until one of the following occurs:

- The HP 5373A receives the *TRG or GET (Group Execute Trigger) command. This is available via HP-IB only.
- The HP 5373A receives the RESTart (or PRESet) commands. This is available via HP-IB or Front Panel.

The SMODe? query returns the currently selected Sample Mode.

Parameters: {SINGle | REPetitive}

Examples: OUTPUT 703;"SMOD,SING" - Tells the HP 5373A to display previous measurement and halt until triggered.

OUTPUT 703;"SMOD?" - Queries for the currently selected sample mode.

SUBS? Subsystem (query only)

Shortform: SUBS? [SUBSystem?]
Longform: SUBS?

The SUBSystem? query returns the currently selected subsystem. The string returned is one of the following:

"DIAG"	- Diagnostics subsystem
"GRAP"	- Graphic subsystem
"INP"	- Input subsystem
"INT"	- Interface subsystem
"IST"	- Instrument State subsystem
"MEAS"	- Measurement subsystem
"NUM"	- Numeric subsystem
"PROC"	- Process subsystem

Example: OUTPUT 703;"SUBS?" - Queries the HP 5373A for the currently active subsystem.

TOD System Clock Time of Day (command/query)

Shortform: TOD [Time Of Day]
Longform: TODAY

The TODay command sets the system time of day. The format is:

TOD,hh,mm,ss

where hh is the hour number (0 through 23), mm is the minutes number (0 through 59), and ss is the seconds number (0 through 59).

The TODay? query returns the current system time in the format hh:mm:ss.

Examples: OUTPUT 703;"TOD,11,20,15" - Sets the HP 5373A system clock to 11:20 (AM) plus 15 seconds.

OUTPUT 703;"TOD?" - Queries for the current setting of the system clock.

WINT? Warmup Interval (query only)

Shortform: WINT? [(has) Warmup INTerval (elapsed)]
Longform: WINTERVAL

The WINTerval query returns the state of the warmup interval. A zero indicates the warmup interval has not elapsed; a one indicates the warmup interval has elapsed.

Example: OUTPUT 703;"WINT?" - Queries for the state of the warmup interval.

WTS Wait to Send (command/query)

Shortform: WTS [Wait To Send]
Longform: WTSEND

The WTSend command controls the placement of measurement results into the output buffer. When the Wait To Send mode is off, and more than one block of measurements is requested, only the last block of measurement data is placed in the output buffer. When Wait To Send mode is on, each block of measurement data is placed in the output buffer; the next block is not started until the last block has been sent over the bus.

In Repetitive sample mode (SMOD, REP), with Wait To Send off, the next measurement starts as soon as the current measurement data is placed in the output buffer. When Wait To Send is on, the next measurement will start only after the current measurement data has been sent out over the HP-IB.

The WTSend? query returns the currently selected Wait To Send mode.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"WTS,ON" - Turns on the Wait To Send mode.

OUTPUT 703;"WTS?" - Queries for the on/off status of the Wait To Send mode.

NOTE

*If the total acquisition (Number of Blocks * Number of Measurements in each Block) is less than the HP 5373A's memory, all blocks are completed before any data is output, regardless of the WTS mode. Use Appendix A as a guide for determining the maximum number of measurements that will fit into memory for multiple-block acquisitions.*

WTS only has an effect as long as the HP 5373A is in REMOTE.

*CLS Clear Status (command)

Shortform: *CLS [CLear Status]
Longform: *CLS

The *CLS command is used to clear the Event Status Register and the Hardware Status Register, setting all bits in these registers to zero. When the Event and Hardware Status registers are cleared, the corresponding summary messages are also cleared, thus indirectly clearing bits 0 and 5 in the Status Byte Register.

Example: OUTPUT 703;"*CLS" - Clears Event and Hardware Status registers.

*ESE Event Status Enable (command/query)

Shortform: *ESE [Event Status Enable]
Longform: *ESE

The *ESE command is used to set selected bits of the Event Status Enable register. An enabled (set to 1) bit in the Enable register is ANDed with its corresponding bit in the Event Status Register to generate a Service Request if an enabled condition occurs. To enable the register bits, send the *ESE command with an integer parameter (range 0 to 255) representing the binary-weighted values of the bits to be set. For example, to set bits 2 and 4, send the decimal integer 20 (4 + 16) as the parameter.

The *ESE? query returns an integer (NR1 format) that is the decimal equivalent of the binary-weighted values of the bits that are currently set to 1.

Refer to chapter 2 for information about the Event Status Register and Event Status Enable Register.

Range: 0 to 255

Examples: OUTPUT 703;"*ESE,36" - Sets bits 2 and 5 of the Event Status Enable register, thus enabling bit 2 (Query Error) and bit 5 (Command Error) of the Event Status Register.

OUTPUT 703;"*ESE?" - Queries for the contents of the Event Status Enable register.

*ESR? Event Status Register (query only)

Shortform: *ESR? [Event Status Register?]
Longform: *ESR?

The *ESR? query returns the contents of the Event Status Register. The value returned is an integer (NR1 format) that is the decimal equivalent of the binary-weighted values of the register bits. For example, a value of 36 indicates that bit 2 (Query Error) and bit 5 (Command Error) are set to 1. Upon reading the Event Status Register, all bits in the register are cleared (set to 0).

Example: OUTPUT 703;"*ESR?" - Queries for the contents of the Event Status Register, and clears the register.

*HSE Hardware Status Enable (command/query)

Shortform: *HSE [Hardware Status Enable]
Longform: *HSE

The *HSE command is used to set selected bits of the Hardware Status Enable register. An enabled (set to 1) bit in the Enable register is ANDed with its corresponding bit in the Hardware Status Register to generate a Service Request if an enabled condition occurs. To enable the register bits, send the *HSE command with an integer parameter (range 0 to 255) representing the binary-weighted values of the bits to be set. For example, to set bits 2 and 4, send the decimal integer 20 (4 + 16) as the parameter.

The *HSE? query returns an integer (NR1 format) that is the decimal equivalent of the binary-weighted values of the bits that are currently set to 1.

Refer to Chapter 2 for information about the Hardware Status Register and Hardware Status Enable Register.

Range: 0 to 255

Examples: OUTPUT 703;"*HSE,80" - Sets bits 4 and 6 of the Hardware Status Enable Register, thus enabling bit 4 (Time Base Error) and bit 6 (Power-On Failure) of the Hardware Status Register.

OUTPUT 703;"*HSE?" - Queries for the contents of the Hardware Status Enable register.

*HSR? Hardware Status Register (query only)

Shortform: *HSR? [Hardware Status Register?]
Longform: *HSR?

The *HSR? query returns the contents of the Hardware Status Register. The value returned is an integer (NR1 format) that is the decimal equivalent of the binary-weighted values of the register bits. For example, a value of 80 indicates that bit 4 (Time Base Error) and bit 6 (Power-On Failure) are set to 1. Upon reading the Hardware Status Register, all bits in the register are cleared (set to 0).

Example: OUTPUT 703;"*HSR?" - Queries for the contents of the Hardware Status Register, and clears the register.

*IDN? Instrument Identification (query only)

Shortform: *IDN? [(instrument) IDeNtification]
Longform: *IDN?

The *IDN? query returns a string containing the model number and firmware revision code in the form:

"Hewlett Packard, 5373A,0,xxxx"

- where "xxxx" is the datecode of the installed firmware revision.

Example: OUTPUT 703;"*IDN?" - Queries for the instrument model number and firmware revision code.

*OPC Operation Complete (command/query)

Shortform: *OPC [OPeration Complete]
Longform: *OPC

The *OPC command causes the instrument to generate the Operation Complete message (OPC, bit 0) in the Event Status Register when all pending selected device operations have been finished.

The *OPC? query returns a "1" when an operation is complete. Note that the value returned will always be "1" because the *OPC query is not parsed until all previous commands have been completed.

Examples: OUTPUT 703;"*OPC" - Set the OPC bit in the Event Status Register to 1 when all operations are finished.

OUTPUT 703;"*OPC?" - Queries for operation completion.

*OPT? Installed Options (query only)

Shortform: *OPT? [(installed) OPTions?]
Longform: *OPT?

The *OPT? query returns a string indicating which options are installed in the HP 5373A. The string returned is either "NONE" (no options installed) or the option number. For example, "030"(Option 030 Rear Panel Inputs installed).

Example: OUTPUT 703;"*OPT?" - Queries for installed options.

*PSC Power-On Status Clear (command/query)

Shortform: *PSC [Power-on Status Clear]
Longform: *PSC

The *PSC command controls the automatic power-on clearing of the Service Request Enable register, Event Status Enable register, and the Hardware Status Enable register. Sending the *PSC command with any number that rounds to a non-zero value causes the HP 5373A to clear (set to 0) all bits in the registers at power-on. Sending "*PSC,0" allows the HP 5373A to send a Service Request at power-on if required (and if the appropriate register bits have been enabled).

The *PSC? query returns the value of the Power-On-Clear flag. A returned value of "0" indicates that the Service Request Enable, Event Status Enable, and Hardware Status Enable registers will retain their status when power is restored to the instrument. A returned value of "1" indicates that the three registers are cleared when power is restored.

Parameters: {0 | (any non-zero number)}

Examples: OUTPUT 703;"*PSC,1" - Clear the Service Request Enable, Event Status Enable, and Hardware Status Enable registers at power-on.

OUTPUT 703;"*PSC?" - Queries for the value of the power-on clear flag.

*RCL Recall Register (command)

Shortform: *RCL [ReCaLl (register)]
Longform: *RCL

The *RCL command restores the instrument to a previously saved configuration from a specified save/recall register. This command is equivalent to the front panel RECALL key function.

Registers 0 through 9 may be specified. Specifying Register 0 recalls the instrument setup that existed just prior to invoking the PRESet or DMSetup functions. Refer to the *SAV command description for information about saving to the registers.

Parameters: {0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

Example: OUTPUT 703;"*RCL,1" - Recall a setup from save-recall register 1.

*RST Reset (command)

Shortform: *RST [ReSeT]
Longform: *RST

The *RST command resets the instrument to the default settings. This command performs the same function as the PRESet command. Note that the *RST command clears the key queue, but does not clear the error queue.

Example: OUTPUT 703;"*RST" - Resets the HP 5373A to default conditions.

*SAV Save Register (command)

Shortform: *SAV [SAVe (register)]
Longform: *SAV

The *SAV command saves an instrument setup to a specified save/recall register. This command performs the same function as the front panel SAVE key. All instrument settings, except the HP-IB configuration, are saved. Registers 1 through 9 may be specified.

Register 0 cannot be specified because it is reserved for automatically storing the instrument setup that existed just prior to invoking the PRESet or DMSetup functions. Register 0 can be recalled (refer to the *RCL command description for information about recalling registers).

Parameters: {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

Example: OUTPUT 703;"*SAV,1" - Saves the current setup in register 1.

*SRE Service Request Enable (command/query)

Shortform: *SRE [Service Request Enable]
Longform: *SRE

The *SRE command sets the bits of the Service Request Enable register. The command must be sent with a decimal number representing the binary-weighted values of the bits to be set. For example, sending "*SRE,32" sets bit 5 of the register to 1, thus enabling bit 5 (Event Status Bit) of the Status Byte register to generate a Service Request if an event occurs.

The *SRE? query returns the value of the bits in the Service Request Enable register. The returned value is a decimal number representing the binary-weighted value of the register bits.

Refer to chapter 2 for information about using the Status Byte and Service Request Enable registers.

Range: 0 to 255

Examples: OUTPUT 703;"*SRE,16" - Sets bit 4 of the Service Request Enable register to 1, thus enabling bit 4 (Message Available) of the Status Byte register.

OUTPUT 703;"*SRE?" - Queries the HP 5373A for the contents of the Service Request Enable register.

*STB? Read Status Byte (query only)

Shortform: *STB? [(read) STatus Byte]
Longform: *STB?

The *STB query is used to read the Status Byte register and the Master Summary Status (MSS) bit. The returned value is an integer representing the binary-weighted values of the register bits. For example, a returned value of "32" indicates that bit 5 (Event Status Bit) of the Status Byte register is set to 1. Sending the *STB query does not alter the contents of the register.

Example: OUTPUT 703;"*STB?" - Queries for the contents of the Status Byte register.

*TRG Trigger (command)

Shortform: *TRG [TRiGger]
Longform: *TRG

The *TRG command is used to trigger a measurement. This command, with one exception, has the same effect as the GET (Group Execute Trigger) command. The exception is for Manual Arm (Totalize) measurements, where the *TRG command performs the same function as the front panel MANUAL ARM key.

Example: OUTPUT 703;"*TRG" - Tells the HP 5373A to take a measurement.

*TST? Self-Test (query only)

Shortform: *TST? [(self) TeST]
Longform: *TST?

The *TST? query initiates a series of self tests that exercise various parts of the HP 5373A. Issuing this command is equivalent to selecting the Self Test option in the Diagnostic Test screen. All tests occur automatically and do not affect data stored in RAM. The *TST? query returns only numbers, 0 for pass or a test number for fail. It does not return ASCII messages as TEST? does.

Example: OUTPUT 703;"*TST?" - Initiates the self test routine and returns the pass/fail result.

NUMERIC SUBSYSTEM COMMANDS

The NUMeric subsystem controls the format for displaying measurement result data. Functions in this subsystem are equivalent to those available via the NUMERIC menu screen. The syntax diagram for the NUMeric subsystem commands are shown in *Figure 5-3* .

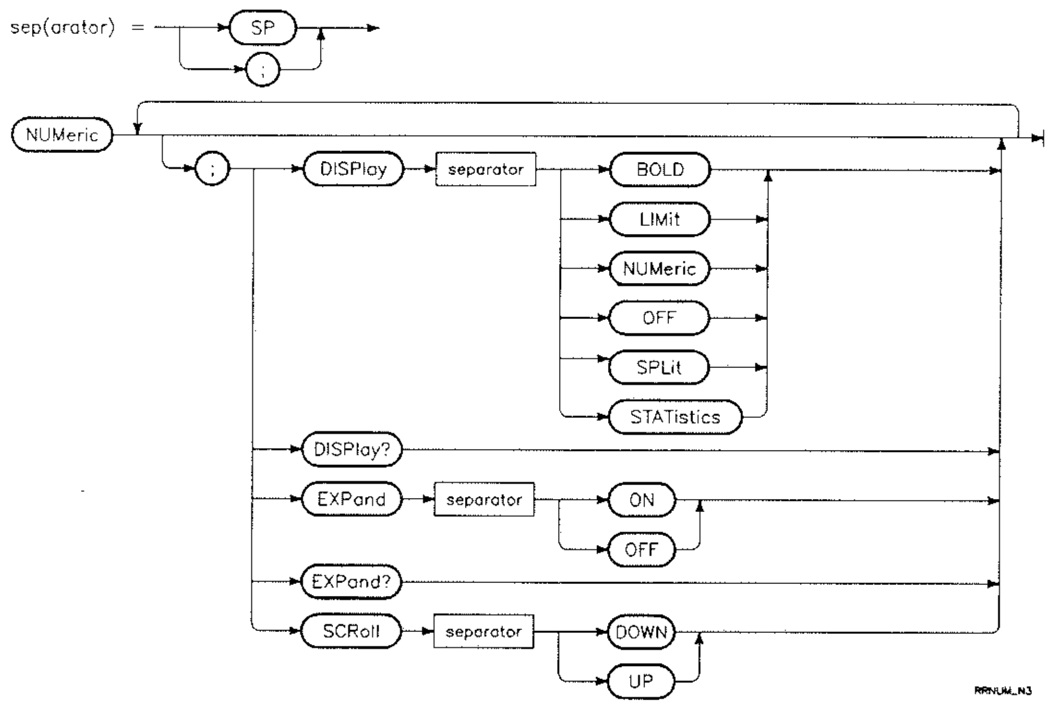


Figure 5-3. NUMeric Subsystem Syntax Diagram

DISP
Display Type
(command/query)

Shortform: DISP [DISPlay (type)]
Longform: DISPLAY

The DISPlay command selects the type of numeric display for screening the results of a measurement. This command also determines the type of data that is returned over the bus, which corresponds to the type of data displayed on the screen. The parameters are:

BOLD	Selects a bold font for displaying measurement results in larger, easier-to-read characters.
LIMit	Displays results for limit testing measurements.
NUMeric	Displays measurement results.
OFF	Turns off the numeric result display and goes to the "DISPLAY BLANK" screen. Send a MENU command such as: "MENU,NUM", to re-display the screen.
SPLit	Displays numeric results and statistical results (Maximum, Mean, Minimum, and Standard Deviation only) on the screen simultaneously.
STATistics	Displays all eight statistical results on the screen (Allan Variance, Maximum, Mean, Minimum, Root Allan Variance, Root Mean Square, Standard Deviation, and Variance).

The DISPlay? query returns the currently selected display type.

Parameter: {BOLD | OFF | LIMit | NUMeric | SPLit | STATistics}

Examples: OUTPUT 703;"NUM;DISP,BOLD" - Tells the HP 5373A to display measurement results in large character format.

OUTPUT 703;"NUM;DISP?" - Queries for the currently selected display type.

EXP
Expand data
(command/query)

Shortform: EXP [EXPand (data)]
Longform: EXPAND

The EXPand command causes the HP 5373A to display expanded results data on the screen for each measurement taken. The expanded data is either Gate Time or Missed Events, depending on the type of measurement being performed.

The EXPand? query returns the status of the expanded data format.

Examples: OUTPUT 703;"NUM;EXP,ON" - Tells the HP 5373A to display gate time or missed events data for each measurement.

OUTPUT 703;"NUM;EXP?" - Queries for status of the expanded data format.

SCR
Scroll Results
(command)

Shortform: SCR [SCRoll (results)]
Longform: SCROLL

The SCRoll command scrolls the results on the measurement result screen up or down one measurement at a time. Specifying "DOWN" scrolls the results toward previous measurements, while specifying "UP" scrolls toward later measurements.

Example: OUTPUT 703;"NUM;SCR,DOWN" - Scrolls the HP 5373A numeric result screen toward previous measurements.

Graphic

GRAPHIC SUBSYSTEM COMMANDS

The GRAPhic subsystem commands control display of measurement results. Choose from:

- Event Time (ETIM)
- Histogram (HIST)
- Time Variation (TVAR)

Multisheet *Figure 5-4* illustrates the graphics subsystem syntax (including minor sublevels MEMory and MGRaph). Major sublevels that are collapsed in the top level appear on separate syntax diagrams for the following:

- Event Time (*Figure 5-5*)
- Histogram (*Figure 5-6*)
- Time Variation (*Figure 5-7*)

NOTE

For most commands, "1" and "0" will be accepted interchangeably for "ON" and "OFF". However, this is not true for commands of the GRAP subsystem. For these commands "ON" and "OFF" must be used. Using "1" or "0" will cause an error.

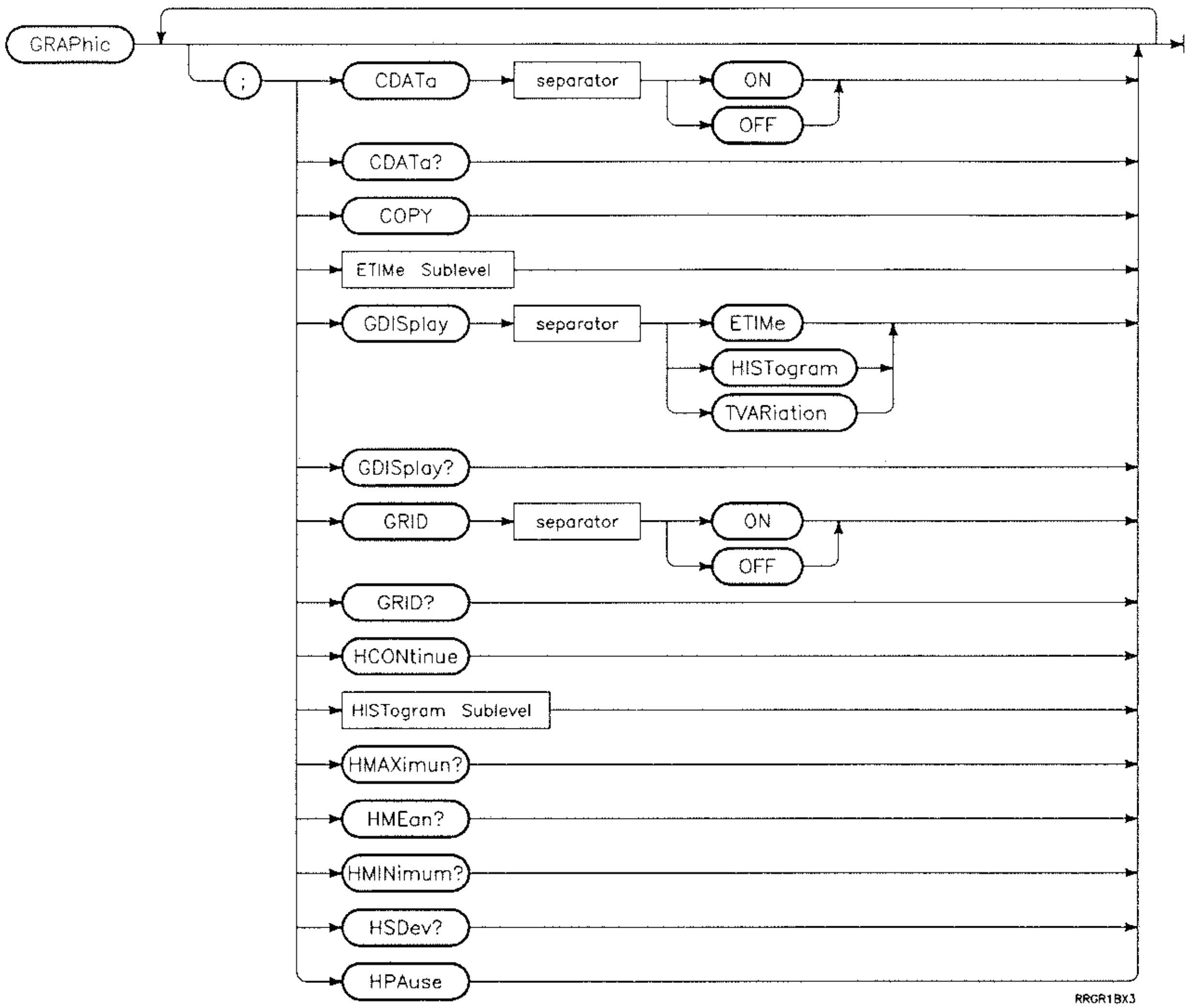


Figure 5-4. GRAPhic Subsystem Syntax Diagram (1 of 4)

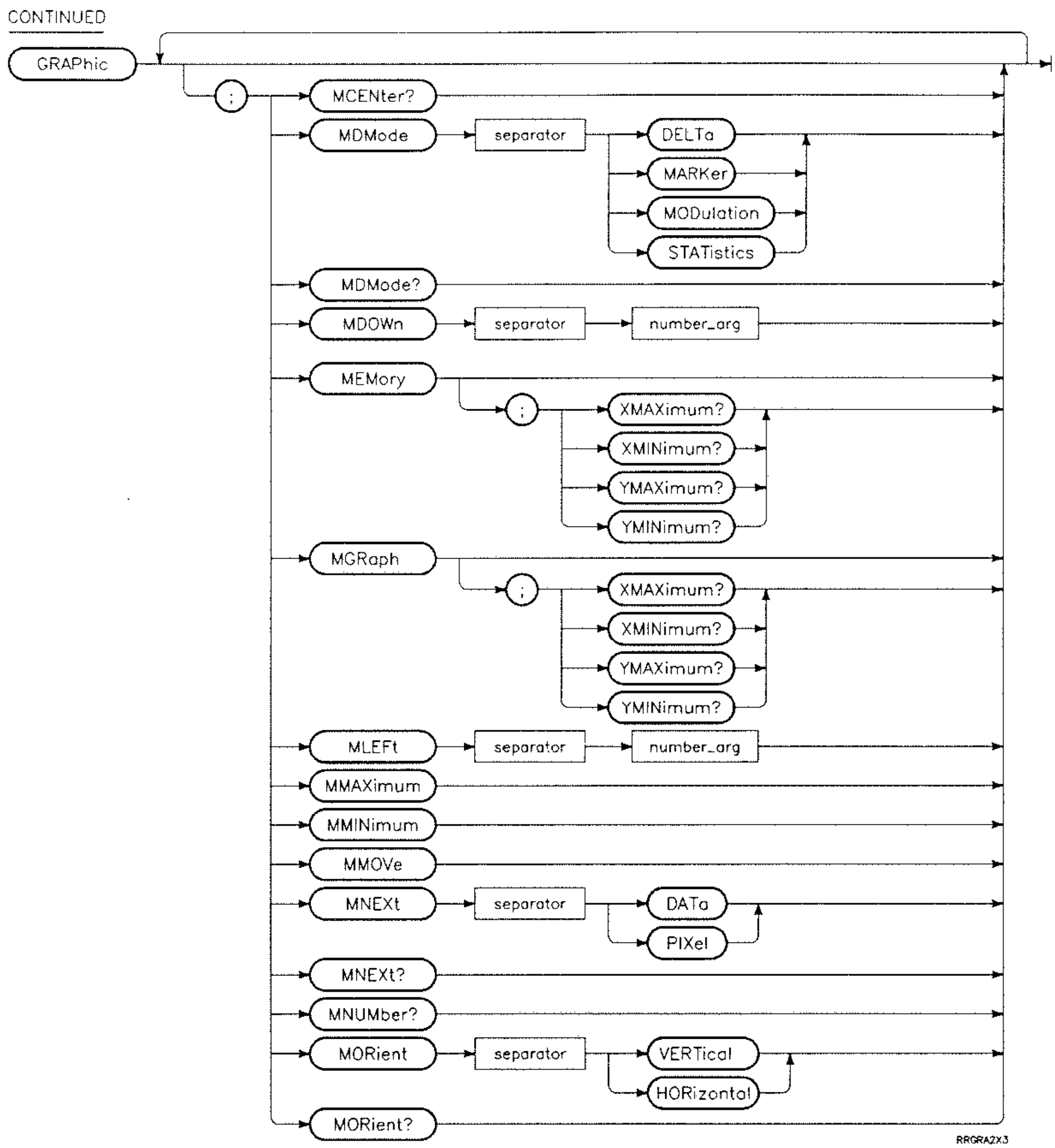


Figure 5-4. GRAPhic Subsystem Syntax Diagram (2 of 4)

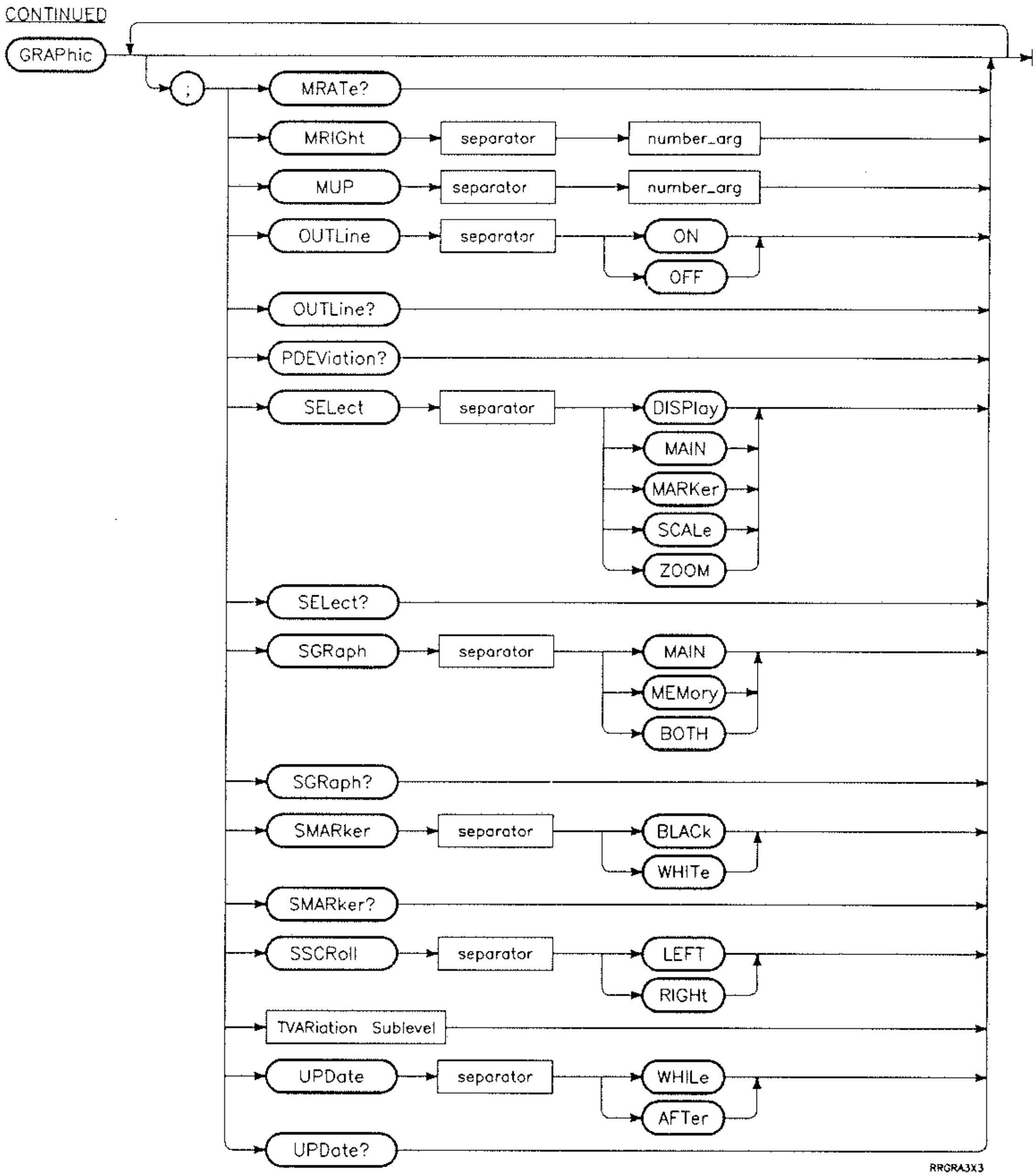


Figure 5-4. GRAPhic Subsystem Syntax Diagram (3 of 4)

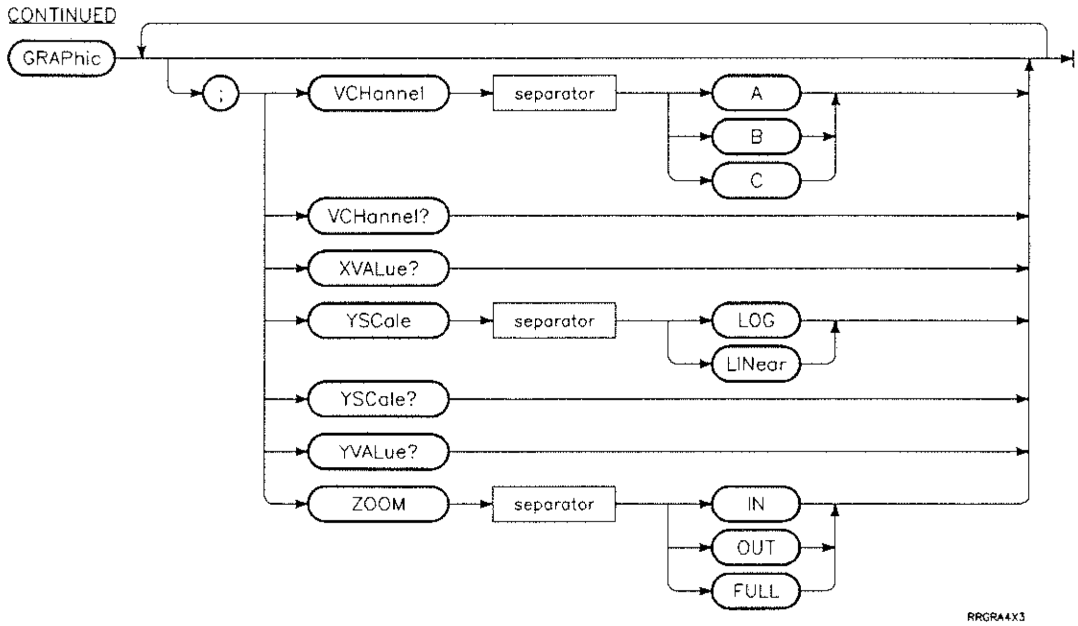


Figure 5-4. GRAPhic Subsystem Syntax Diagram (4 of 4)

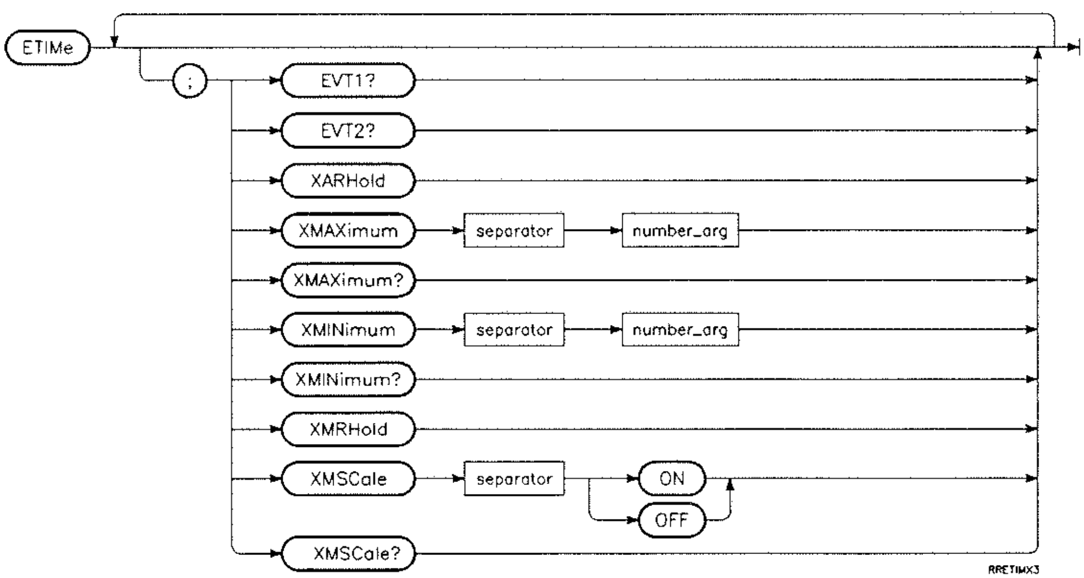


Figure 5-5. GRAPhic Subsystem, ETIMe Sublevel Syntax Diagram

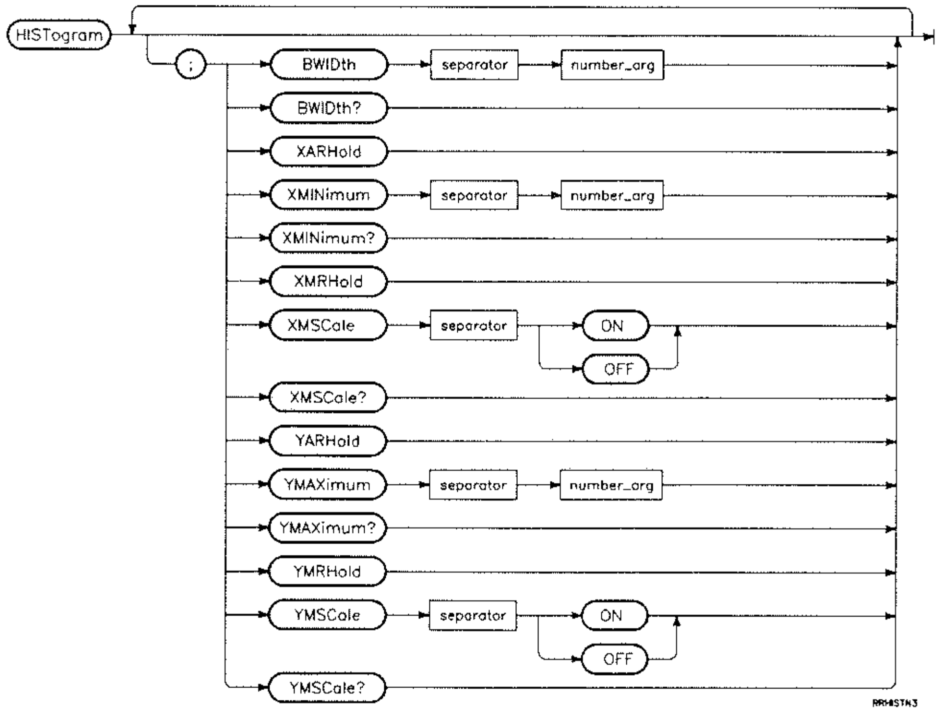


Figure 5-6. GRAPhic Subsystem, HISTogram Sublevel Syntax Diagram

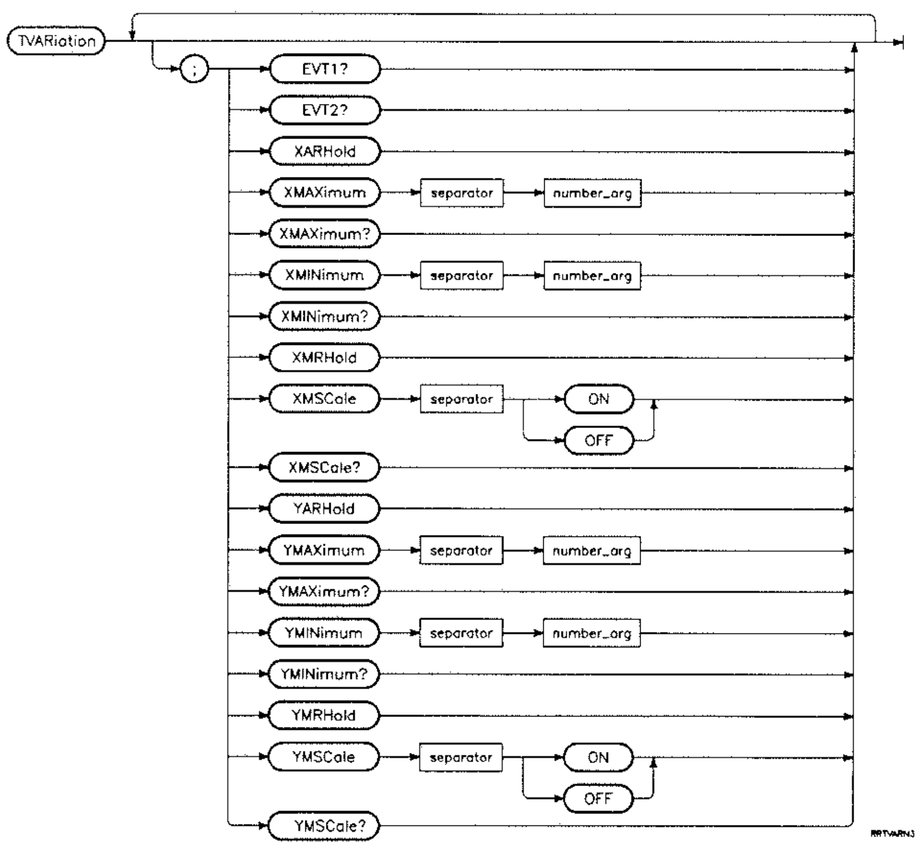


Figure 5-7. GRAPhic Subsystem, TVARiation Sublevel Syntax Diagram

CDAT
Connect data on/off (command/query)

Shortform: CDAT [(turn) Connect DATa (on or off)]
Longform: CDATA

The CDATa command enables or disables data connection on the Time Variation graph. When Connect Data is on, successive measurement data points within a block are connected.

The CDATa? query returns the currently selected Connect Data Mode.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"GRAP;CDAT,ON" - Turns the Connect Data feature on.

OUTPUT 703;"GRAP;CDAT?" - Queries for current Connect Data status.

COPY
Copy Graph to Memory (command)

Shortform: [COPY (main graph to memory)]
Longform: COPY

The COPY command copies the currently displayed Main graph to memory. Any applied zooming and scrolling attributes are stored. Later, you can re-display a stored graph using the SGRaph command. You can re-display a single stored graph or if you operate in dual-display mode you can choose to display both main and memory graphs.

Example: OUTPUT 703;"GRAP;COPY" Copies the current Main display graph to memory.

ETIMe
Event Time
(command)

Shortform: ETIM [Event TIMe]
Longform: ETIME

The ETIMe command selects the Event Time graph sublevel. Additional sublevel commands control Event Time graph features.

An Event Time graph plots measurement start and stop occurrences versus time. This graph is only available for Time Interval measurements.

Example: OUTPUT 703;"GRAP;ETIM;XMSC,ON" - Sets Event Time as graph sublevel and sets the X-axis Manual Scale mode on.

EVT1?
Event Count on
First Channel
(query only)

Shortform: EVT1? [EVenT 1?]
Longform: EVT1?

The EVT1? query returns the current event count on the first channel. The first channel is defined as the first channel in the source equation (for instance A is the first channel in the source equation A→B).

Example: OUTPUT 703;"GRAP;ETIM;EVT1?" - Queries for the current event count of the first channel.

EVT2?
Event Count on
Second Channel
(query only)

Shortform: EVT2? [EVenT 2?]
Longform: EVT2?

The EVT2? query returns the current event count on the second channel. The second channel is defined as the second channel in the source equation (for instance B is the second channel in the source equation A→B).

Example: OUTPUT 703;"GRAP;ETIM;EVT2?" - Queries for the current event count of the second channel.

XARH
X-axis Auto Range Hold
(command)

Shortform: XARH [X-axis Auto Range Hold]
Longform: XARHOLD

Use the XARHold command to copy the current X-axis values to the Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;ETIM;XARH" - Copies current X-axis values to the Manual Scaling parameters.

XMAX
Set X-axis Maximum
(command/query)

Shortform: XMAX [X-axis MAXimum (value)]
Longform: XMAXIMUM

Use the XMAXimum command to set the X-axis maximum value.

The XMAXimum? query returns the X-axis maximum value.

Range: 0 to 1E+8 seconds

Examples: OUTPUT 703;"GRAP;ETIM;XMAX,2.0" - Sets the X-axis maximum value to 2.0 seconds.

OUTPUT 703;"GRAP;ETIM;XMAX?" - Queries for the current X-axis maximum value.

XMIN
Set X-axis Minimum
(command/query)

Shortform: XMIN [X-axis MINimum (value)]
Longform: XMINIMUM

Use the XMINimum command to set the X-axis minimum value.

The XMINimum? query returns the X-axis minimum value.

Range: 0 to 1E+8 seconds

Examples: OUTPUT 703;"GRAP;ETIM;XMIN,0" - Sets the minimum X-axis value to 0.0 seconds.

OUTPUT 703;"GRAP;ETIM;XMIN?" - Queries for the current X-axis minimum value.

XMRH
X-axis Marker Range Hold
(command)

Shortform: XMRH [X-axis Marker Range Hold]
Longform: XMRHOLD

Use the XMRHold command to copy marker values to Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;ETIM;XMRH" - Copy marker values to Manual Scaling parameters.

XMSC
X-axis Manual Scale
(command/query)

Shortform: XMSC [X-axis Manual Scaling]
Longform: XMSCALE

Use the XMSCale command to set the X-axis Manual Scaling on or off.

The XMSCale? query returns the current Manual Scaling status.

Parameter: {ON | OFF}

Examples: OUTPUT 703;"GRAP;ETIM;XMSC,ON" - Sets X-axis Manual Scaling on.

OUTPUT 703;"GRAP;ETIM;XMSC?" - Queries for the current Manual Scaling status.

GDIS
Graphic Display
(command/query)

Shortform: GDIS [Graphic DISplay]
Longform: GDISPLAY

Use the GDISplay command to choose the displayed graph.

Parameters: {HISTogram | TVARiation | ETIMe }

Examples: OUTPUT 703;"GRAP;HIST" - Selects Histogram as the displayed graph.

OUTPUT 703;"GRAP;GDIS?" - Queries for the currently displayed graph.

GRID
Grid
(command/query)

Shortform: GRID
Longform: GRID

Use the GRID command to turn the Grid display on or off. The GRID? query returns the currently selected Grid display mode.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"GRAP;GRID,ON" - Turn the Grid display on.

OUTPUT 703;"GRAP;GRID?" - Queries for the current status of grid display.

HCON
Histogram Continue
(command)

Shortform: HCON [HCONtinue (a paused display)]
Longform: HCONTINUE

This command is one part of a pair of commands that control the Pause/Continue Hardware Histogram feature. After sending a PAUSe command, issue an HCONtinue command to continue the acquisition. (Refer to HPAuse command.)

Example: OUTPUT 703;"GRAP;HCON" - Restart a paused Hardware Histogram acquisition.

HIST
Histogram
(command)

Shortform: HIST [HISTogram]
Longform: HISTOGRAM

The HISTogram command selects the Histogram sublevel. Additional sublevel commands control Histogram graph features.

Example: OUTPUT 703;"GRAP;HIST;YMSC,ON" - Turns Y-axis Manual Scaling on.

BWID
Set Bin Width
(command/query)

Shortform: BWID [(set) Bin WIDth]
Longform: BWIDTH

Use the BWIDth command to set the desired Bin Width for Histogram graphs.

Range: 2E-10 to 1E+24

Examples: OUTPUT 703;"GRAP;HIST;BWID,2E-10" - Sets Bin Width to 200 pS.

OUTPUT 703;"GRAP;HIST;BWID?" - Queries for the current Bin Width value.

XARH
X-axis Auto Range Hold
(command)

Shortform: XARH [X-axis Auto Range Hold]
Longform: XARHOLD

Use XARHold to copy the current X-axis values to manual scaling parameters. This copies the X-axis minimum and maximum values and uses them to set up XMINimum and BWIDth parameters.

Example: OUTPUT 703;"GRAP;HIST;XARH - Copies current X-axis values to the Manual Scaling parameters.

XMIN
Set X-Axis Minimum
(command/query)

Shortform: XMIN? [X-axis MINimum (value)]
Longform: XMINIMUM

Use the XMINimum command to set the Histogram X-axis minimum value.

The XMINimum? query returns the current X-axis minimum value.

Positive Range: 1E-12 to 1E+24 and 0

Negative Range: –1E24 to –1E-12,

Examples: OUTPUT 703;"GRAP;HIST;XMIN,1E+6" - Sets the X-axis minimum value to 1E+6.

OUTPUT 703;"GRAP;HIST;XMIN?" - Queries for the current X-axis minimum value.

XMRH
X-axis Marker Range Hold
(command)

Shortform: XMRH [X-axis Marker Range Hold]
Longform: XMRHOLD

The XMRHold command presets XMINimum and XMAXimum to the vertical marker values.

Example: OUTPUT 703;"GRAP;HIST;XMRH" - Copies marker values to Manual Scaling parameters.

XMSC
X-axis Manual Scale
(command/query)

Shortform: XMSC [X-axis Manual SCale]
Longform: XMSCALE

Use the XMSCale command to turn X-axis Manual Scaling on or off.

The XMSCale? query requests the current X-axis Manual Scaling status.

Examples: OUTPUT 703;"GRAP;HIST;XMSC,ON" - Sets Manual Scaling on.

OUTPUT 703;"GRAP;HIST;XMSC?" - Queries for the current X-axis Manual Scaling status.

YARH
Y-axis Auto Range Hold
(command)

Shortform: YARH [Y-axis Auto Range Hold]
Longform: YARHOLD

Use YARHold to copy the current Y-axis values to the Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;HIST;YARH" - Copy Y-axis values to Manual Scaling parameters.

YMAX
Y-axis Maximum Value
(command/query)

Shortform: YMAX [Y-axis MAXimum (value)]
Longform: YMAXIMUM

Use the YMAXimum command to set the Histogram Y-axis maximum value.

The YMAXimum? query returns the current Y-axis maximum value.

Range: 5 to 1E+12

Examples: OUTPUT 703;"GRAP;HIST;YMAX,1000" - Sets Y-axis maximum value to 1000.

OUTPUT 703;"GRAP;HIST;YMAX?" - Queries for the current Y-axis maximum value.

YMRH
Y-axis Marker Range Hold
(command)

Shortform: YMRH [Y-axis Marker Range Hold]
Longform: YMRHOLD

Use the YMRHold command to copy marker values to Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;HIST;YMRH" - Copy markers.

YMSC
Y-axis Manual Scaling
(command/query)

Shortform: YMSC [Y-axis Manual SCaling]
Longform: YMSCALE

Use YMSCale to turn the Y-axis Manual Scaling on or off.

The YMSCale? query returns the current Y-axis Manual Scaling status.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"GRAP;HIST;YMSC,ON" Sets Y-axis Manual Scaling on.

OUTPUT 703;"GRAP;HIST;YMSC?" - Queries for the current Y-axis Manual Scaling status.

HMAX? Histogram Maximum Value (query only)

Shortform: HMAX? [Histogram MAXimum (value)]
Longform: HMAXIMUM?

Use HMAXimum? to query the current setting of the Histogram maximum value.

Example: OUTPUT 703;"GRAP;HMAX?" - Queries for the current Histogram maximum value.

HME? Histogram Mean Value (query only)

Shortform: HME? [Histogram MEan (value)]
Longform: HMEAN?

Use HMEAn? to query the statistical mean of a Histogram.

Note: Correct Histogram Statistics are only available when the HP 5373A is in the STATS mode.

Example: OUTPUT 703;"GRAP;HME?" - Queries for the current Histogram Minimum value.

HMIN? Histogram Minimum Value (query only)

Shortform: HMIN? [Histogram MINimum (value)]
Longform: HMINIMUM?

Use HMAXimum? to query the current Histogram minimum value.

Example: OUTPUT 703;"GRAP;HMIN?" - Queries for the current Histogram Mean value.

HPA Histogram Pause (command)

Shortform: HPA [Histogram PAuse]
Longform: HPAUSE

This command controls the pause portion of the Pause/Continue Histogram feature. When received, the Histogram stops before data acquisition is complete. Issue the

HCONTinue command to continue the Histogram. (Refer to HCONTinue command.)

Example: OUTPUT 703;"GRAP;HPA" - Pauses the Hardware Histogram acquisition in progress.

HSD? Histogram Standard Deviation Value (query only)

Shortform: HSD? [Histogram Standard Deviation]
Longform: HSDEV

Use HSDeviation? to query the Histogram standard deviation value.

Example: OUTPUT 703;"GRAP;HSD?" - Queries for the current Histogram Standard Deviation value.

MCEN? Modulation Center Value (query only)

Shortform: MCEN? [Modulation CENter (value)]
Longform: MCENTER?

The MCENter? queries asks for the modulation center value. The result is obtained by analyzing the data between the vertical display markers for maximum Y-axis peaks. The center value is the midpoint between the peaks. It is calculated as one of the results when MDMode equals MODulation.

Example: OUTPUT 703;"GRAP;MCEN?" - Queries for the Modulation Center value.

MDM Marker Display Mode (command/query)

Shortform: MDM [Marker Display Mode]
Longform: MDMODE

The MDMode command sets the Marker Display Mode to MARKer, DELTa, STATistics or MODulation.

MARKer – Marker coordinates (X and Y)

DELTa – marker values (Delta X and Y between the two markers)

STATistics – (between the two markers)

MODulation – parameters (between the two markers)

Parameters:	{MARKer | DELTa | STATistics | MODulation}
Example:	OUTPUT 703;"GRAP;MDM,MARK" - Sets Marker Display Mode to Marker.
	OUTPUT 703;"GRAP;MDM?" - Queries for current Marker Display Mode.

MDOW
Move Marker Down
(command)

Shortform: MDOW [(move) Marker DOWn]
Longform: MDOWN

Use MDOWn to move the graphics display marker down. This simulates using the front panel knob. The complementary command is MUP.

Range: 1 to 80

Example: OUTPUT 703;"GRAP;MDOW,40" - Moves marker down 40 pixels.

MEM
Memory Graph
(command)

Shortform: MEM [MEMory (graph)]
Longform: MEMORY

The MEMory command selects the Memory Graph sublevel.

Example: OUTPUT 703;"GRAP;MEM" - Selects the Memory Graph sublevel.

XMAX?
X-Axis Maximum
(query only)

Shortform: XMAX? [X-axis Maximum (value)]
Longform: XMAXIMUM?

Use XMAXimum? to query the X-axis maximum value for the Memory Graph.

Example: OUTPUT 703;""GRAP;MEM;XMAX?" - Queries for current X-axis maximum value.

XMIN?
X-Axis Minimum
(query only)

Shortform: XMIN? [X-axis MINimum (value)]
Longform: XMINIMUM?

Use XMINimum? to query the X-axis minimum value for the Memory Graph.

Example: OUTPUT 703;"GRAP;MEM;XMIN?" - Queries for current X-axis minimum value.

YMAX?
Y-Axis Maximum?
(query only)

Shortform: YMAX? [Y-axis MAXimum (value)]
Longform: YMAXIMUM?

Use YMAXimum? to query the Y-axis maximum value for the Memory Graph.

Example: OUTPUT 703;"GRAP;MEM;YMAX?" - Queries for current Y-axis maximum value.

YMIN?
Y-Axis Minimum?
(query only)

Shortform: YMIN? [Y-axis MINimum (value)]
Longform: YMINIMUM?

Use YMINimum? to query the Y-axis minimum value for the Memory Graph.

Example: OUTPUT 703;"GRAP;MEM;YMIN?" - Queries for current Y-axis minimum value

---

MGR
Main Graph
(command)

Shortform: MGR [Main GRaph]
Longform: MGRAPH

The MGRaph command selects the Main graph sublevel.

Example: OUTPUT 703;"GRAP;MGR" - Selects the Main Graph sublevel.

XMAX?
X-Axis Maximum?
(query only)

Shortform: XMAX? [X-axis MAXimum (value)]
Longform: XMAXIMUM?

Use XMAXimum? to query the X-axis maximum value for Main Graph.

Examples: OUTPUT 703;"GRAP;MGR;XMAX?" - Queries for current X-axis maximum value.

XMIN?
X-Axis Minimum?
(query only)

Shortform: XMIN? [X-axis MINimum (value)]
Longform: XMINIMUM?

Use XMINimum? to query the X-axis minimum value for the Main Graph.

Example: OUTPUT 703;"GRAP;MGR;XMIN?" - Queries for current X-axis minimum value.

YMAX?
Y-Axis Maximum?
(query only)

Shortform: YMAX? [Y-axis MAXimum (value)]
Longform: YMAXIMUM?

Use YMAXimum? to query the Y-axis maximum value for the Main Graph.

Example: OUTPUT 703;"GRAP;MGR;YMAX?" - Queries for current Y-axis maximum value.

YMIN?
Y-Axis Minimum?
(query only)

Shortform: YMIN? [Y-axis MINimum (value)]
Longform: YMINIMUM?

Use YMINimum? to query the Y-axis minimum value for the Main Graph.

Example: OUTPUT 703;"GRAP;MGR;YMIN?" - Queries for current Y-axis minimum value

---

MLEF
Move Marker Left
(command)

Shortform: MLEF [(move) Marker LEFt]
Longform: MLEFT

Use MLEFt to move the graphics display marker left. This simulates using the front panel knob. The complementary command is MRIGht.

Range: 1 to 124

Example: OUTPUT 703;"GRAP;MLEF,80" - Moves the marker left 80 display points. For Histograms, this is 80 bins, for Time Variation or Event Time this is 80 display columns.

NOTE

If MNEXT,PIXEL is set before the MLEF command is issued, the MLEF command moves the marker N pixels to the left, where N is the parameter sent with the MLEF command.

IF MNEXT,MEAS is set before the MLEF command is issued, the MLEF command moves the marker one data point to the left, and the parameter sent with MLEF is ignored.

MMAX Move Marker to Maximum Value (command)

Shortform: MMAX [(move) Marker (to) MAXimum]
Longform: MMAXIMUM

Use the MMAXimum command to move the marker to the maximum value displayed. For a Histogram, this is the maximum bin height displayed; for a Time Variation graph this is the maximum Y-value, which is the maximum measured value

Example: OUTPUT 703;"GRAP;MMAX" - Move marker to maximum value.

MMIN Move Marker to Minimum Value (command)

Shortform: MMIN [(move) Marker (to) MINimum]
Longform: MMINIMUM

Use the MMINimum command to move the marker to the minimum value displayed. For a Histogram, this is the minimum bin height displayed; for a Time Variation graph this is the minimum Y-value, which is the minimum measured value.

Example: OUTPUT 703;"GRAP;MMIN" - Move marker to minimum value.

MMOV Copy Inactive Marker to Active (command)

Shortform: MMOV [Marker MOVe (to position of active marker)]
Longform: MMOVE

Use the MMOVe command to move the inactive marker to the active marker position.

Example: OUTPUT 703;"GRAP;MMOV" - Move inactive marker to active marker position.

MNEX Set Marker Next Mode (command/query)

Shortform: MNEX [Marker NEXt]
Longform: MNEXT

Use the MNEXt command to set the marker control mode to move-marker-to-next where next means either a pixel or a data point. On a Time Variation graph, if data points are far apart and you always want the marker on a data point, use (next) MEASurement. Conversely, to get the marker in a place between data points, use (next) PIXel.

The MNEXt? query returns the Marker Next status.

Parameters: {MEASurement | PIXel}

Examples: OUTPUT 703;"GRAP;MNEX,PIX" - Sets marker to move to next pixel.

OUTPUT 703;"GRAP;MNEX?" - Queries for the current Marker Next mode.

MNUM? Get Measurement Number (query only)

Shortform: MNUM? [Measurement NUMber]
Longform: MNUMBER?

The MNUMber? query requests the measurement number associated with the active vertical marker. This command is only relevant for Time Variation or Event Time graphs.

Example: OUTPUT 703;"GRAP;MNUM?" - Queries for the current measurement number associated with the active vertical marker.

MOR
Marker Orientation
(command/query)

Shortform: MOR [Marker ORientation]
Longform: MORIENT?

The MORient command is used to select the vertical or horizontal marker orientation.

The MORient? query returns the current marker orientation.

Parameters: {VERTical | HORizontal}

Examples: OUTPUT 703;"GRAP;MOR,VERT" - Sets Marker Orientation to vertical.

OUTPUT 703;"GRAP;MOR?" - Queries for the current Marker Orientation.

MRAT?
Get Modulation Rate
(query only)

Shortform: MRAT? [(get) Modulation RATe]
Longform: MRATE?

The MRATe? query returns the Modulation Rate value. This result is obtained by analyzing the data between the vertical display markers to arrive at an estimate of the modulation rate. Partial periods between the markers do not detract from the accuracy of this estimate. Modulation parameters are calculated when the Marker Display Mode has been set to Modulation.

If the HP 5373A cannot compute the modulation rate, the value –1 is returned.

Example: OUTPUT 703;"GRAP;MRAT?" - Queries for the Modulation Rate.

MRIG
Move Marker Right
(command)

Shortform: MRIG [(move) Marker RIGht]
Longform: MRIGHT

Use MRIGht to move the graphics display marker right. This simulates using the front panel knob. The complementary command is MLEFt.

Range: 1 to 124

Example: OUTPUT 703;"GRAP;MRIG,40" - Moves the marker right 40 display points.

NOTE

If MNEXT,PIXEL is set before the MRIG command is issued, the MRIG command moves the marker N pixels to the right, where N is the parameter sent with the MRIG command.

IF MNEXT,MEAS is set before the MRIG command is issued, the MRIG command moves the marker one data point to the right, and the parameter sent with MRIG is ignored.

MUP
Move Marker Up
(command)

Shortform: MUP [(move) Marker UP]
Longform: MUP

Use MUP to move the graphics display marker up. This simulates using the front panel knob. The complementary command is MDOWn.

Range: 1 to 90

Examples: OUTPUT 703;"GRAP;MUP,40" - Moves marker up 40 pixels.

OUTL
Outline Mode
(command/query)

Shortform: OUTL [OUTLine (mode)]
Longform: OUTLINE

This command applies to Histograms. It turns outline mode on or off. When OUTLine is on, the instrument displays only the silhouette of a Histogram (Meaning only the tops of the Histogram bins are drawn, connected vertically as required.) Conversely, setting OUTLine off draws lines from the top of each respective bin to the Histogram base (stated another way, the right and left sides of the bin extend from top to bottom). The query form asks for the current setting.

The panorama graph is always drawn in OUTLine on mode.

The OUTLine? query returns the Outline Mode status.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"GRAP;OUTL,ON" - Turns Outline Mode on.

OUTPUT 703;"GRAP;OUTL?" - Queries for current Outline Mode.

PDEV? Peak Deviation (query only)

Shortform: PDEV? [Peak DEViation]
Longform: PDEViation?

Use the PDEViation? query to obtain the peak deviation value. This result is obtained by analyzing the data between the vertical display markers to arrive at an estimate of the maximum Y-axis peaks. The returned value is the (maximum positive peak) minus (maximum negative peak).

Modulation parameters are calculated when the Marker Display Mode has been set to Modulation.

Example: OUTPUT 703;"GRAP;PDEV?" - Queries for current Peak Deviation value.

SEL Select Graph Menu Level (command/query)

Shortform: SEL [SELect (graph menu)]
Longform: SELECT

Use the SELect command to select the graph menu level. Five options are available.

- MAIN
- MARKer
- ZOOM
- SCALe
- DISPlay

The SELect? query requests current the menu level displayed.

Parameters: {MAIN | MARKer | ZOOM | SCALe | DISPlay}

Examples: OUTPUT 703;"GRAP;SEL,ZOOM" - Selects ZOOM level softkeys.

OUTPUT 703;"GRAP;SEL?" - Queries for current Graph menu softkey level.

SGR Show Graph (command/query)

Shortform: SGR [Show GRaph]
Longform: SGRAPH

The SGRaph command selects from three types of displays:

- Main graph only
- Memory graph only
- Main and Memory simultaneously.

The SGRaph? query returns the current graph display selection.

Parameters: {MAIN | MEM | BOTH}

Examples: OUTPUT 703;"GRAP;SGR,MAIN" - Selects display of Main graph.

OUTPUT 703;"GRAP;SGR?" - Queries for current graph display selection.

SMAR Select Active Marker (command/query)

Shortform: SMAR [Select (active) MARker]
Longform: SMARKER

The SMARker command selects the active marker. There are two choices.

- BLACk
- WHITe

The SMAR? query returns the current active marker color.

Parameters: {BLACk | WHITe}

Examples: OUTPUT 703;"GRAP;SMAR<BLAC" - Sets active marker to Black.

OUTPUT 703;"GRAP;SMAR?" - Queries for current active marker color.

SSCR Screen Scroll (command)

Shortform: SSCR [Screen SCRoll]
Longform: SSCROLL

The SSCRoll command scrolls the screen one "page's" worth — a page is defined by the portion of the total graph currently displayed. The command contrõls the scrolling of a screen from left to right or vice versa.

Parameters: {LEFT | RIGHt}

Example: OUTPUT 703;"GRAP;SSCR,LEFT" - Scrolls screen to the left.

TVAR (command)

Shortform: TVAR [Time VARiation]
Longform: TVARIATION

The TVARiation command selects the Time Variation graph sublevel. Additional sublevel commands control Time Variation graph features.

Example: OUTPUT 703;"GRAP;TVAR;XARH" - Copies the current X-axis values to the manual scaling parameters.

EVT1? Event Count on First Channel (query only)

Shortform: EVT1? [EVenT 1?]
Longform: EVT1?

The EVT1? query returns the current event count on the first channel. The first channel is defined as the first channel in the source equation (for instance A is the first channel in the source equation A/B).

Example: OUTPUT 703;"GRAP;TVAR;EVT1?" - Queries for the current event count on the first channel.

EVT2? Event Count on Second Channel (query only)

Shortform: EVT2? [EVenT 2?]
Longform: EVT2?

The EVT2? query returns the current event count on the second channel. The second channel is defined as the second channel in the source equation (for instance, B is the second channel in the source equation A/B).

Example: OUTPUT 703;"GRAP;TVAR;EVT2?" - Queries for the current event count on the second channel.

XARH
X-axis Auto Range Hold
(command)

Shortform: XARH [X-axis Auto Range Hold]
Longform: XARHOLD

Use the XARHold command to copy the current X-axis values to the Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;TVAR;XARH" - Copies current X-axis values to the Manual Scaling parameters.

XMAX
Set X-axis Maximum
(command/query)

Shortform: XMAX [X-axis MAXimum (value)]
Longform: XMAXIMUM

Use the XMAXimum command to set the X-axis maximum value.

The XMAXimum? query returns the current X-axis maximum value.

Range: 0 to 1E+8 seconds

Examples: OUTPUT 703;"GRAP;TVAR;XMAX,2.0" - Sets maximum X-axis value to 2.0 seconds.

OUTPUT 703;"GRAP;TVAR;XMAX? - Queries for the current X-axis maximum value.

NOTE

The XMAX? query outputs different formats depending on whether the graph addressed is main graph (MGR) or memory graph (MEM). The main graph query returns data in numeric format, while the memory graph query returns data in string format. For example:

If the XMAX for MGR and MEM are both 746.4782 μs

GRAP;MGR;XMAX? returns the numeric value 746.4782E-06
GRAP;MEM;XMAX? returns the string value 746.4782 μs.

XMIN
Set X-axis Minimum
(command/query)

Shortform: XMIN [X-axis MINimum (value)]
Longform: XMINIMUM

Use the XMINimum command to set the X-axis minimum value.

The XMINimum? query returns the current X-axis minimum value.

Range: 0 to 1E+8 seconds

Examples: OUTPUT 703;"GRAP;TVAR;XMIN,0.0" - Sets minimum X-axis value to zero seconds.

OUTPUT 703;"GRAP;TVAR;XMIN?" - Queries for the current setting.

NOTE

The XMIN? query outputs different formats depending on whether the graph addressed is main graph (MGR) or memory graph (MEM). The main graph query returns data in numeric format, while the memory graph query returns data in string format. For example:

If the XMIN for MGR and MEM are both 746.4782 μs

GRAP;MGR;XMIN? returns the numeric value 746.4782E-06
GRAP;MEM;XMIN? returns the string value 746.4782 μs.

XMRH
X-axis Marker Range Hold
(command)

Shortform: XMRH [X-axis MaRker Hold]
Longform: XMRHOLD

The XMRHold command presets XMINimum and XMAXimum to the vertical marker values.

Example: OUTPUT 703;"GRAP;TVAR;XMRH" - Copies marker values to Manual Scaling parameters.

XMSC
X-axis Manual Scale
(command/query)

Shortform: XMSC [X-axis Manual SCale]
Longform: XMSCALE

Use the XMSCale command to turn X-axis Manual Scaling on or off.

The XMSCale? query returns the current X-axis Manual Scaling status.

Examples: OUTPUT 703;"GRAP;TVAR;XMSC,ON" - Sets Manual Scaling on.

OUTPUT 703;"GRAP;XMSC?" - Queries for the current X-axis Manual Scaling status.

YARH
Y-axis Auto Range Hold
(command)

Shortform: YARH [Y-axis Auto Range Hold]
Longform: YARHOLD

Use YARHold to copy the current Y-axis values to the Manual Scaling parameters.

Example: OUTPUT 703;"GRAP;TVAR;YARH" - Copies current Y-axis values to the Manual Scaling parameters.

YMAX
Set Y-axis Maximum
(command/query)

Shortform: YMAX [Y-axis MAXimum (value)]
Longform: YMAXIMUM

Use the YMAXimum command to set the Y-axis maximum value.

Positive Range: 1E–12 ≤n ≤1E+36, and 0
Negative Range: –1E+24 ≤n ≤ –1E–12

Examples: OUTPUT 703;"GRAP;TVAR;YMAX,1E6" - Sets Y-axis maximum value to 1E6.

OUTPUT 703;"GRAP;TVAR;YMAX?" - Queries for the current Y-axis maximum value.

NOTE

The YMAX? query outputs different formats depending on whether the graph addressed is main graph (MGR) or memory graph (MEM). The main graph query returns data in numeric format, while the memory graph query returns data in string format. For example:

If the YMAX for MGR and MEM are both 746.4782 μs

GRAP;MGR;YMAX? returns the numeric value 746.4782E-06
GRAP;MEM;YMAX? returns the string value 746.4782 μs.

YMIN
Set Y-axis Minimum
(command/query)

Shortform: YMIN [Y-axis MINimum (value)]
Longform: YMINIMUM

Use the YMINimum command to set the Y-axis minimum value.

Positive Range: $1E{-}12 \le n \le 1E{+}24$, and 0
Negative Range: $-1E{+}24 \le n \le -1E{-}12$

Examples: OUTPUT 703;"GRAP;TVAR;YMIN,1E+6" - Sets Y-axis minimum value to 1E+6.

OUTPUT 703;"GRAP;TVAR;YMIN?" - Queries for the current Y-axis minumum value.

NOTE

The YMIN? query outputs different formats depending on whether the graph addressed is main graph (MGR) or memory graph (MEM). The main graph query returns data in numeric format, while the memory graph query returns data in string format. For example:

If the YMIN for MGR and MEM are both 746.4782 μs

GRAP;MGR;YMIN? returns the numeric value 746. 4782E-06
GRAP;MEM;YMIN? returns the string value 746.4782 μs.

YMRH
Y-axis Marker Range Hold (command)

Shortform: YMRH [Y-axis Marker Range Hold]
Longform: YMRHOLD

The YMRHold command presets YMINimum and YMAXimum to the horizontal marker values.

Example: OUTPUT 703;"GRAP;TVAR;YMRH" - Copies marker values to Manual Scaling parameters.

YMSC
Y-axis Manual Scale (command/query)

Shortform: YMSC [Y-axis Manual SCaling]
Longform: YMSCALE

Use the YMSCale command to turn the Y-axis Manual Scaling on or off.

The YMSCale? query returns the current Y-axis Manual Scaling status.

Parameter: {ON | OFF}

Examples: OUTPUT 703;"GRAP;TVAR;YMSC,ON" - Sets Y-axis Manual Scaling on.

OUTPUT 703;"GRAP;TVAR;YMSC?" - Queries for the current Y-axis Manual Scaling status.

UPD
Set Update Mode (command/query)

Shortform: UPD [UPDate (mode)]
Longform: UPDATE

Use the UPDate command to set the graphic update mode to update the display after each data acquisition pass (WHILe) or after the final pass (AFTer). Accordingly, the command only applies to multiple-pass measurements.

The UPDate? query returns the currently selected Update mode.

Parameters: {WHILe | AFTer}

Examples: OUTPUT 703;"GRAP;UPD,WHIL" - Sets graph updating to occur after each pass.

OUTPUT 703;"GRAP;UPD?" - Queries for the current Update mode.

VCH View Channel (command/query)

Shortform: VCH [View CHannel]
Longform: VCHANNEL

Use the VCHannel command to select the View Channel for dual-channel, dual-result measurements. These measurements are: Frequency, Period, or Totalize, A&B, B&C, or A&C.

The VCHannel? query returns the currently selected View Channel.

Parameters: {A | B | C}

Examples: OUTPUT 703;"GRAP;VCH,A" - Selects channel A as the view channel.

OUTPUT 703;"GRAP;VCH?" - Queries for the current view channel selection.

XVAL? Get X-axis Value (query only)

Shortform: XVAL? [X-axis VALue]
Longform: XVALUE?

TheXVALue? query returns the current X-value for Marker or Delta Marker Display Mode Values.

Example: OUTPUT 703;"GRAP;XVAL?" - Queries for the current Marker or Delta X-values.

YSC Y-axis Scale (command/query)

Shortform: YSC [Y-axis SCale]
Longform: YSCALE

The YSCale command sets the Y-axis scaling mode for Histograms. The scale choices are LOG or LINear.

The YSCale? query returns the current Y-axis scaling mode.

Parameters: {LOG | LINear}

Examples: OUTPUT 703;"GRAP;LOG" - Sets Y-axis scale to logarithmic scale.

OUTPUT 703;"GRAP;YSC?" - Queries for the current Y-axis scaling mode.

YVAL?
Get Y-axis Value
(query only)

Shortform: YVAL? [Y-axis VALue]
Longform: YVALUE?

The YVALue? query returns the current Y-value for Marker or Delta Marker Display Mode values.

Example: OUTPUT 703;"GRAP;YVAL?" - Queries for the current Marker or Delta Y-values.

ZOOM
Zoom a Graph
(command)

Shortform: ZOOM [ZOOM]
Longform: ZOOM

This control zooms (magnifies) any graph. Three views are possible: in, out, and full. ZOOM IN gives you increased resolution, ZOOM OUT gives you more of the "big picture". ZOOM FULL returns the graph to full scale (original acquisition picture).

Parameters: {IN | OUT | FULL}

Examples: OUTPUT 703;"GRAP;ZOOM,FULL" - Sets display for full scale.

Measurement (Function/ Pre-trigger)

MEASUREMENT SUBSYSTEM COMMANDS

The MEASurement subsystem controls the HP 5373A measurement modes, arming modes, measurement size, and measurement holdoff (start) and sampling (stop) conditions. The commands in this subsystem are equivalent to those available via the front panel FUNCTION and PRE-TRIGGER menus. Syntax diagrams for the MEASurement subsystem commands are shown in *Figure 5-8*. Major sublevels SAMPle (*Figure 5-9*) and STARt (*Figure 5-10*) are diagrammed separately.

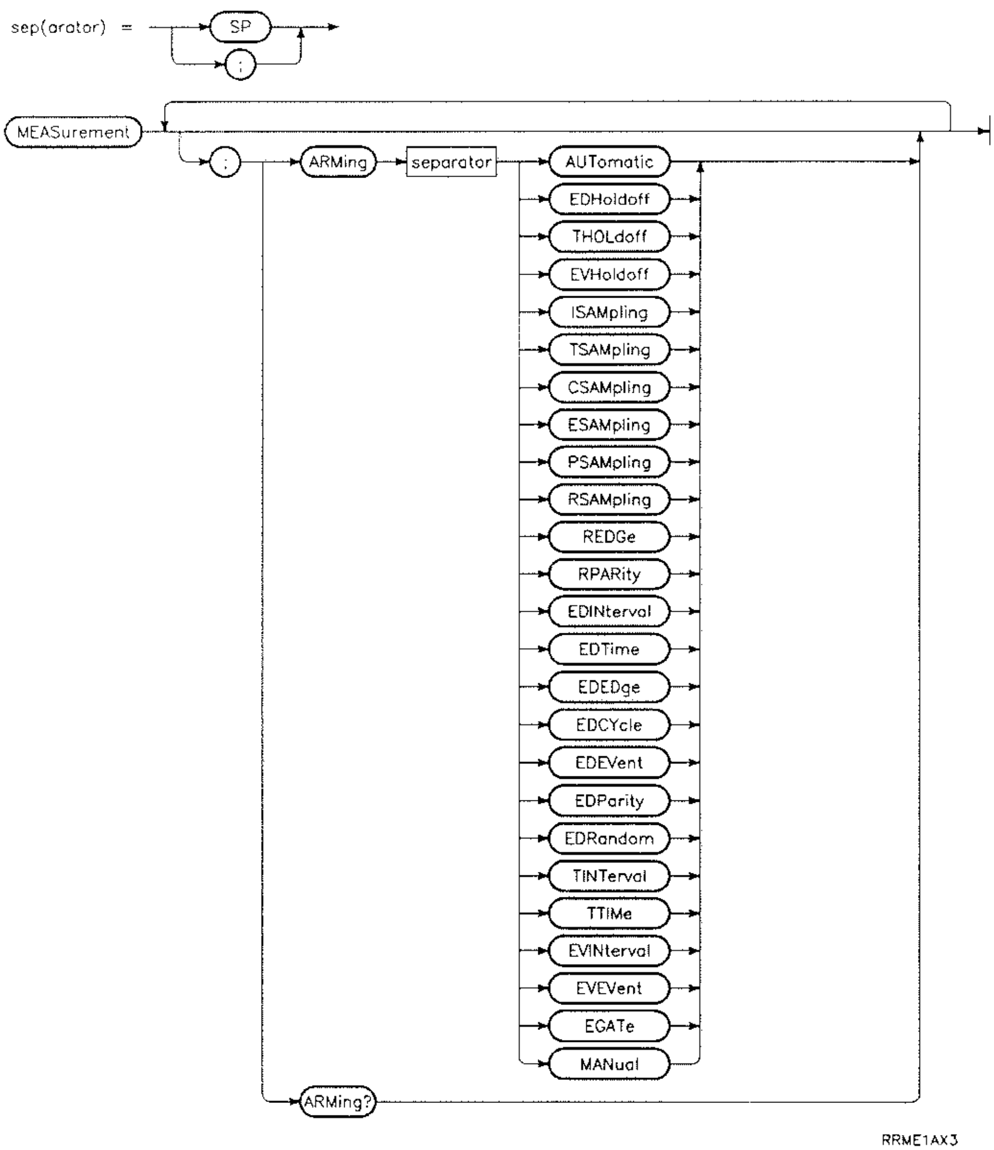


Figure 5-8. MEASurement Subsystem (FUNCTION/PRE-TRIGGER) Syntax Diagram (1 of 4)

CONTINUED

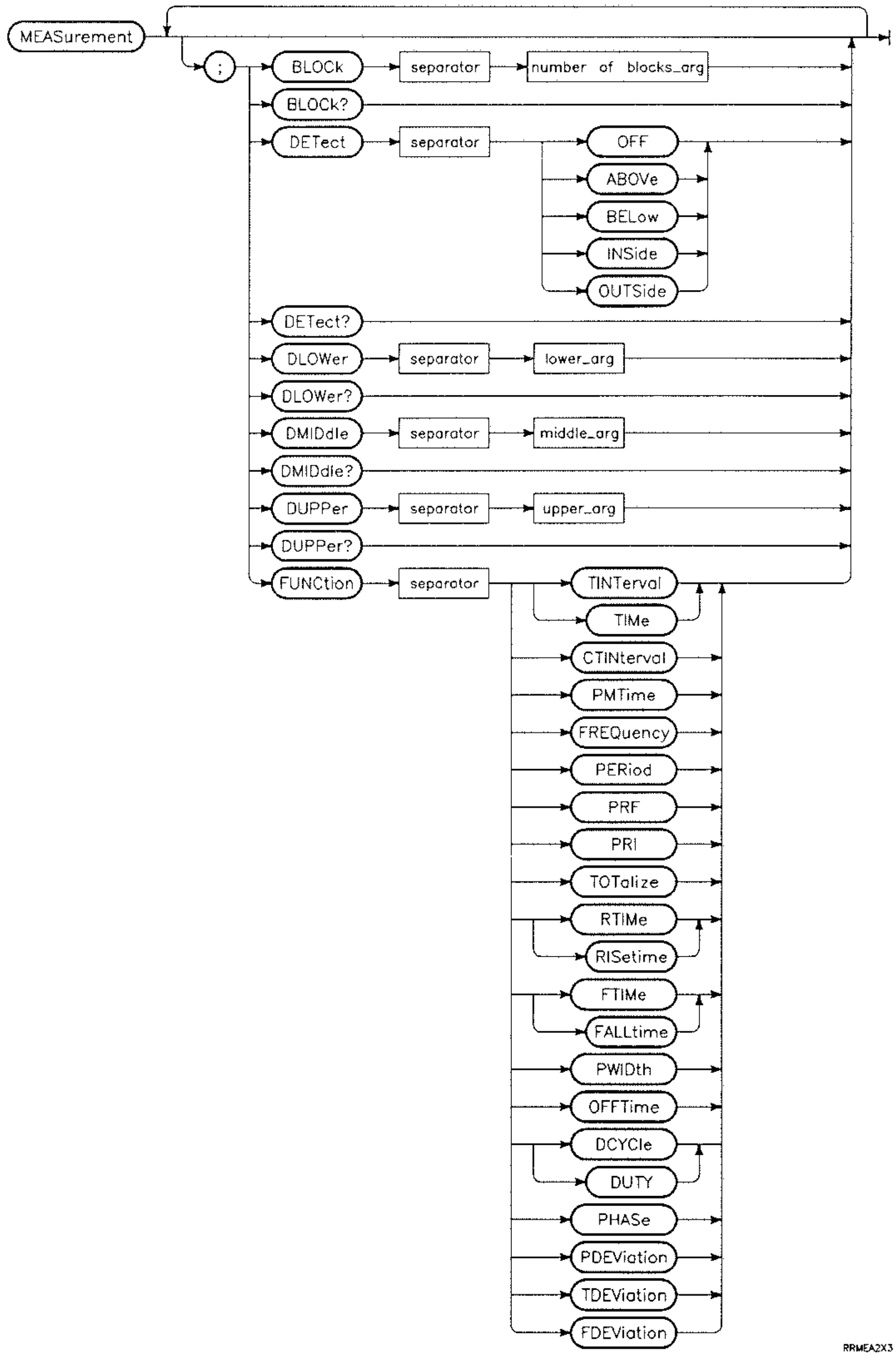


Figure 5-8. MEASurement Subsystem (FUNCTION/PRE-TRIGGER) Syntax Diagram (2 of 4)

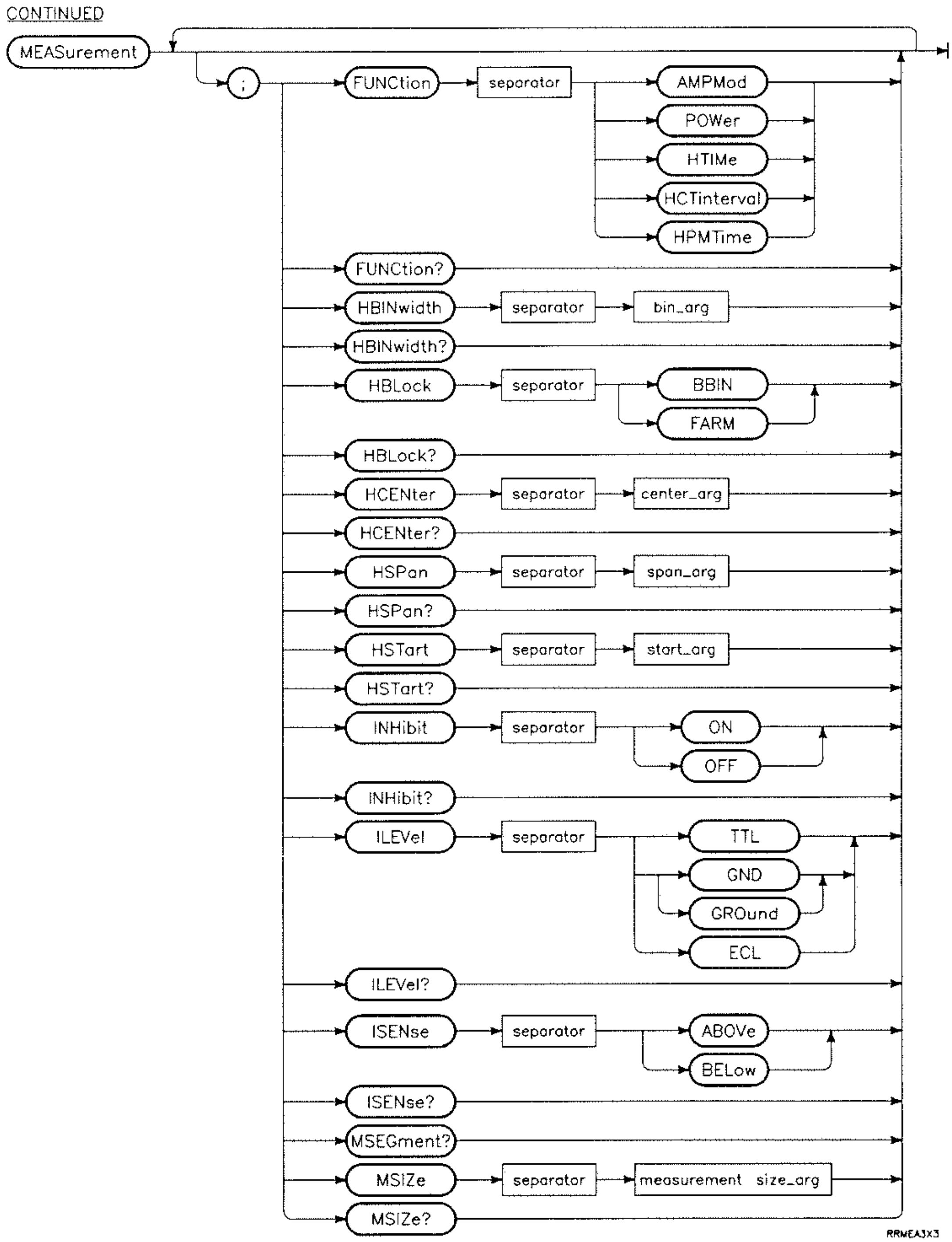
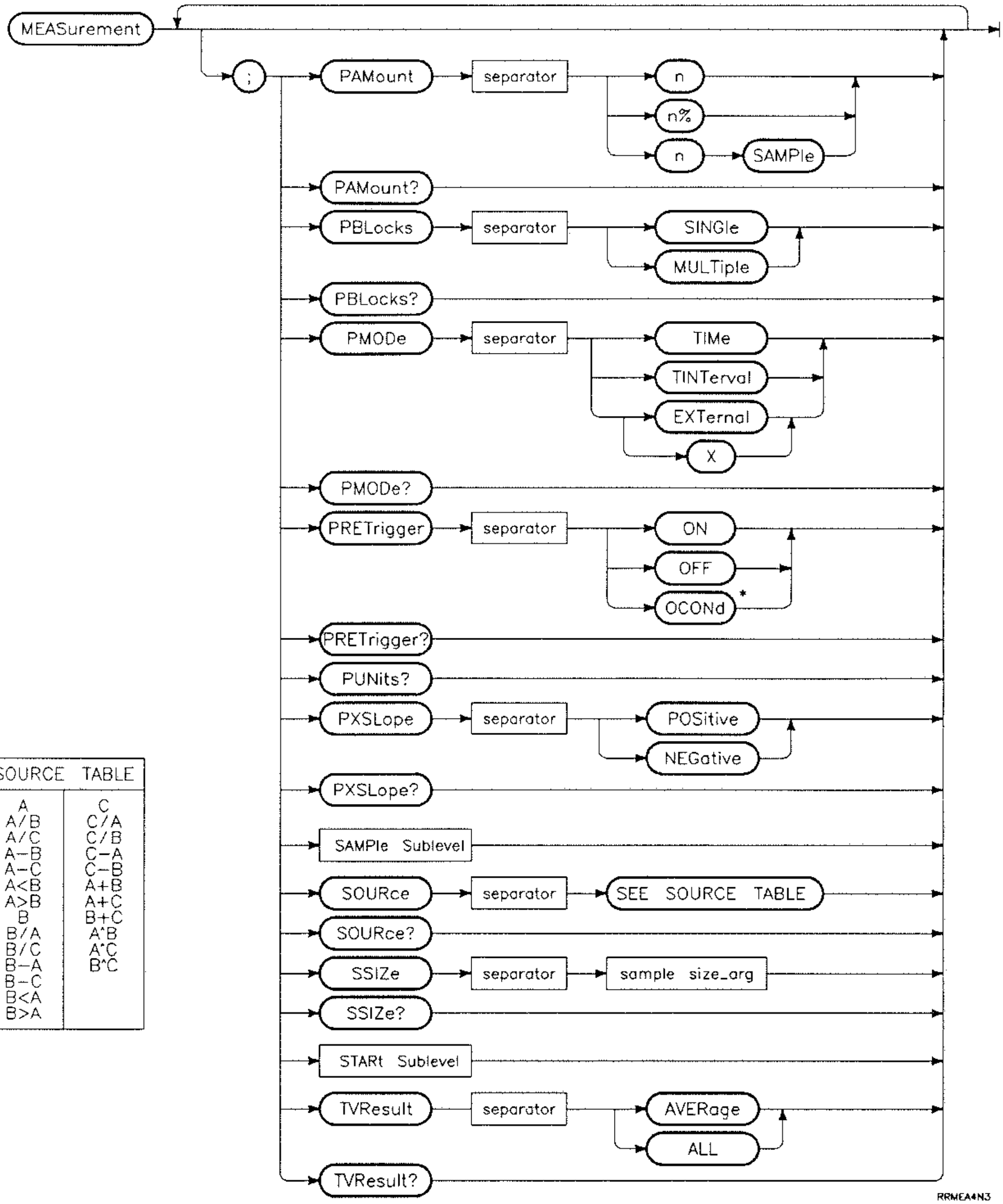


Figure 5-8. MEASurement Subsystem (FUNCTION/PRE-TRIGGER) Syntax Diagram (3 of 4)

CONTUINED



SOURCE	TABLE
A	C
A/B	C/A
A/C	C/B
A−B	C−A
A−C	C−B
A<B	A+B
A>B	A+C
B	B+C
B/A	A*B
B/C	A*C
B−A	B*C
B−C	
B<A	
B>A	

* See PRET in MEASurement command list

Figure 5-8. MEASurement Subsystem (FUNCTION/PRE-TRIGGER) Syntax Diagram (4 of 4)

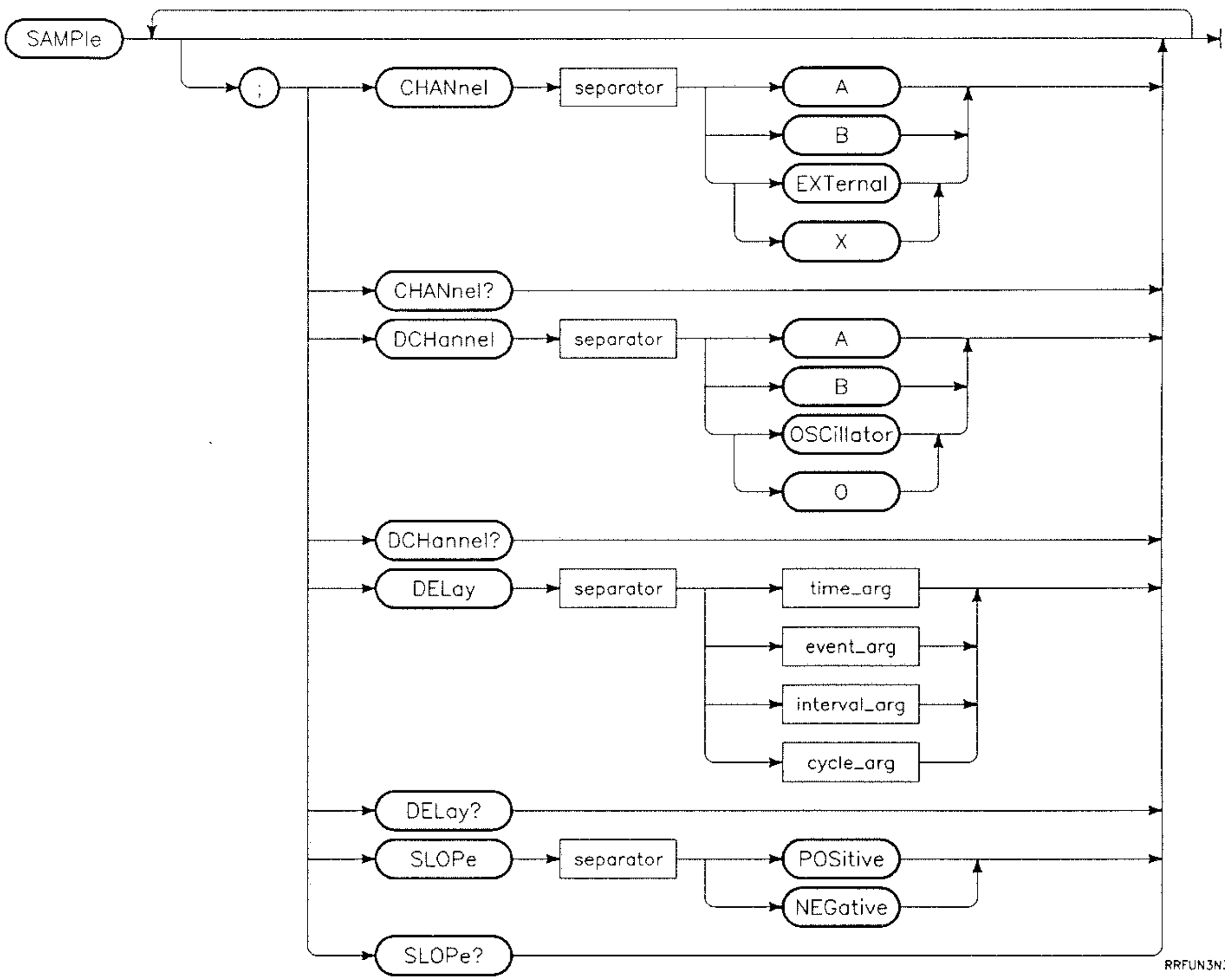


Figure 5-9. MEASurement Subsystem SAMPle Sublevel Syntax Diagram

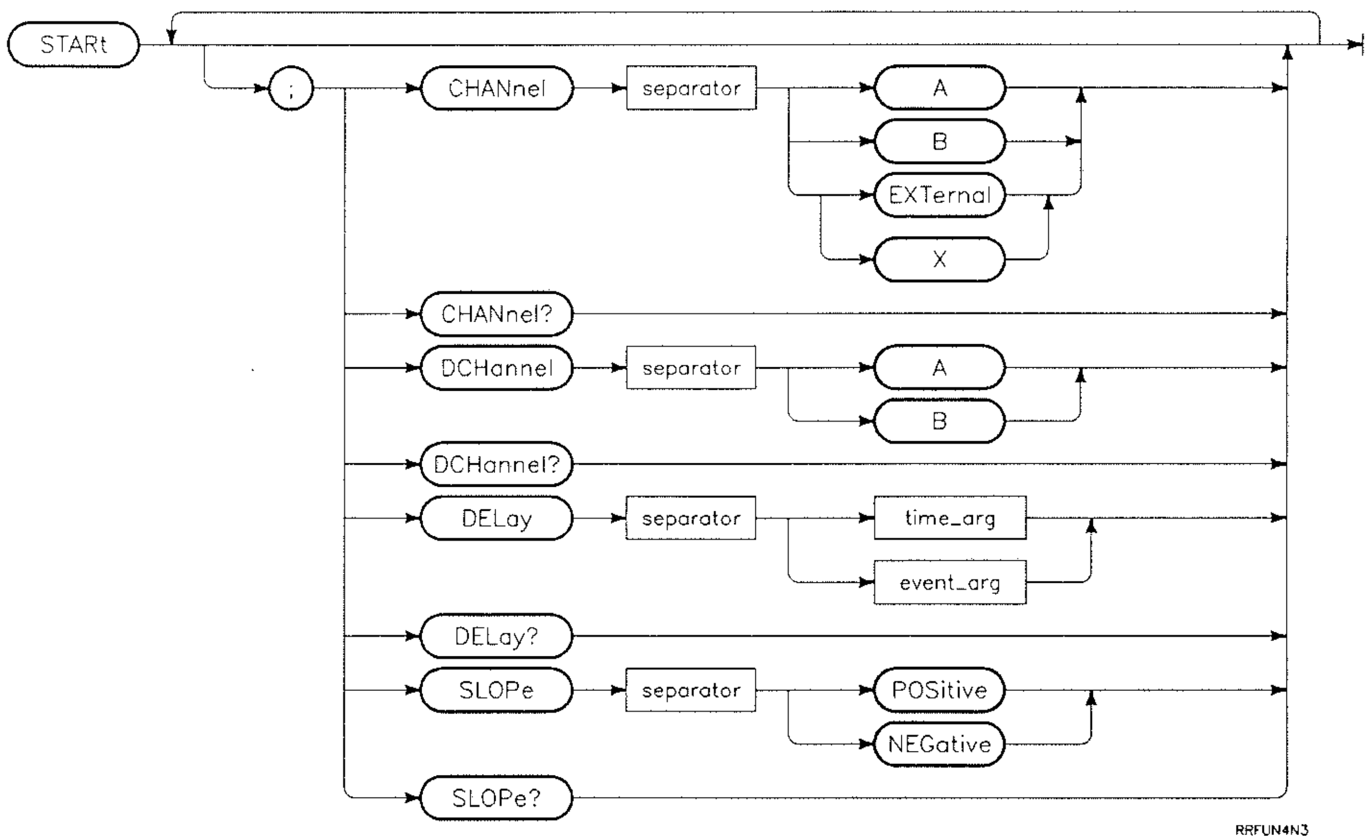


Figure 5-10. MEASurement Subsystem STARt Sublevel Syntax Diagram

ARM Arming (command/query)

Shortform: ARM [ARMing]
Longform: ARMING

The ARMing command selects the arming for measurement of input signals. Only certain arming modes are allowed for a given measurement function. Appendix D describes allowed combinations of measurement functions and arming modes. Refer to the operating manual for detailed descriptions of the arming modes.The ARMing? query returns the currently active arming mode

Parameters: Arming options are:

PARAMETER	ARMING
AUTOMATIC:	{AUTo}
HOLDOFF OPTIONS: (Auto Sampling)	{EDge Holdoff | Time HOLdoff | EVent Holdoff}
SAMPLING OPTIONS: (No Holdoff)	{Interval SAMpling | Time SAMpling | Cycle SAMpling | Edge SAMpling | Parity SAMpling | Random SAMpling| Repetitive EDGe | Repetitive PARity |
HOLDOFF SAMPLING OPTIONS:	{EDge INterval | EDge Time | EDge EDge | EDge CYcle | EDge EVent | EDge Parity |EDge Random | Time INTerval | Time TIMe | EVent INterval | EVent EVent | Externally GATed | MANual}

Examples: OUTPUT 703;"MEAS;ARM,AUT" - Sets the HP 5373A to use Automatic arming (start the measurement on the next trigger edge, and repetitively sample on each trigger edge).

OUTPUT 703;"MEAS;ARM?" - Queries for the currently selected Arming mode.

BLOC Block Size (command/query)

Shortform: BLOC [BLOCk (size)]
Longform: BLOCK

The BLOCk command sets the number of blocks of measurements taken (the block size).

The BLOCk? query returns the current block size (number of blocks).

Range: 1 to 99,999,999

Examples: OUTPUT 703;"MEAS;BLOC,10" - Sets the HP 5373A to take 10 blocks of measurements, each block consisting of the number of measurements set by the MSIZe command.

OUTPUT 703;"MEAS;BLOC?" - Queries for the number of blocks.

DET Time Interval Detect (command/query)

Shortform: DET [(time interval) DETect]
Longform: DETECT

The DETect command determines the setting of the Time Interval Detect feature. Use this command to enable or disable time interval detection. In addition to enabling time interval detection, this command selects a desired detector configuration.

The TI Detect feature is not available for all functions and armings. When the TI detect feature is unavailable due to a function or arming conflict, the DETect state is not forced off. This way, the intended state does not have to be reprogrammed when the conflict is resolved. If DETect? is performed when TI Detect is not allowed, the response is "UNAV".

- ABOVe means the detector responds to measurements above a set threshold.
- BELow means the detector responds to measurements below a set threshold
- INSide means that the detector responds to measurements above one threshold and below another.

- OUTSide means the detector responds to measurements below the lower threshold or above the upper threshold.

Parameters: {OFF | ABOVe | BELow | INSide | OUTSide}

Example: OUTPUT 703;"MEAS;DET,OFF" - Sets the TI Detect mode to off.

OUTPUT 703;"MEAS;DET?" - Queries for the current TI Detect mode.

DLOW Detector Lower Threshold (command/query)

Shortform: DLOW [Detector LOWer (threshold)]
Longform: DLOWER

The DLOWer command sets the lower detector (threshold) value. This threshold defines the lower boundary condition for the DET INSide and OUTSide cases.

Range: (Refer to *Table 5-5* below for DMIDdle, DUPPer, DLOWer)

Examples: OUTPUT 703;"MEAS;DLOW,4" - Sets lower detect threshold to 4 seconds.

OUTPUT 703;"MEAS;DLOW?" - Queries for the currently selected lower detector (threshold) setting.

Table 5-5. Range Table For DMIDdle, DUPPer, and DLOWer Commands

DMID, DLOW, DUPP	HPMT / PMT		Other functions supporting TI Detect	
	Measurement Mode		Measurement Mode	
	Normal	Fast	Normal	Fast
Minimum	–4 sec	–65 μsec	0 sec	0 sec
Maximum	4 sec	65 μsec	8 sec	131 μsec

DMID Detector Middle Threshold (command/query)

Shortform: DMID [Detector MIDdle (threshold)]
Longform: DMIDDLE

The DMIDdle command sets the middle detector (threshold) value. This threshold defines the boundary condition for DET ABOVe or DET BELow.

Range: (Refer to *Table 5-5* for DMIDdle, DUPPer, DLOWer)

Examples: OUTPUT 703;"MEAS;DMID,0" - Sets middle detector threshold value to zero seconds.

OUTPUT 703;"MEAS;DMID?" - Queries for the currently selected middle detector (threshold) setting.

DUPP Detector Upper Threshold (command/query)

Shortform: DUPP [Detector UPPer (threshold)]
Longform: DUPPER

The DUPPer command sets the upper detector (threshold) value. This threshold defines the upper boundary condition for the DET INSide and DET OUTSide cases.

Range: (Refer to *Table 5-5* for DMIDdle, DUPPer, DLOWer)

Examples: OUTPUT 703;"MEAS;DUPP,4" -Sets upper detector threshold value to 4 seconds.

OUTPUT 703;"MEAS;DUPP?" - Queries for the currently selected upper detector (threshold) setting.

FUNC Function (command/query)

Shortform: FUNC [FUNCtion]
Longform: FUNCTION

The FUNCtion command selects the desired measurement function.

The possible function selections are listed below:

PRF
FREQuency
PRI
PERiod
TOTalize
TIMe or TINTerval (Time Interval)
PMTinterval (Plus or Minus Time Interval)
CTINterval (Continuous Time Interval)
RISetime or RTIMe
FALLtime or FTIMe

PWIDth (Pulse Width)
OFFTime (Pulse Offtime)
DUTY or DCYCle (Duty Cycle)
PHASe
PDEViation (Phase Deviation)
TDEViation (Time Deviation)
FDEViation (Frequency Deviation)
POWer (Envelope Power)
AMPM (AMPlitude Modulation)
HCTinterval (Histogram Continuous Time Interval)
HPMTime (Histogram Plus or Minus Time Interval)
HTIMe (Histogram Time Interval)

Examples: OUTPUT 703;"MEAS;FUNC,TOT" - Sets the HP 5373A function to Totalize.

OUTPUT 703;"MEAS;FUNC?" - Queries for the currently selected measurement function.

NOTE

Programming to a new function can modify other instrument states, although this only happens if the current setting is unavailable for the new function. Refer to Appendix D Arming Modes for a table that lists all the default arming and measurement source settings for each measurement function.

NOTE

The PRF and Frequency function selection commands may be used interchangeably. The PRI and Period function selection commands may be used interchangeably. The HP 5373A will make the final choice of PRF, Frequency, PRI, or Period as its function depending on the combination of function command given and the input pod detected for the specified channel. If the HP 5373A detects an HP 53702A at the input for the specified channel, PRF or PRI will be the selected function, respectively. Otherwise, the function selected will be Frequency or Period, respectively. The FUNC? query will return the selected function, which may not necessarily be the same as the one you specified, as described above in this note.

HBIN Histogram Bin Width (command/query)

Shortform: HBIN [Histogram BIN (width)]
Longform: HBINWIDTH

The HBINwidth command sets the bin width for any of the histogram functions (HCTinterval, HPMTime HTIMe). Setting HBINwid can result in a change of the values associated with the HCENter, HSPan, and HSTart commands (refer to operating manual for more information). The HBINwidth? query returns the currently set value.

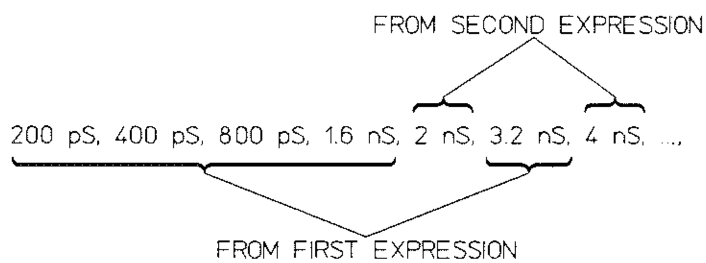
Allowable bin width choices can be computed from the expressions:

200 pS * 2^N where N=0,1,2,....,

and

2 nS * 2^M where M=0,1,2,3...,

The merging of these expressions produces the list of allowed bin widths, which begins with the sequence:



If you program the HBINwidth value between allowed settings, the next larger allowed setting is set.

Table 5-6. Range Limits for Histogram Functions

	HCT HTIM		HPMT	
	NORM (32bit)	**FAST (16 bit)**	**NORM (32 bit)**	**FAST (16 bit)**
HBIN (min)	200 ps	200 ps	200 ps	200 ps
HBIN (max)	1.6777216 ms	64 ns	1.6777216 ms	64 ns
HCEN (min)	200 ns	200 ns	–3.9999998 s	–64.8 µsec
HCEN (max)	7.9999998 s	130.8 µsec	–3.9999998 s	64.8 µsec
HSP (min)	400 ns	400 ns	400 ns	400 ns
HSP (max)	3.3554432 s	128 µsec	3.3554432 s	128 µsec
HST (min)	0 S	0 s	–4.0 s	–65.0 µsec
HST (max)	7.9999996 s	130.6 µsec	3.9999996 s	64.6 µsec

Examples: OUTPUT 703;"MEAS;HBIN,200E-12" - Sets the hardware histogram bin width to 200 pS.

OUTPUT 703;"MEAS;HBIN?" - Queries for the currently selected histogram bin width setting.

NOTE

Histogram commands can interact. The commands, HBINwid, HCENter, HSPan and HSTart each control a parameter that is relevant when the function is HCTinterval, HPMTime, or HTIMe. These parameters are inter-dependent. A detailed description of this inter-dependence is provided in the operating manual

HBL Histogram Block (command/query)

Shortform: HBL [Histogram BLock]
Longform: HBLOCK

Use the HBLock command to establish what will happen between blocks during a multiple block hardware histogram acquisition. If BBIN (Big BIN) is selected, the result of each block is accumulated into an internal software histogram result. The histogram IC is cleared so that each bin can start from zero on the next block. The effect is that a huge bin size is achieved (2^{50} measurements per bin) at the expense of increased time between blocks. If FARM (Fast ARM) is selected, no transfer occurs between blocks. The histogram IC is not cleared, so the result accumulates over all the blocks, but the bin limit is defined by the actual bin count limit on the IC ($2^{24}-1$). The benefit of this mode is that the time between blocks is <2 microseconds.

The HBLOck? query returns the Histogram Block status.

Parameters: {FARM | BBIN}

Examples: OUTPUT 703;"MEAS;HBL,BBIN" - Sets the Histogram Block mode to BBIN (Big Bin).

OUTPUT 703;"MEAS;HBL?" - Queries for the currently selected Histogram Block mode.

HCEN Histogram Center (command/query)

Shortform: HCEN [Histogram CENter]
Longform: HCENTER

The HCENter command sets the histogram center for any histogram functions (HCTinterval, HPMTime, HTIMe). Setting HCENter can result in a change of the value associated with the HSTart command (refer to the Operating Manual for further information about how the HCENter value can interact with HSTart).

The HCENter? query returns the current Histogram Center value.

Range: Refer to *Table 5-6* (you can set values in 200 pS increments).

Examples: OUTPUT 703;"MEAS;HCEN,1E-6" - Sets the Histogram Bin Center to 1 μsecond.

OUTPUT 703;"MEAS;HCEN?" - Queries for the currently selected Histogram Bin Center setting.

HSP Histogram Span (command/query)

Shortform: HSP [Histogram SPan]
Longform: HSPAN

The HSPan command sets the total time spanned for any of the histogram functions (HCTinterval, HPMTime, HTIMe) Setting HSPan can result in a change of the values associated with the HBINwidth, HCENter, or HSTart commands (refer to the Operating Manual for further information about how setting the HSPan value interacts with HBINwid, HCENter, or HSTart.)

The HSPan? query returns the current Histogram Span value.

Range: The Histogram Span is always 2000 times the HBINwidth (bin width) value, thus the allowed choices for HSPan are the allowed choices for HBINwidth multiplied by 2000. The HSPan sequence begins with:

400 nS, 800 nS, 1.6 μsec, 3.2 μsec, 4 μsec

Minimum and maximum limits are listed in *Table 5-6.*

Examples: OUTPUT 703;"MEAS;HSP,400 E-9" - Sets the Histogram Span to 400 nS.

OUTPUT 703;"MEAS;HSP?" - Queries for the currently selected Histogram Span setting.

HST Histogram Start (command/query)

Shortform: HST [Histogram STart]
Longform: HSTART

The HSTart command sets the start time for any of the histogram functions (HCTinterval, HPMTime, HTIMe). Setting HSTart can result in a change of the value associated with the HCENter command (refer to Operating Manuals for further information about how these commands interact).

The HSTart? query returns the current Histogram Start value.

Range: Histogram start time is adjustable in 200 pS increments. Limits are shown in *Table 5-6.*

Examples: OUTPUT 703;"MEAS;HST,1E-3" - Sets the hardware Histogram Start time to 1 millisecond.

OUTPUT 703;"MEAS;HST?" - Queries for the currently selected Histogram Start setting.

INH Inhibit Measurement (command/query)

Shortform: INH [INHibit (measurement)]
Longform: INHIBIT

The INHibit command turns the Inhibit feature on or off. The INHibit? query returns the current Inhibit status.

NOTE

Some instrument setups do not support the Inhibit feature (refer to HP 5373A Reference Function, Arming, Channel Matrix table in Appendix D.) When a setup is selected that does not support Inhibit, the INHibit state is not forced to off. This way, you do not have to remember to reprogram the INHibit state when the setup is later changed to one that does support inhibit. If you query INHibit (INHibit?) when the setup does not support inhibit, the response is UNAV (unavailable). This response is an indication the state of INHibit is not meaningful for the current setup.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"MEAS;INH,ON" - Sets Inhibit feature to on.

OUTPUT 703;"MEAS;INH?" - Queries for the current Inhibit status.

ILEV Inhibit Level (command/query)

Shortform: ILEV [Inhibit LEVel]
Longform: ILEVEL

The ILEVel command sets the Inhibit level value to transistor-transistor logic (TTL), ground (GND, GROund) or emitter coupled logic (ECL). Respective values are:

TTL	1.4 v
Ground	0 v
ECL	–1.3 v

The ILEVel? query returns the current Inhibit Level value.

Parameters: {TTL | GND | GROund | ECL }

Examples: OUTPUT 703;"MEAS;ILEV,TTL" - Sets Inhibit parameter to TTL logic levels.

OUTPUT 703;"MEAS;ILEV?" - Queries for the currently set Inhibit level.

ISEN Inhibit Sense (command/query)

Shortform: ISEN [Inhibit SENse]
Longform: ISENSE

The ISENse command works in conjunction with the ILEVel value. Use the ISENse command to cause the inhibit feature to respond above or below the value specified in ILEVel.

The ISENse? query returns the current Inhibit Sense seting.

Parameters: {ABOVe | BELow}

Examples: OUTPUT 703;"MEAS;ISEN,ABOV" - Sets Inhibit to occur above the value set using ILEVel.

OUTPUT 703;"MEAS;ISEN?" - Queries for the currently set Inhibit Sense.

MSEG? Memory Segmentation (query only)

Shortform: MSEG? [Memory SEGmentation]
Longform: MSEGMENT?

The MSEGment query identifies if the memory segmentation feature is used for the current instrument setup. If in use, the query returns a 1; otherwise the query returns a 0. Segmentation is a condition that exists as a consequence of other settings. You cannot set segmentation, but you can see if it is applicable.

Example: OUTPUT 703;"MEAS;MSEG?" - Queries to determine if Memory Segmentation is currently active.

MSIZ Measurement Size (command/query)

Shortform: MSIZ [Measurement SIZe]
Longform: MSIZE

The MSIZe command sets the number of measurements taken within each measurement block. The maximum MSIZe is dependent on the selected function, the selected measurement source (or sources, as the case may be), and the arming mode. (Refer to *Table 5-7, Maximum Measuerments Per Block* on the next page.) The measurement block size is set by the BLOCk command, up to a maximum of 2E+9 blocks. The product of the measurement size and the number of blocks can never exceed 1E+15.

The MSIZe? query returns the current measurement size.

Range: Refer to *Table 5-7*.

Examples: OUTPUT 703;"MEAS;MSIZ,100" - Sets the number of measurements within each measurement block to 100.

OUTPUT 703;"MEAS;MSIZ?" - Queries for the currently selected measurement size.

Table 5-7. Maximum Measurements per Block (1 of 2)

ARMING MODE	MEASUREMENT FUNCTION											
	TI	CTI	± TI		PRF, FREQUENCY, PRI, PERIOD		TOTALIZE		PULSE WIDTH, OFFTIME, RISETIME, FALLTIME, DUTY CYCLE	PHASE	ENVELOPE POWER, AMPLITUDE MODULATION	PHASE DEVIATION TIME DEVIATION, FREQUENCY DEVIATION
	A	A	A	A → B	A	DUAL[1]	A	DUAL[1]	A	A rel B	A	A
	B	B	B	B → A	B	RATIO[2]	B	RATIO[2]		B rel A	B	B
	A → B				C	SUM[3]		SUM[3]				
	B → A					DIFF[4]		DIFF[4]				
AUTOMATIC												
AUTOMATIC	4096	8191		4096	8191	4095			4095	4095	1	8191
HOLDOFF												
EDGE HOLDOFF	4096	8190		4096	8190					4095		8190
TIME HOLDOFF	4096	8190			8190							
EVENT HOLDOFF	4096	8190			8190							
SAMPLING												
INTERVAL SAMPLING	4096	8191		4096	8191	4095	4095	4095		4095		8191
TIME SAMPLING					1							
CYCLE SAMPLING					8191							
EDGE SAMPLING					8191	4095	4095	4095				
PARITY SAMPLING				4096								
REPET EDGE SAMPLING	4096	8191		4096								
REPET EDGE-PARITY SAMPLING				4096								
RANDOM SAMPLING	4096			4096								
HOLDOFF/SAMPLING												
EDGE/INTERVAL	4096	8090		4096	8090	4095	4095	4095		4095		8190
EDGE/TIME					1							
EDGE/EDGE					8190		4095	4095				
EDGE/CYCLE					8190							
EDGE/EVENT			1	1	1							
EDGE/PARITY				4096								
EDGE/RANDOM	4096			4096								
TIME/INTERVAL					8190		4095					
TIME/TIME			1	1	1							
EVENT/INTERVAL					8190							
EVENT/EVENT			1	1	1							
EXTERNALLY GATED					4096		2048	2048				
MANUAL							1	1				

1. DUAL. Simultaneous Dual-channel, (2 results). Frequency and Period options are: A&B, A&C, B&C. Totalize option is: A&B.
2. RATIO. Frequency and Period ratio options are: A/B, A/C, B/A, B/C, C/A, C/B. Totalize ratio options are: A/B, B/A.
3. SUM. Frequency and Period sum options are: A+B, A+C, B+C. Totalize sum option is: A+B.
4. DIFFERENCE. Frequency and Period difference options are: A–B, A–C, B–A, B–C, C–A, C–B. Totalize difference options are: A–B, B–A.

Table 5-7. Maximum Measurements per Block (2 of 2)

ARMING MODE	MEASUREMENT FUNCTION				
	HISTOGRAM TIME INTERVAL		HISTOGRAM CONTINUOUS TIME INTERVAL	HISTOGRAM ±TIME INTERVAL	
	A→A		A	A→A	A→B
	B→B		B	B→B	B→A
		A→B			
		B→A			
AUTOMATIC	●	●	1		●
EDGE HOLDOFF	●	●	●		●
TIME HOLDOFF	●	●	●		
EVENT HOLDOFF	●	●	●		
INTERVAL SAMPLING	●	●	●		●
PARITY SAMPLING					●
REPET EDGE SAMPLING	●	●	●		●
REPET EDGE-PARITY SAMPLING					●
RANDOM SAMPLING	●	●			●
EDGE/INTERVAL	●	●	●		●
EDGE/PARITY					●
EDGE/RANDOM	●	●			●
EDGE/EVENT				1	1
TIME/TIME				1	1
EVENT/EVENT				1	1

● = 2E+9 measurement blocks

NOTE: Maxima on this page are for the configuration 1 block of N measurements.

PAM Pre-trigger Amount (command/query)

Shortform: PAM [Pre-trigger AMount]
Longform: PAMOUNT

The PAMount command sets the pre-trigger amount. Amount is expressed in percent of block size or as an absolute number. For example, 50 percent means that half of the measurement block will be acquired prior to the Pre-trigger event. Or, if expressed as samples, the HP 5373A will acquire this number of samples prior to the Pre-trigger event.

The PAMount value is not used for hardware Histogram measurement functions (HTT, HCT, HPMT). With these functions, acquisition ends upon occurrence of the Pre-trigger event.

The PAMount? query returns the current Pre-trigger amount.

Parameters: { n | n % | n SAMPle }

Examples: OUTPUT 703;"MEAS;PAM,60" - Sets Pre-trigger amount to 60 percent.

OUTPUT 703;"MEAS;PAM?" - Queries for the currently set Pre-trigger amount.

PBL Pre-trigger Blocks (command/query)

Shortform: PBL [Pre-trigger BLocks]
Longform: PBLOCK

The PBLock command provides block control for pre-trigger measurements. When set to SINGle, a single block is endlessly monitored until the pre-trigger condition occurs. When set to MULTiple, repeated block arm/block acquire cycles occur until the pre-trigger condition occurs. The MULTiple setting allows the arming to be used to qualify the portion of the input to be measured.

The PBLock? query returns the current pre-trigger block control status.

Parameters: {SINGle | MULTiple}

Examples: OUTPUT 703;"MEAS;PBL,SING" - Sets the measurement to SINGle (following the initial block arming, the measurement continues without re-arming).

OUTPUT 703;"MEAS;PBL?" - Queries for the currently selected pre-trigger block control status.

PRET Pre-trigger Control (command/query)

Shortform: PRET [PRE-Trigger (control)]
Longform: PRETRIGGER

The PRETrigger command is used to enable (ON) or disable (OFF) the pre-trigger capability of the HP 5373A (see PAMount, PXSLope for related pre-trigger commands).

A third choice, OCON (On CONditionally) is available for hardware histogram measurement functions (HTIMe, HPMTinterval, HCTinterval). When OCONd is selected, Pre-trigger will be active, but the measurement will also end if the prescribed number of measurements have been taken.

PRETrigger? returns the current value.

Parameters: {ON | OFF | OCONd}

Examples: OUTPUT 703;"MEAS;PRET ON" - Enables the pre-trigger feature.

OUTPUT 703;"MEAS;PRET?" - Queries for the current Pre-trigger state.

PMOD
Pre-trigger mode
(command/query)

Shortform: PMOD [Pre-trigger MODe]
Longform: PMODE

The PMODe command sets the pre-trigger mode. Time (TIMe or TINTinterval) means the pre-trigger occurs when the time interval detector specification is met, EXTernal (X) means the pre-trigger occurs when an external arm trigger occurs.

The PMODe? query returns the current pre-trigger mode.

Parameters: {TIMe | TINTinterval | EXTernal | X }

Examples: OUTPUT 703;"MEAS;PMOD,EXT" - Sets pre-trigger mode to occur on the external arm signal.

OUTPUT 703;"MEAS;PMOD?" - Queries for the currently set pre-trigger mode.

PUN?
Pre-trigger Units
(query only)

Shortform: PUN? [Pre-trigger UNits]
Longform: PUNITS?

The PUNits query returns either "PCT" or "SAMP", indicating the units of the PAMount value.

Example: OUTPUT 703;"MEAS;PUN?" - Queries for the pre-trigger units type.

PXSL Pre-trigger External Arm Slope (command/query)

Shortform: PXSL [Pre-trigger eXternal (arm) SLope]
Longform: PXSLOPE

The PXSLope command controls the external arm slope that defines the pre-trigger event. This command is of importance only when the pre-trigger source is the external arm input (as opposed to the time-interval detector).

The PXSLope? query returns the current pre-trigger external arm slope.

Parameter: {POSitive | NEGative }

Examples: OUTPUT 703;"MEAS;PXSL,POS" - Sets pre-trigger arming to occur on a positive edge.

OUTPUT 703;"MEAS;PXSL?" - Queries for the current pre-trigger external arm slope.

SAMP Sample Arm Sublevel (command)

Shortform: SAMP [SAMPle (arm sublevel)]
Longform: SAMPLE

The SAMPle command selects the arming mode sublevel for selecting sampling arming conditions (for continuous gating measurements) or stop arming conditions (for non-continuous gating measurements). After SAMPle is selected, four subcommands (CHANnel, Delay CHannel, DELay value, and SLOPe) can be used to configure various sample or stop arming conditions.

Example: OUTPUT 703;"MEAS;SAMP" - Selects the SAMPle arm sublevel for setting up sample arm or stop arm conditions.

The Sample Arm Sublevel command has four subcommands: CHANnel, Delay CHannel, DELay value, and SLOPe. These subcommands are described below:

CHAN Arming Channel (command/query)

Shortform: CHAN [(arming) CHANnel]
Longform: CHANNEL

The CHANnel command for the sample arm sublevel selects the input channel to be the source of the sampling or stop arming signal.

The CHANnel? query returns the currently selected arming source for sample arming or stop arming.

Parameters: {A | B | EXTernal | X }

Examples: OUTPUT 703;"MEAS;SAMP;CHAN,A" - Selects Channel A to be the sampling or stop arming signal source.

OUTPUT 703;"MEAS;SAMP;CHAN?" - Queries for the sample arming or stop arming source.

DCH Delay Channel (command/query)

Shortform: DCH [Delay CHannel]
Longform: DCHANNEL

The DCHannel command for the sample arm sublevel selects the input channel to be the source of a delayed arming signal for sampling arming or stop arming.

The DCHannel? query returns the currently selected source for the delay arming signal.

NOTE

The oscillator parameter (O or OSC) for the sample arm DCHannel command can only be selected for cycle sampling (CSAMpling) arming. For all other types of arming, only A or B parameters can be selected.

Parameters: {A | B | OSCillator | O }

Examples: OUTPUT 703;"MEAS;SAMP;DCH,OSC" - Selects the HP 5373A timebase oscillator as the source of a delayed sample arming or stop arming signal.

OUTPUT 703;"MEAS;SAMP;DCH?" - Queries for the input channel source of the delay arming signal.

DEL Delay Value (command/query)

Shortform: DEL [DELay (value)]
Longform: DELAY

The DELay command for the SAMPle arm sublevel selects the number of occurrences required to satisfy the selected sample arm or stop arm conditions (time, events, interval, or cycles). Refer to Appendix D for information about allowable arming modes.

The DELay? query returns the currently selected delay value.

Parameters: The ranges for each type of sample arming and stop arming condition are listed below.

If the Measurement Mode is NORMal:

TIME:	2 nanoseconds to 8 seconds in 2 ns steps
EVENTS:	1 to 4E+9 events in 1 event steps

If the Measurement Mode is FAST:

TIME:	2 nanoseconds to 131 μs in 2 nS steps
EVENTS:	1 to 65,000 events in 1 event steps
INTERVAL:	100 ns to 131 μs in 100 ns steps
CYCLES:	$2^4, 2^8, 2^{12}$

Examples: OUTPUT 703;"MEAS;SAMP;DEL,10" - If Event sample arming is currently selected, this example tells the HP 5373A to sample arm for 10 events. If Time sample arming is currently selected, the time value of 10 would default to the Time range upper limit of 8 seconds.

OUTPUT 703;"MEAS;SAMP;DEL?" - Queries for the current delay value.

SLOP Slope (command/query)

Shortform: SLOP [SLOPe]
Longform: SLOPE

The SLOPe command for the sample arm sublevel selects the slope of the sample arming or stop arming signal.

The SLOPe? query returns the currently selected arming channel slope.

Parameters: {POSitive | NEGative}

Examples: OUTPUT 703;"MEAS;SAMP;CHAN,A;SLOP, POS" - Selects positive edge of Channel A signal for sample arming or stop arming.

OUTPUT 703;"MEAS;SAMP;CHAN,A;SLOP?" - Queries for the Channel A arming slope for sample or stop arming.

SOUR
Measurement Source (command/query)

Shortform: SOUR [(measurement) SOURce]
Longform: SOURCE

The SOURce command for the Measurement subsystem selects the desired input channel configuration for your measurements. The selections allowed depend on the type of measurement. For example, The A/B combination only applies to Frequency, Period, and Totalize measurements. Refer to Appendix D, *Arming Modes* for a table that lists all the possible combinations of measurements and input configurations.

The SOURce? query returns the currently selected measurement source channel.

NOTE

- *(A^B) is equivalent to A&B.*
- *(A<B) is equivalent to A rel B.*
- *(B<A) is equivalent to B rel A.*
- *(A>B) is equivalent to A→B.*
- *(B>A) is equivalent to B→A.*

Parameters: Possible input channel combinations are given in the arming mode table in appendix D, *Arming Modes*

Examples: OUTPUT 703;"MEAS;SOUR,(A>B)" - Selects Channel A→B input configuration for making a Time Interval or plus or minus Time Interval measurement.

OUTPUT 703;"MEAS;SOUR?" - Queries for the currently selected measurement source channel.

SSIZ
Sample Size
(command/query)

Shortform: SSIZ [Sample SIZe]
Longform: SSIZE

The SSIZe command performs the same function as the MSIZe command. In other words, the SSIZe command sets the number of measurements taken within each measurement block. The maximum sample size is dependent on the selected function, the selected measurement source (or sources, as the case may be) and the arming mode. (Refer to *Table 5-7* and Appendix D, *Arming Modes.*) The measurement block size is set by the BLOCk command, up to a maximum of 2E+9 blocks. The product of the measurement size and the number of blocks can never exceed 1E+15.

Range: Refer to *Table 5-7.*

Examples: OUTPUT 703;"MEAS;SSIZ,100" - Sets the sample size within each measurement block to 100.

OUTPUT 703;"MEAS;SSIZ?" - Queries for the currently selected sample (measurement) size.

STAR
Start Arm Sublevel
(command)

Shortform: STAR [STARt (arm sublevel)]
Longform: START

The STARt command selects the arming mode sublevel for selecting block holdoff conditions (for continuous gating measurements) or start arming conditions (for non-continuous gating measurements. After STARt is selected, four subcommands (CHANnel, Delay CHannel, DELay value, and SLOPe) can be used to configure various block holdoff or start arming conditions.

Example: OUTPUT 703;"MEAS;STAR" - Selects the STARt arm sublevel for setting up block holdoff or start arm conditions.

The Start Arm Sublevel command has four subcommands: CHANnel, Delay CHannel, DELay (value), and SLOPe. These subcommands are described below:

CHAN
Channel
(command/query)

Shortform: CHAN [CHANnel]
Longform: CHANNEL

The CHANnel command for the start arm sublevel selects the input channel to be the source of the block holdoff or start arming signal.

The CHANnel? query returns the currently selected arming source for block holdoff or start arming.

Parameters: {A | B | EXTernal | X }

Examples: OUTPUT 703;"MEAS;STAR;CHAN,A" - Selects Channel A to be the block holdoff or start arming signal.

OUTPUT 703;"MEAS;STAR;CHAN?" - Queries for the block holdoff or start arming source.

DCH
Delay Channel
(command/query)

Shortform: DCH [Delay CHannel]
Longform: DCHANNEL

The DCHannel command for the start arm sublevel selects the input channel to be the source of a delayed arming signal for block holdoff or start arming.

The DCHannel? query returns the currently selected source for the delay arming signal

Parameters: {A | B}

Examples: OUTPUT 703;"MEAS;STAR;DCH,B" - Selects Channel B as the source of a delayed block holdoff or start arming signal.

OUTPUT 703;"MEAS;STAR;DCH?" - Queries for the input channel source of the delay arming signal.

DEL Delay (command/query)

Shortform: DEL [DELay]
Longform: DELAY

The DELay command for the start arm sublevel selects the number of occurrences required to satisfy the selected block holdoff or start arm conditions (time or events). Refer to appendices for arming mode information.

The DELay? query returns the currently selected delay value.

Parameters: The ranges for each type of block holdoff and start arming condition are:

If the Measurement Mode is NORMal:

TIME:	2 nanoseconds to 8 seconds (resolution = 2 ns)
EVENTS:	1 to 4E+9 events

If the Measurement Mode is FAST:

TIME:	2 nanoseconds to 131 μs (resolution = 2 ns)
EVENTS:	1 to 65,000 events (resolution = 1 event)

NOTE

An entered Time value greater than 8 seconds defaults to 8 seconds. An entered event count greater than 4E+9 defaults to 4E+9 (in NORMal Mode).

Examples: OUTPUT 703;"MEAS;STAR;DEL,10" - If Event sample arming is currently selected, this example tells the HP 5373A to holdoff for 10 events. If Time holdoff is currently selected, the entered value of 10 would default to 8 seconds.

OUTPUT 703;"MEAS;STAR;DEL?" - Queries for the current delay value.

SLOP
Slope
(command/query)

Shortform: SLOP [SLOPe]
Longform: SLOPE

The SLOPe command for the start arm sublevel selects the slope of the block holdoff or start arming signal.

The SLOPe? query returns the currently selected arming channel slope.

Parameters: {POSitive | NEGative}

Examples: OUTPUT 703;"MEAS;STAR;CHAN,A;SLOP, POS" - Selects positive edge of Channel A signal for block holdoff or start arming.

OUTPUT 703;"MEAS;STAR;CHAN,A;SLOP?" - Queries for the Channel A arming slope for block holdoff or start arming.

TVR
Time Variation Result
(command/query)

Shortform: TVR [Time Variation Result]
Longform: TVRESULT

The TVResult command determines the setting of the Time Variation Result mode. Use this command to specify the kind of Time Variation result you want computed when a multiple block measurement is made that can entirely fit in the measurement memory.

The TVResult? query returns the currently selected result mode.

The parameters mean:

- AVERage: average the Time Variation Result over all the blocks (where result size = block size),
- ALL: every block measurement sequentially appears in the result.

The TVResult? query returns the current Time Variation Result mode.

Note the state of this parameter is forced to AVERage or ALL under certain conditions when a choice is not available.

Parameters: {AVERage | ALL}

Examples:

OUTPUT 703;"MEAS;TVR,AVER" - Sets the instrument to compute the average time variation over all the blocks.

OUTPUT 703;"MEAS;TVR?," - Queries for the current Time Variation Result mode.

Input

INPUT SUBSYSTEM COMMANDS

The INPut subsystem commands control the input characteristics and signal conditioning for the input channels and the external arm channel. The INPut subsystem command functions are equivalent to those available via the front panel INPUT menu screen. The top level syntax diagram for the INPut subsystem commands is shown in *Figure 5-11. Figure 5-12* illustrates the SOURce Sublevel syntax.

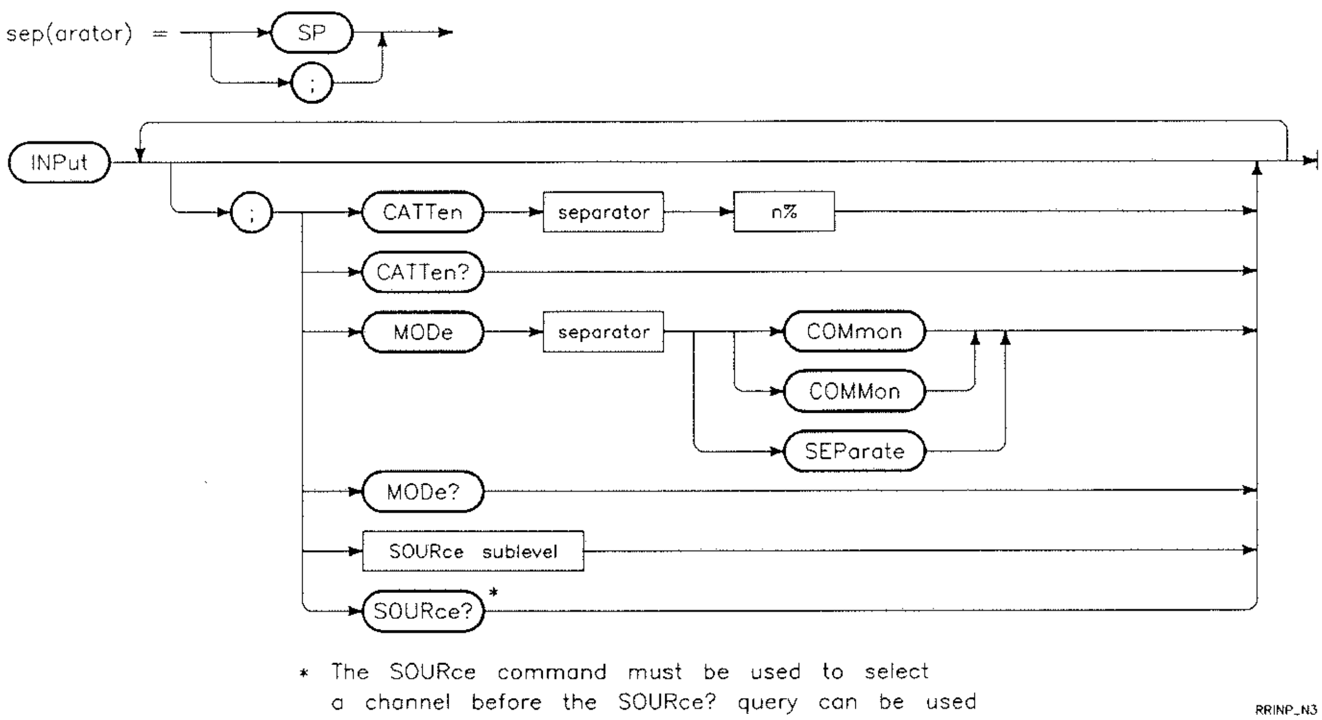


Figure 5-11. INPut Subsystem Syntax Diagram

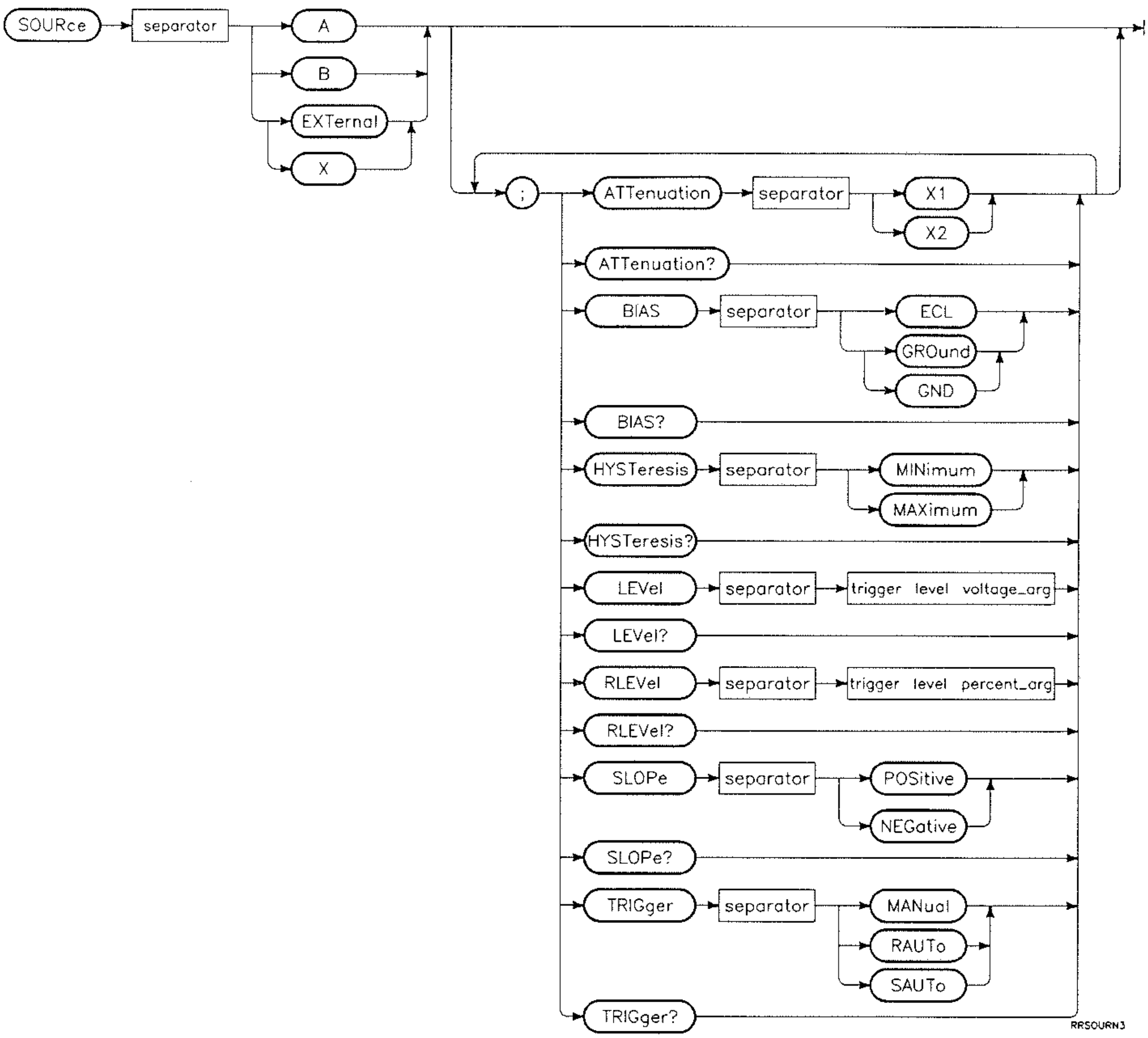


Figure 5-12. SOURce Sublevel Syntax Diagram

CATT
Input C Channel Attenuation
(command/query)

Shortform: CATT [(input) C Channel ATTenuation]
Longform: CATTEN

The CATTen command sets the C channel attenuation. This command differs from the A and B channel attenuation commands which are set by keyword. Instead, the CATTen setting is expressed as a percentage. A reading of 0% means that the minimum amount of attenuation is set; 100% means the maximum setting is seet.

The CATTen? query returns the value of the current C channel attenuation setting.

Range: 0 - 100% (in 5% increments)

Examples: OUTPUT 703;"INP;CATT 50" - Sets the C channel attenuation to 50%.

OUTPUT 703;"INP;CATT?" - Queries for the current C channel attenuation setting.

MOD
Input Mode
(command/query)

Shortform: MOD [(input) MODe]
Longform: MODE

The MODe command selects the input mode for Channels A and B. Select COMMon to simultaneously route the input signal to Channel A and Channel B, or SEParate to use each channel individually. The input impedance and sensitivity specifications are the same for either Separate or Common mode. Note: COMmon is also accepted.

The MODe? query returns the value of the input mode setting.

Parameters: {COMmon | COMMon | SEParate}

Examples: OUTPUT 703;"INP;MOD,SEP" - Sets the input mode to Separate to allow Channel A and Channel B to be used individually.

OUTPUT 703;"INP;MOD?" - Queries for the current input mode.

SOUR Input Source (command/query)

Shortform: SOUR [(input) SOURce]
Longform: SOURCE

The SOURce command selects the current input channel. Subsequent input function selections (such as Attenuation, Slope, and Trigger Level) apply until the source selection is changed. You can select Channel A, Channel B, or External Arm.

The SOURce? query returns the current source selection. The SOURce command must be sent prior to the SOURce? query.

Parameters: {A | B | EXTernal} X |

Examples: OUTPUT 703;"INP;SOUR,B" - Selects Channel B as the current input source

OUTPUT 703;"INP;SOUR?" - Queries for the channel input source.

The Input Source command has eight sublevel commands: ATTenuation, BIAS, CATTen, HYSTeresis, LEVel, Relative LEVel, SLOPe, and TRIGger.

ATT Attenuation (command/query)

Shortform: ATT [ATTenuation]
Longform: ATTENUATION

The ATTenuation command sets the attenuation value for input channels A or B. Channel A or B can be individually set to one of two values: X1 for 0 dB attenuation, or X2 for 8 dB attenuation. At 8 dB attenuation, an input signal can have an amplitude 2.5 times the normal operating range.

Note that attenuation may not be selectable depending on the currently installed input pod.

The ATTenuation? query returns the value of the current attenuation setting.

Parameters: {X1 | X2}

Examples: OUTPUT 703;"INP;SOUR,A;ATT,X2" - Sets Channel A input attenuation to 8 dB.

OUTPUT 703;"INP;SOUR,A;ATT?" - Queries for the current attenuation setting of Channel A.

BIAS
Termination Bias
(command/query)

Shortform: BIAS [(termination) BIAS]
Longform: BIAS

The BIAS command sets the termination (bias) voltage for the 50Ω impedance provided by the standard (HP 54002A) input pod. Either ECL (ECL) or ground (GROund or GND) may be selected. Selecting ECL sets the bias voltage to –2 V to preserve fidelity of ECL input signals. Selecting GROund or GND sets the bias voltage to 0 V.

The BIAS command has no effect for any input channel in which the HP 5373A does not detect an HP 54002A 50Ω Input Pod. The BIAS command effect when an HP 54002A pod is detected is described in the paragraph above. When an HP 53702A Envelope Detector Pod is detected, the HP 5373A automatically sets the bias level to AC. When any input pod other than those described above is installed, the HP 5373A sets the bias level to 0V or GND.

The BIAS? query returns the value of the current termination voltage: "ECL" for –2 V, "GRO" for 0 V, or "AC" when the HP 53702A pod is detected.

Parameters: {ECL | GND | GROund | AC}

NOTE

"AC" is listed here only because it is a possible response for the BIAS? query. It is not recognized when entered as a parameter value for the BIAS command.

Examples: OUTPUT 703;"INP;SOUR,A;BIAS,ECL" - Sets termination voltage to –2 V.

OUTPUT 703;"INP;SOUR,A;BIAS?" - Queries the HP 5373A for the current termination voltage of Channel A.

HYST
Input Hysteresis
(command/query)

Shortform: HYST[(input) HYSTeresis]
Longform: HYSTERESIS

The HYSTeresis command sets the input hysteresis for Channels A or B to the MAXimum or MINimum hysteresis value.

The HYSTeresis? query returns the value of the input hysteresis setting.

Parameters: {MINimum | MAXimum}

Examples: OUTPUT 703;"INP;SOUR,A;HYST,MIN" - Sets the input channel A hysteresis to the minimum level.

OUTPUT 703;"INP;SOUR,A;HYST?" - Queries for the current channel A input hysteresis setting.

LEV
Trigger Level
(command/query)

Shortform: LEV [(trigger) LEVel]
Longform: LEVEL

The LEVel command sets the trigger level for the currently selected input channel when the HP 5373A is in the Manual trigger mode (refer to the TRIGger command description). The desired trigger level in volts is sent with the command. If the HP 5373A is in Repetitive Auto or Single Auto mode, this command is ignored if a normally correct trigger level value is sent. If an incorrect value is sent, an error message is displayed.

The LEVel? query returns the value of the current trigger level for the selected input channel.

53702A Range: X1 attenuation: –0 Vdc to -2.0 Vdc in 2 mV steps, nominal

54002A Range: X1 attenuation: –2.0 Vdc to +2.0 Vdc in 2 mV steps, nominal

54002A Range: X2.5 attenuation: –5.0 Vdc to +5.0 Vdc in 5 mV steps, nominal

54001A Range: - –20.0 Vdc to +20.0 Vdc in 20 mV steps, nominal

External Arm Range: –5.0 Vdc to +5.0 Vdc in 20 mV steps, nominal

Examples: OUTPUT 703;"INP;SOUR,A;LEV,-1" - Sets the Channel A trigger level to –1 V.

OUTPUT 703;"INP;SOUR,B;LEV?" - Queries for the current Channel B trigger level.

RLEV
Relative Trigger Level
(command/query)

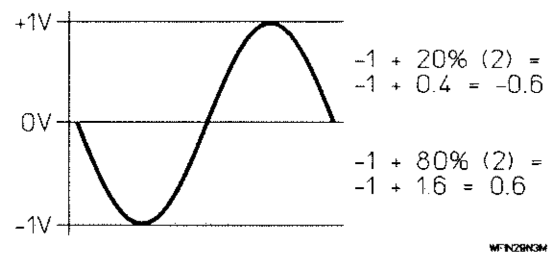
Shortform: RLEV [Relative (trigger) LEVel]
Longform: RLEVEL

The RLEVel command sets the relative trigger level in either Repetitive Auto or Single Auto mode for the selected input channel. The desired relative trigger level (in percentage) is sent with the command. If the HP 5373A is in manual trigger mode, this command is ignored if a normally correct relative trigger level value is sent. If an incorrect value is sent, an error message is displayed.

The trigger points are determined by the following formula:

> Trigger point = minimum peak + (maximum peak – minimum peak) × percentage

Thus a selected relative trigger level of 20% for Channel A and 80% for Channel B applied to a signal that has a maximum peak of +1 V and a minimum peak of –1 V (2 V p-p) results in trigger points of +0.6 V and –0.6 V, respectively. Here is how that result is derived:



The RLEV? query returns the value of the current trigger level for the selected input channel.

Preset: NOMINALLY set to 50% point of input signal

Range: NOMINALLY between maximum and minimum peaks of input signal, in 1% steps

Level Resolution: X1 attenuation: 2mV steps NOMINAL

Level Resolution: X2.5 attenuation: 5 mV steps NOMINAL

Examples: OUTPUT 703;"INP;SOUR,A;RLEV,20" - Sets the Channel A relative trigger level to 20% of the maximum and minimum peaks of the input signal.

OUTPUT 703;"INP;SOUR,B;RLEV?" - Queries for the current Channel B relative trigger level.

SLOP
Slope
(command/query)

Shortform: SLOP [SLOPe]
Longform: SLOPE

The SLOPe command in the Input subsystem selects the positive (rising) or negative (falling) edge of the selected input signal for use as an event for triggering measurements. This trigger slope selection is independent of the arming and gating slope selections.

The SLOPe? query returns the current slope setting for the selected channel.

Parameters: {POSitive | NEGative}

Examples: OUTPUT 703;"INP;SOUR,A;SLOP,POS" - Selects the rising edge of the Channel A input signal for triggering measurements.

OUTPUT 703;"INP;SOUR,B;SLOP?" - Queries for the currently selected slope for Channel B.

TRIG
Trigger
(command/query)

Shortform: TRIG [TRIGger]
Longform: TRIGGER

The TRIGger command selects the form of triggering for the selected input channel. In Manual mode, the LEVel command can be used to set the trigger level voltage; in Repetitive Auto or Single Auto mode, the trigger level is set automatically based on the relative level (in percent) specified using the RLEVel command.

In Single Auto (SAUTo) mode, triggering occurs only once when initially selected; in Repetitive Auto (RAUTo) mode, triggering occurs immediately before each measurement.

Note that Repetitive Auto triggering slows down the overall measurement rate because time measurements cannot be made during the Peak Amplitude measurement phase of Auto triggering.

The TRIGger? query returns the current trigger level mode.

Parameters: {MANual | RAUTo | SAUTo}

Examples: OUTPUT 703;"INP;SOUR,A;TRIG,MAN" - Sets the Channel A input trigger mode to manual.

OUTPUT 703;"INP;SOUR,A;TRIG?" - Queries for the current Channel A input trigger mode.

Process
(Math)

PROCESS (MATH) SUBSYSTEM COMMANDS

The PROCess (MATH) subsystem controls math functions, statistical functions and test limit value setup. The functions in this subsystem are equivalent to those available via the MATH menu screen. The top level syntax diagram for the PROCess (MATH) subsystem commands is *Figure 5-13*. The SOURce sublevel command syntax is illustrated in *Figure 5-14*.

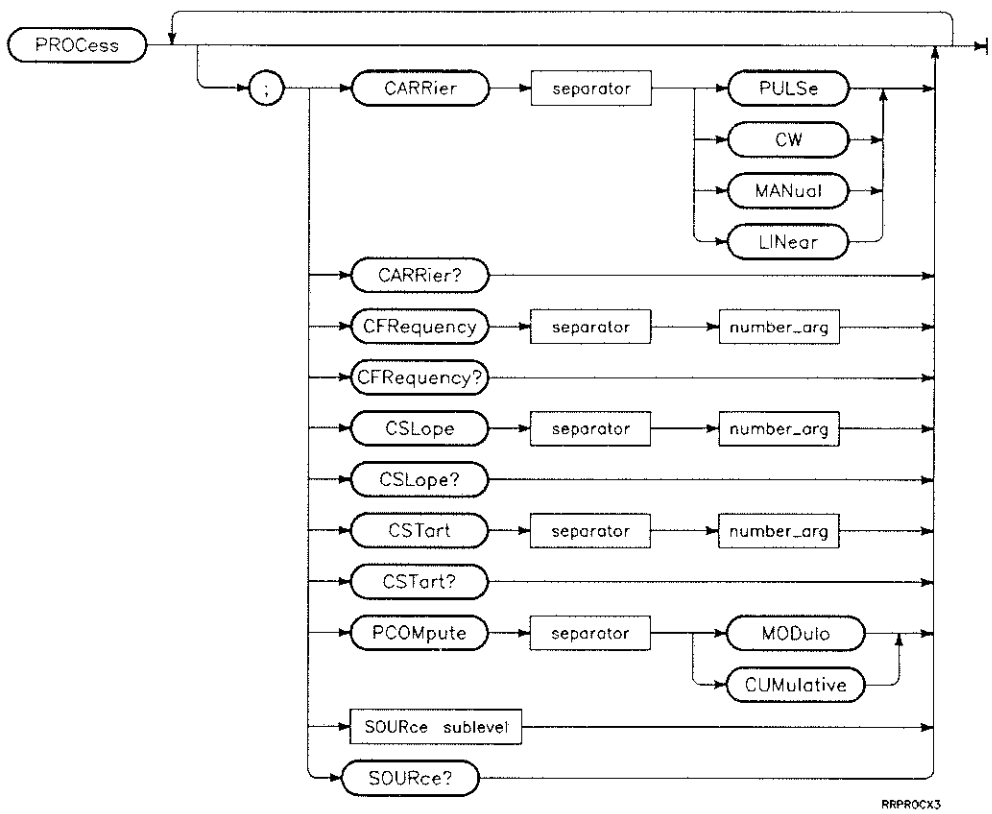


Figure 5-13. PROCess (MATH) Subsystem Syntax Diagram

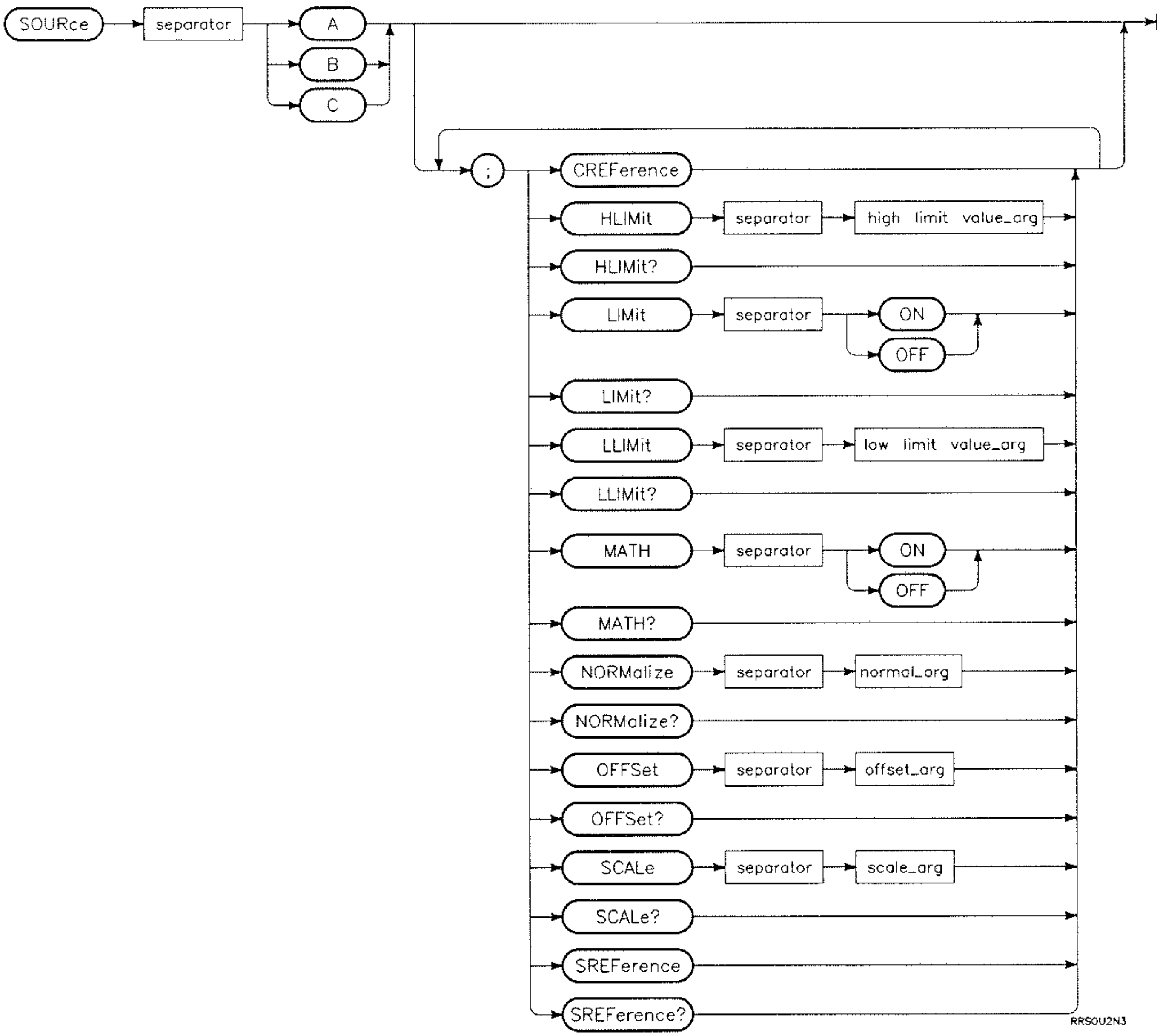


Figure 5-14. SOURce Sublevel Syntax Diagram (1 of 2)

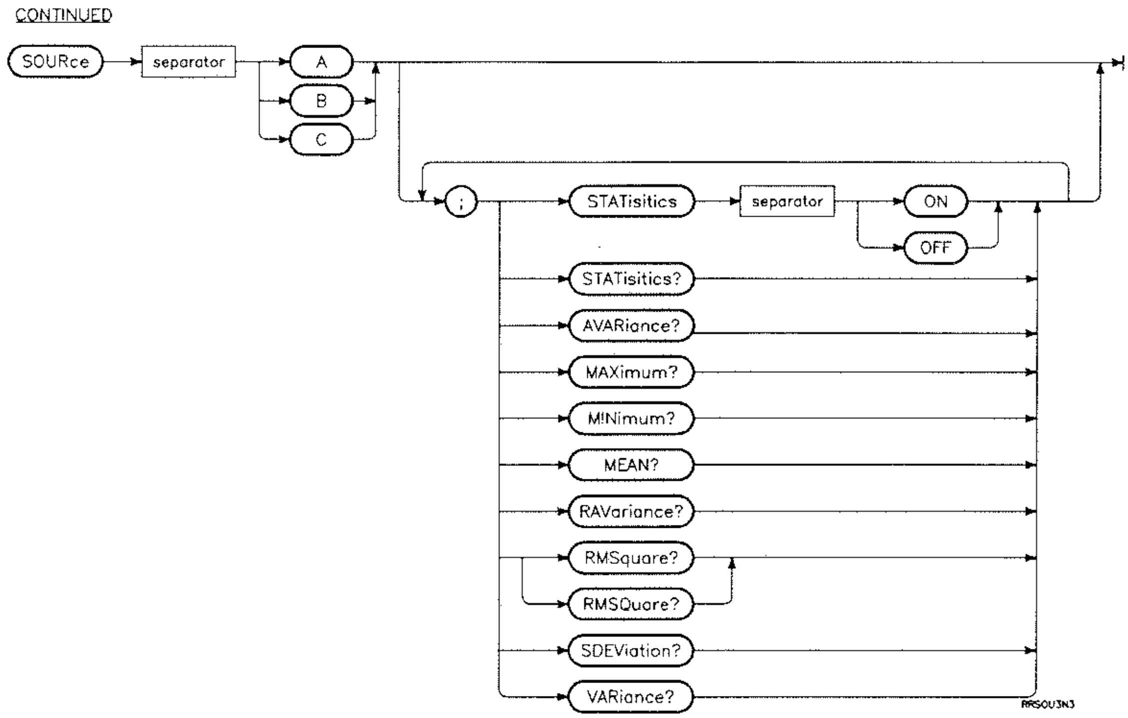


Figure 5-14. SOURce Sublevel Syntax Diagram (2 of 2)

CARR
Set Carrier Control
(command/query)

Shortform: CARR [CARRier]
Longform: CARRIER

The CARRier command specifies how the Carrier Frequency is determined for a Phase Deviation, Time Deviation, or Frequency Deviation measurement. The Carrier Frequency can be:

- calculated automatically, from a pulse input. When this method is chosen, any measurement that includes off-time data is not used in the calculation.
- calculated automatically, from a CW input.
- specified as a constant frequency, entered manually.

  Manual Carrier Range is 1E–12 to 10E+9 (Hz). The HP-IB command is CFR, described below.
- specified as a frequency that changes at a constant rate, determined by manually-entered slope and start values.

  Slope Range is 1E–12 to 10E+9 (Hz/msec).

  Starting Point Range is 1E–12 to 10E+9 (Hz).

  The HP-IB commands for Slope and Starting Point are CSL and CST, and are described below.

NOTE

In the Pulse mode, the on-time and off-time data points are determined as follows:

- *for Frequency, all points whose frequency is less than 1/5 of the maximum frequency are considered to be off-time data points and are rejected.*
- *for PRI, all points whose average period is greater than 5 times the minimum average period are considered to be off-time data points and are rejected.*

The CARRier? query returns the current Carrier Frequency determination mode.

Parameters: {PULSe | CW | MANual | LINear}

Examples: OUTPUT 703;"PROC;CARR,CW" - Selects automatic Carrier Frequency determination from measurement data.

OUTPUT 703;"PROC;CARR?" - Queries for the current Carrier Frequency determination mode.

Comments:

Refer to chapter 3, "Special Purpose Measurements," for a description of the Phase Deviation, Time Deviation, and Frequency Deviation measurement functions.

CFR Carrier Frequency (command/query)

Shortform: CFR [Carrier FRequency]
Longform: CFREQUENCY

Use the CFRequency command to specify the Carrier Frequency for the Manual mode Carrier Frequency manually (CARRier parameter is set to MANual).

The CFRequency? query returns the current Manual mode Carrier Frequency value.

Range (Hz): 1E–12 to 1E+10

Examples: OUTPUT 703;"PROC;CFR,3.14E+6" - Sets the Carrier Frequency to 3.14 MHz.

OUTPUT 703;"PROC;CFR?" - Queries for the current Carrier Frequency value.

CSL Linear Carrier Frequency Slope (command/query)

Shortform: CSL [Carrier SLope]
Longform: CSLOPE

Use the CSLope command to specify the Carrier Frequency Slope value for the Linear Carrier mode (CARRier parameter is set to LINear).

The CSLope? query returns the current Linear mode Carrier Frequency Slope value.

Range (Hz/μsec): 1E–12 to 1E+9

Examples: OUTPUT 703;"PROC;CSL,3.14E+6" - Sets the Linear Carrier Frequency Slope to 3.14 MHz/μsec.

OUTPUT 703;"PROC;CSL?" - Queries for the current Linear mode Carrier Frequency Slope value.

CST Linear Carrier Starting Frequency (command/query)

Shortform: CST [Carrier STart]
Longform: CSTART

Use the CSTart command to specify the Carrier Frequency Starting value for Linear Carrier Frequency mode (CARRier parameter is set to LINear).

The CSTart? query returns the current Linear mode Carrier Frequency Starting value.

Range (Hz): 1E–12 to 1E+9

Examples: OUTPUT 703;"PROC;CST,3.14E+6" - Sets the Linear mode Carrier Frequency Starting value to 3.14 MHz.

OUTPUT 703;"PROC;CFR?" - Queries for the current Linear mode Carrier Frequency Starting value.

PCOM Phase Computation (command/query)

Shortform: PCOM [Phase COMpute]
Longform: PCOMPUTE

Use the PCOMpute command to select the Phase Computation mode. MODulo phase computation provides phase measurement results which have been reduced to a result ≤360°. CUMulative phase computation allows for phase measurement results >360°.

The PCOMpute? query returns the current Phase Computation mode.

Parameters: {MODulo | CUMulative}

Examples: OUTPUT 703;"PROC;PCOM,MOD" - Sets the Phase Computation mode to modulo 360°.

OUTPUT 703;"PROC;PCOM?" - Queries for the Phase Computation mode.

SOUR Source (command/query)

Shortform: SOUR [SOURce (sublevel)]
Longform: SOURCE

The PROCess (MATH) subsystem SOURce command selects an input channel signal (Channel A, B, or C). Subsequent processing function commands then apply to the selected channel..

The SOURce? query returns the current selected channel. The SOURce command must be sent prior to the SOURce? query.

Parameters: {A | B | C }

Examples: OUTPUT 703;"PROC;SOUR,A" - Selects Channel A as the signal source for subsequent processing functions.

OUTPUT 703;"PROC;SOUR?" - Queries for the current signal source.

The PROCess (MATH) subsystem subsystem SOURce sublevel command has 18 subcommands.

CREF Clear Reference (command)

Shortform: CREF [Clear REFerence]
Longform: CREFERENCE

The CREFerence command sets the Reference value to 0. Reference values for each channel are set separately using the SREFerence command.

Example: OUTPUT 703;"PROC;SOUR,A;CREF" - Sets Channel A reference value to 0.

HLIM High Limit (command/query)

Shortform: HLIM [High LIMit]
Longform: HLIMIT

The HLIMit command sets the upper limit for limit testing of the processed results. HLIMit values are set for each channel separately.

Positive Range: 1E–34 ≤n ≤+1E+34, and 0

Negative Range: –1E+34 ≤n ≤–1E–34

Resolution: 1E–34

The HLIMit? query returns the value of the current High Limit. The value is returned in exponential form (NR3 format).

Examples: OUTPUT 703;"PROC;SOUR,A;HLIM,+1E-9" - Sets the High Limit value to 1×10^{-9} for limit testing of Channel A results.

OUTPUT 703;"PROC;SOUR,B;HLIM?" - Queries for the current High Limit value for Channel B.

LIM
Limit Testing
(command/query)

Shortform: LIM [LIMit (testing)]
Longform: LIMIT

The LIMit command selects the limit test mode for setting upper and lower limits on the processed results. Both upper and lower limits can be specified using the HLIMit and LLIMit commands. When a result falls outside the set limits, this will be indicated on the display and over the HP-IB (an SRQ is generated if at least one value is out of range). In this case, the Limit Error bit (LME, bit 7) in the Hardware Status Register will be set to indicate that a limit error has occurred, thus setting the HSB bit (bit 0) in the Status Byte.

The LIMit? query returns the current status of the limit testing mode.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"PROC;SOUR,A;LIM,ON" - Enables limit testing mode for Channel A.

OUTPUT 703;"PROC;SOUR,A;LIM?" - Queries for Channel A Limit Testing status.

LLIM
Lower Limit
(command/query)

Shortform: LLIM [Lower LIMit]
Longform: LLIMIT

The LLIMit command sets the lower limit for limit testing of the processed results. LLIMit values are set for each channel separately.

Positive Range: $+1E{-}34 \leq n \leq +1E{+}34$, and 0

Negative Range: $-1E{+}34 \leq n \leq -1E{-}34$

Resolution: 1E–34

The LLIMit? query returns the value of the current Lower Limit. The value is returned in exponential form (NR3 format).

Examples: OUTPUT 703;"PROC;SOUR,A;LLIM,-1E-9" - Sets the Lower Limit value to -1×10^{-9} for limit testing of Channel A results.

OUTPUT 703;"PROC;SOUR,B;LLIM?" - Queries for the current Lower Limit value for Channel B.

MATH
Math Modifiers
(command/query)

Shortform: MATH [MATH (modifiers)]
Longform: MATH

The MATH command turns on math options for modifying the measurement results. The HP 5373A normally calculates measurement results in fundamental units of Seconds, Hertz, Volts, percent and degrees. Selecting MATH provides the option of using offset, normalize, and scaling functions to modify the results.

To enable the Math modifiers, send "ON"; to disable them, send "OFF". Math modifiers are enabled and disabled separately for each channel. Math modifier values are also set separately for each channel.

The three functions, Offset, Normalize, and Scale, are applied (together with the reference set by the SREFerence command) as follows:

Math result = [(Measurement result – Reference / Normalize) + Offset] ×Scale

If no reference is set (reference = 0), the above formula is equivalent to:

Math result =
[Measurement result / Normalize) + Offset] × Scale

The MATH? query returns the current status of the Math modifiers.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"PROC;SOUR,A;MATH,ON" - Enables the Math modifiers for Channel A.

OUTPUT 703;"PROC;SOUR,B;MATH?" - Queries for the current math status for Channel B.

NORM Normalize (command/query)

Shortform: NORM [NORMalize]
Longform: NORMALIZE

The NORMalize command allows raw measurement data to be divided by a user-entered normalization constant. Normalization values are set separately for each channel.

Positive Range: $+1E{-}12 \le n \le +1E{+}12$

Negative Range: $-1E{+}12 \le n \le -1E{-}12$

Resolution: 1E–12

Note that if zero is entered, the value will default to 1E–12.

The NORMalize? query returns the value of the current Normalization value in exponential form (NR3 format).

Examples: OUTPUT 703;"PROC;SOUR,A;NORM,60" - Sets Channel A Normalization value to 60.

OUTPUT 703;"PROC;SOUR,B;NORM?" - Queries for the current Normalization value for Channel B.

OFFS
Offset
(command/query)

Shortform: OFFS [OFFSet]
Longform: OFFSET

After the application of the Normalization value, the OFFSet command can be used to enter an Offset value to be added to a measurement value. Offset values are set separately for each channel.

Positive Range: +1E–12 ≤n ≤+1E+12, and 0

Negative Range: –1E+12 ≤n ≤–1E–12

Resolution: 1E–12

The OFFSet? query returns the current Offset value in exponential form.

Example: OUTPUT 703;"PROC;SOUR,A;OFFS,-10E-9" - Sets a Channel A Offset value to -10×10^{-9}.

OUTPUT 703;"PROC;SOUR,A;OFFS?" - Queries for the Channel A Offset value.

SCAL
Scale
(command/query)

Shortform: SCAL [SCALe]
Longform: SCALE

The SCALe command sets a Scaling factor for multiplication of the result value. The Scaling factor is applied after Normalization and Offset have been applied. Scaling factors are set separately for each channel. Refer to the MATH command description for information about how the Scaling factor relates to normalization and offset values.

The SCALe? query returns the value of the current Scaling factor in exponential format.

Positive Range: +1E–12 ≤n ≤+1E+12, and 0

Negative Range: –1E+12 ≤n ≤–1E–12

Resolution: 1E–12

Example: OUTPUT 703;"PROC;SOUR,A;SCAL,1E+3" - Sets Channel A Scaling value to 1000.

OUTPUT 703;"PROC;SOUR,A;SCAL?" - Queries for the Channel A Source value.

SREF
Set Reference
(command/query)

Shortform: SREF [Set REFerence]
Longform: SREFERENCE

The SREFerence command is used to assign a Reference value based on the current measurement size and statistics mode. The SREFerence value can be thought of as a negative number added to the measured result, giving a delta value between the measurement and Reference value.

If Statistics are enabled, the Reference value is set to the mean of the last sample set. If Statistics are disabled, the reference value is set to the first valid sample in the last measurement pass. For segmented measurements, the "last measurement pass " refers to the first block; for unsegmented measurements this is the last block.

If SREFerence is selected with Math and Statistics already enabled, the statistical mean used as the Reference value is the mean of the measurement results before any math processing.

There is no numeric entry for this command. If a Reference value other than the last sample or mean is required, the OFFSet command can be used.

Reference values are set separately for each channel. If the SREFerence command is sent for a channel which is not currently being measured, the reference will be set to zero.

The SREFerence? query returns the current value for the specified channel.

CREFerence clears the reference channel.

Examples: OUTPUT 703;"PROC;SOUR,A;SREF" - Sets Reference value for Channel A.

OUTPUT 703;"PROC;SOUR,A;SREF?" - Queries for the current Reference value of Channel A.

STAT
Statistics
(command/query)

Shortform: STAT [STATistics]
Longform: STATISTICS

The STATistics command enables and disables the statistical functions of the HP 5373A. When enabled, all statistical measurements are calculated, even if they are not observed on the currently displayed screen. The results displayed depend on the currently selected menu screen. Any of the eight statistical values may be queried independently of the currently displayed menu screen.

The values calculated are:

- Mean value
- Minimum value
- Maximum value
- Variance
- Standard Deviation
- RMS (Root Mean Square)
- Allan Variance*
- Root Allan Variance*

*The Allan Variance and Root Allan Variance are not calculated if the Inhibit function is enabled.

Statistics mode is enabled separately for each channel.

The STATistics? query returns the current status of Statistics mode.

Parameters: {ON | OFF}

Example: OUTPUT 703;"PROC;SOUR,B;STAT,ON" - Enables the Statistics mode.

OUTPUT 703;"PROC;SOUR,B;STAT?" - Queries for the Current Channel B Statistics status.

NOTE

For ease of reference, all PROCess (MATH) subsystem query-only commands relating to STATistics are grouped together below instead of in the previous alphabetical listing.

There are eight queries relating to Statistics: Allan VARiance?, MAXimum?, MEAN?, MINimum?, Root Allan Variance?, Root Mean SQuare?, Standard DEViation?, and VARiance?. For details of the statistical formulas used, see Chapter 9, Math Menu in the HP 5373A Operating Manual.

NOTE

The HP 5373A will return 1E38 for these queries if statistics are not being calculated for the channel selected with SOURce.

AVAR?
Allan Variance
(query only)

Shortform: AVAR? [Allan VARiance?]
Longform: AVARIANCE?

The AVARiance? query returns the value of the Allan Variance. The Allan variance is calculated by taking the sum of squares of differences between consecutive measurements for the last set of measurements. This value will not be calculated if the Inhibit function is enabled.

Example: OUTPUT 703;"PROC;SOUR,A;AVAR?" - Queries for the Allan Variance value.

MAX?
Maximum
(query only)

Shortform: MAX? [MAXimum?]
Longform: MAXIMUM?

The MAXimum? query returns the maximum value in the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,A;MAX?" - Queries for the maximum value in the last set of measurements.

MEAN?
Mean
(query only)

Shortform: MEAN? [MEAN?]
Longform: MEAN?

The MEAN? query returns the mean value of the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,A;MEAN?" - Queries for the arithmetic mean of the last set of measurements.

MIN?
Minimum
(query only)

Shortform: MIN? [MINimum?]
Longform: MINIMUM?

The MINimum? query returns the minimum value in the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,B;MIN?" - Queries for the minimum value in the last set of measurements.

RAV?
Root Allan Variance
(query only)

Shortform: RAV? [Root Allan Variance?]
Longform: RAVARIANCE?

The RAVariance? query returns the value of the Root Allan Variance (the square root of the Allan Variance) for the last set of measurements. This value will not be calculated if the Inhibit function is enabled.

Example: OUTPUT 703;"PROC;SOUR,B;RAV?" - Queries for the Root Allan Variance value.

RMS? / RMSQ?
Root Mean Square
(query only)

Shortforms: RMS? [Root Mean Square]
RMSQ? [Root Mean SQuare]
Longform: RMSQUARE?

The RMSquare? (or RMSQquare?) query returns the value of the Root Mean Square for the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,B;RMS?" - Queries for the Root Mean Square value.

SDEV?
Standard Deviation
(query only)

Shortform: SDEV? [Standard DEViation?]
Longform: SDEVIATION?

The SDEViation? query returns the value of the Standard Deviation for the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,A;SDEV?" - Queries for the Standard Deviation value.

VAR?
Variance
(query only)

Shortform: VAR? [VARiance?]
Longform: VARIANCE?

The VARiance? query returns the Variance value for the last set of measurements.

Example: OUTPUT 703;"PROC;SOUR,B;VAR?" - Queries for the variance value.

INSTRUMENT STATE SUBSYSTEM COMMANDS

The ISTate subsystem commands control storage and write-protection of stored instrument configuration data. Commands in this subsystem are equivalent to the front panel INSTRUMENT STATE menu screen. The syntax diagram for the ISTate subsystem command (including the REGister sublevel) is illustrated in *Figure 5-15.*

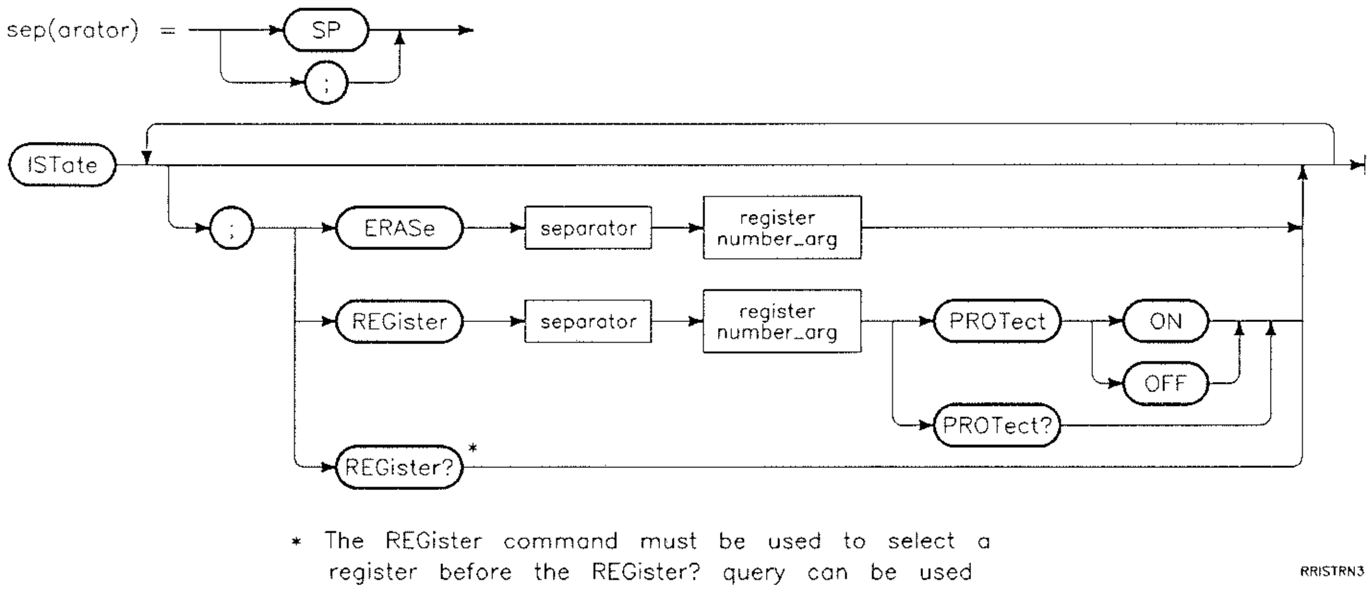


Figure 5-15. Instrument State Syntax Diagram

ERAS
Erase Register
(command)

Shortform: ERAS [ERASe (register)]
Longform: ERASE

The ERASe command erases a specified non-protected register. If a protected register is specified, a "Register protected" error occurs. If Register 0 is specified, a "Register out of range" error occurs.

Parameters: {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

Example: OUTPUT 703;"IST;ERAS,3" - Erases register 3 if that register is not protected.

REG
Register
(command/query)

Shortform: REG [REGister]
Longform: REGISTER

The REGister command specifies the register to be protected by the PROTect command. Register 1 through 9 may be specified for protection. Register 0 is always protected; if specified, a "Register out of range" error occurs.

Refer to the *SAV and *RCL system commands for information about using the registers for saving and recalling instrument setup information.

The REGister? query returns the number of the currently specified register.

Parameters: {1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9}

Examples: OUTPUT 703;"IST;REG,1" - Specifies the register to protect.

OUTPUT 703;"IST;REG?" - Queries for the currently specified register.

The Register command has the following sublevel:

PROT
Protect Register
(command/query)

Shortform: PROT [PROTect (register)]
Longform: PROTECT

The PROTect command selects the protection status for the register specified by the REGister command. A protected register cannot be erased with the ERASe command or written to with the *SAV command. If a protected register is specified with the ERASe or *SAV commands, a "Register protected" error occurs. Registers 1 through 9 can be specified for protection. Register 0 is always protected; if specified, a "Register out of range" error occurs.

The PROTect? query returns the protection status of the currently specified register.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"IST;REG,3;PROT,ON" - Turns on protection for register 3.

OUTPUT 703;"IST;REG,3;PROT?" - Queries for the protection status of Register 3.

INTERFACE (SYSTEM) SUBSYSTEM COMMANDS

The INTerface (SYSTEM) subsystem controls the output data format, the print source for printing or plotting measurement data, and the setting of measurement timeout values. The functions in this subsystem are equivalent to the front panel SYSTEM menu (except you cannot set the HP-IB address over the bus). Syntax diagrams for the INTerface (SYSTEM) subsystem commands are illustrated in *Figure 5-16.* This subsystem does not have sublevels.

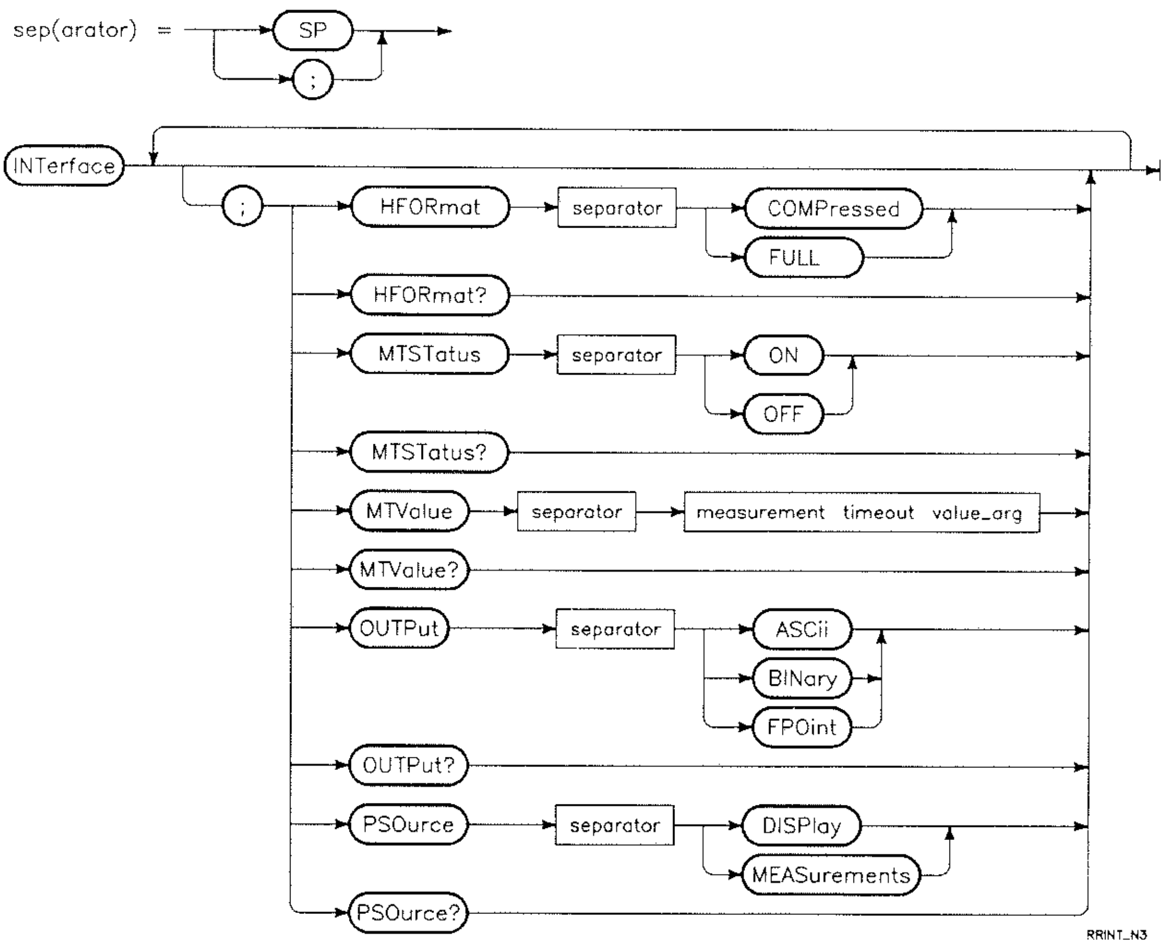


Figure 5-16. INTerface (SYSTEM) Subsystem Syntax Diagram

HFOR Histogram Format (command/query)

Shortform: HFOR [Histogram FORmat]
Longform: HFORMAT

Use the HFORmat command to select the way histogram data is output. If COMPressed is selected, only non-zero valued bins are output. For many acquisitions, this can result in much less total data transfer. With this mode, the results are output in pairs: bin index, value at that index. If FULL is selected, all 2000 bin values are output. With this mode, there is no need to send bin index information, since it is implied by the sequence of the data. This mode allows for direct transfer into a user-defined array.

With either format, length information is provided first, then the actual results. For ASCII output format, the length information is the number of values that follow. For floating point, the length information is the number of bytes that follow (8 times the number of results).

As an example, consider a histogram with 20 hits at bin 52 and 8 hits at bin 955. if COMPressed is selected OHIS? will return:

> 4 (indicating 4 values to follow), 52 (indicating bin position), 20 (indicating occurrences at bin 52), 955, 8

If full is selected, OHIS? will return:

> 2000 (indicating 2000 values to follow), 0, 0, ..., 20 (in 52nd position), 0, 0, ..., 8 (in 955th position), 0, 0, ...

The HFORmat? query returns the status of the Histogram format.

Parameters: {COMPressed | FULL}

Examples: OUTPUT 703;"INT;HFOR,COMP" - Send only non-zero valued bins.

OUTPUT 703;"INT;HFOR?" - Queries for the current status of the HFORmat setting.

MTST Measurement Timeout Status (command/query)

Shortform: MTST [Measurement Timeout STatus]
Longform: MTSTATUS

The MTSTatus command controls the Measurement Timeout function. When MTSTatus is turned ON, a measurement timeout value up to 36000 seconds may be selected using the MTValue command. When the measurement time exceeds the specified timeout value, a message is displayed on the screen to indicate that the measurement has not yet completed. The measurement continues to run to completion, if possible. When MTSTatus is turned OFF, no Measurement Timeout message is displayed.

The MTSTatus? query returns the current status of the Measurement Timeout function.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"INT;MTST,ON" - Turns on the Measurement Timeout function.

OUTPUT 703;"INT;MTST?" - Queries for the current Measurement Timeout status.

MTV Measurement Timeout Value (command/query)

Shortform: MTV [Measurement Timeout Value]
Longform: MTVALUE

The MTValue command is used to set the Measurement Timeout value. A time value up to 36000 seconds may be set. When the selected timeout value is exceeded, a timeout message is displayed on the screen. For example, if a timeout value of 5 seconds is specified and the total measurement time is 10 seconds, a message will be displayed after 5 seconds indicating that the measurement has not yet completed. The MTSTatus function must be turned on for the MTValue command to be used.

The MTValue? query returns the current Measurement Timeout value.

Range: 0 seconds to 36000 seconds, in 1 second steps.

Examples: OUTPUT 703;"INT;MTV,100" - Sets the HP 5373A Measurement Timeout Value to 100 seconds.

OUTPUT 703;"INT;MTV?" - Queries for the Measurement Timeout Value.

OUTP
Output Format
(command/query)

Shortform: OUTP [OUTPut (format)]
Longform: OUTPUT

The OUTPut command selects the output format for sending data to the controller. Either ASCII, Floating Point, or Binary can be selected. Changing the output format will change subsequent data outputs; any data currently in the output buffer will remain in the format that it originated in. Note that switching to Binary output format will empty the output buffer.

The OUTPut? query returns the selected output format. Note that queries are prohibited in Binary output mode, therefore, the response to the OUTPut? query can never indicate the Binary output format.

Parameter: {ASCii | Floating POint | BINary}

Examples: OUTPUT 703;"INT;OUTP,ASC" - Tells the HP 5373A to output data to the controller in ASCII measurement format.

OUTPUT 703;"INT;OUTP?" - Queries for the selected output format.

PSO
Print Source
(command/query)

Shortform: PSO [Print SOurce]
Longform: PSOURCE

The PSOurce command selects the data source for generating a hardcopy output using the PRINt or PLOT commands. When using the PRINt command, MEASurements can be selected to get a copy of the measurements results, or DISPlay can be selected to get a copy of the current screen. When using the PLOT command, only DISPlay can be chosen as the data source.

The PSOurce? query returns the currently selected data source.

Parameters: {DISPlay | MEASurements}

Examples: OUTPUT 703;"INT;PSO,DISP" - Selects the screen as the data source for printing.

OUTPUT 703;"INT;PSO?" - Queries for the selected data source.

Diagnostic
(Test)

DIAGNOSTIC (TEST) SUBSYSTEM COMMANDS

The DIAGnostic (TEST) subsystem performs selected diagnostic tests on the HP 5373A to verify that functions work properly and to identify faulty parts or assemblies. These diagnostic functions are identical to those available via the front panel TEST menu. A subset of these tests occurs automatically when you set power to ON.

The syntax diagram for the DIAGnostic (TEST) subsystem is illustrated in *Figure 5-17* This subsystem does not have any sublevel commands.

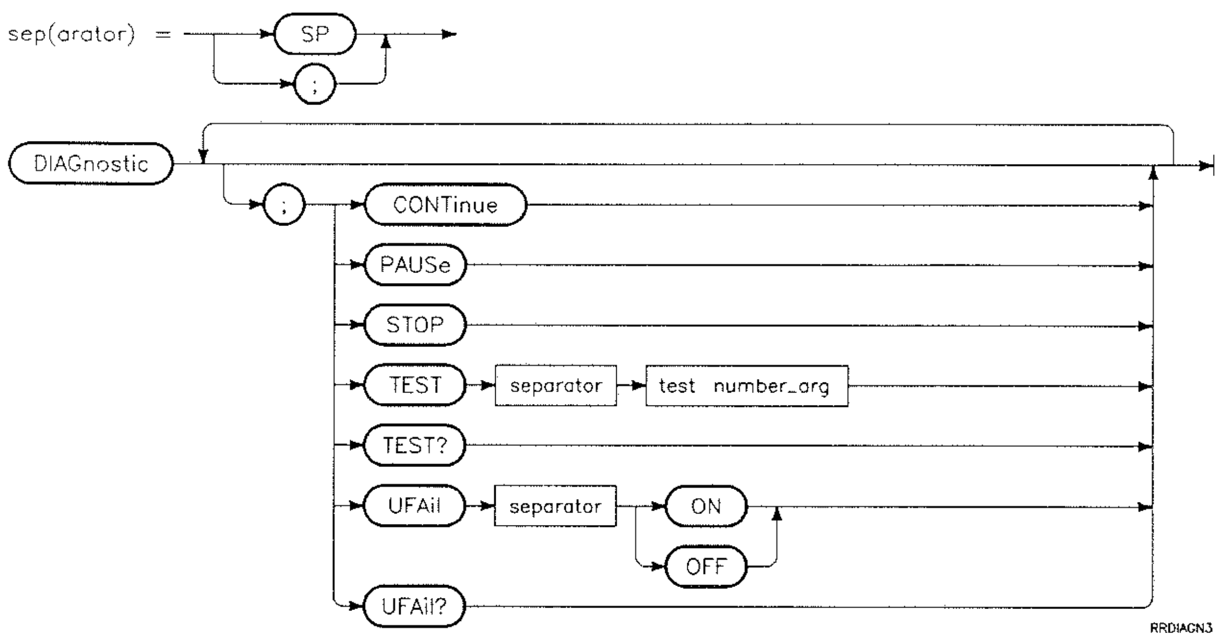


Figure 5-17. DIAGnostic (TEST) Subsystem Syntax Diagrams

CONT Continue Test (command)

Shortform: CONT [CONTinue (test)]
Longform: CONTINUE

The CONTinue command resumes execution of a test suspended by the PAUSe command.

Example: OUTPUT 703;"DIAG;CONT" - Continues the last diagnostic test before the PAUSe command.

PAUS Pause Test (command)

Shortform: PAUS [PAUSe (test)]
Longform: PAUSE

The PAUSe command pauses the currently running test until a CONTinue command is received.

Example: OUTPUT 703;"DIAG;PAUS" - Suspends the current diagnostic test.

STOP Stop Test (command)

Shortform: STOP [STOP]
Longform: STOP

The STOP command terminates the execution of the currently running diagnostic test.

Example: OUTPUT 703;"DIAG;STOP" - Terminates the currently running test.

TEST Run Test (command/query)

Shortform: TEST [(run) TEST]
Longform: TEST

The TEST command executes the diagnostic corresponding to the test number sent with the command. The test numbers are:

1. Self Test
2. Time Base
3. Input Pods
4. Input Amplifiers
5. Histogram
6. Count ICs
7. Gate Timer
8. Measurement RAM
9. System ROM
10. System RAM
11. Non-volatile RAM
12. Real Time Clock
13. Coprocessor
14. CRT RAM
15. LED Latch
16. CRT Controller
17. Key Controller
18. DMA Controller
19. Front Panel
20. CRT Adjustment
21. CRT Video Pattern
22. External Amp
23. Randomizer
24. Calibrate Interps
25. Cal. Sensitivity

The TEST? query returns the PASS/FAIL status and message of the last diagnostic test that was run.

Examples: OUTPUT 703;"DIAG;TEST,12" - Executes the Real Time Clock diagnostic test.

OUTPUT 703;"DIAG;TEST?" - Queries the HP 5373A for the status and message of the last test.

UFA Run Until Fail (command/query)

Shortform: UFA [(run) Until FAil]
Longform: UFAIL

The UFAil command selects the Until Fail testing mode. In this mode, the HP 5373A continues running a diagnostic test until a failure occurs, at which point the test is paused. To enable the Until Fail mode, send "ON"; to turn off the mode, send "OFF".

After a test has been paused, it can be continued by sending the CONTinue command or by pressing the "Run" softkey on the displayed test screen.

The UFAil? query returns the current status of the Until Fail mode.

Parameters: {ON | OFF}

Examples: OUTPUT 703;"DIAG;UFA,ON" - Runs the currently selected diagnostic test and pauses if a failure occurs.

OUTPUT 703;"DIAG;UFA?" - Queries for the current on/off status of the Until Fail mode.

A HP-IB INTERCONNECTION

APPENDIX

A

HP-IB INTERCONNECTION

INTRODUCTION

The HP 5373A has two HP-IB operating modes:

- Talk/Listen — This mode is for bi-directional communication. The HP 5373A can receive commands and setups from the controller, and can send data and measurement results.
- Talk Only — In this mode, the HP 5373A can send data and measurement results; it cannot receive commands or setups from the controller.

Refer to the operating manual for instructions on how to change the HP-IB operating modes. Section 2 of this programming manual describes how to set the HP 5373A HP-IB address.

CONNECTING TO A CONTROLLER

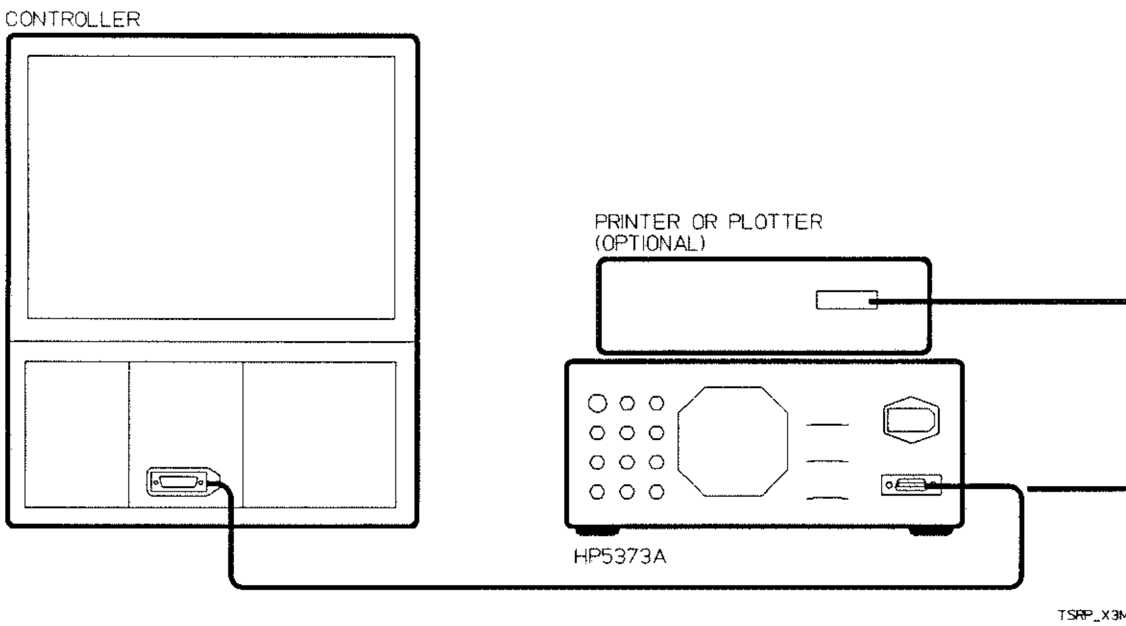
Connect the HP 5373A to a controller by simply installing an HP-IB cable (such as an HP 10833A cable) between the two units *(Figure A-1).*

Figure A-1, (2 of 2) provides interconnection data about the rear panel HP-IB connector on the HP 5373A. This connector is compatible with the HP 10833A/B/C/D cables. Up to 15 HP-IB compatible instruments (including the controller) can be interconnected in a system. The HP-IB cables have identical "piggy-back" connectors on both ends so that several cables can be connected to a single source without special adapters or switch boxes. System components and devices may be connected in virtually any configuration desired. There must, of course, be a path from the controller to every device operating on the bus.

CAUTION

AVOID stacking more than three or four cables on any one connector. Multiple connectors produce leverage that can damage a connector mounting.

Using a screwdriver can damage the threads inside the head of the lockscrew. Instead of using a screwdriver, finger-tighten connector lockscrews so they cannot come loose during operation.

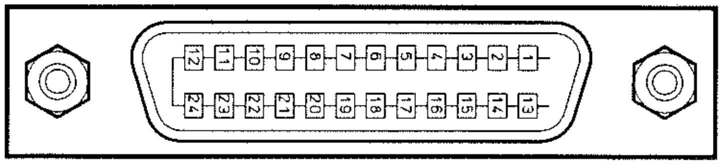


Model Number	Cable Length
10833A	1 metre (3.3 ft.)
10833B	2 metres (6.6 ft.)
10833C	4 metres (13.2 ft.)
10833D	0.5 metres (1.6 ft.)

Figure A-1. Interconnection Data (1 of 2)

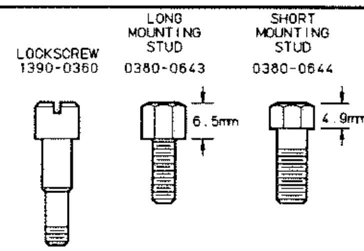
PIN	LINE
1	DIO1
2	DIO2
3	DIO3
4	DIO4
13	DIO5
14	DIO6
15	DIO7
16	DIO8
5	EOI
17	REN
6	DAV
7	NRFD
8	NDAC
9	IFC
10	SRQ
11	ATN
12	SHIELD-CHASSIS GROUND
18	P/O TWISTED PAIR WITH PIN 6
19	P/O TWISTED PAIR WITH PIN 7
20	P/O TWISTED PAIR WITH PIN 8
21	P/O TWISTED PAIR WITH PIN 9
22	P/O TWISTED PAIR WITH PIN 10
23	P/O TWISTED PAIR WITH PIN 11
24	ISOLATED DIGITAL GROUND

THESE PINS ARE INTERNALLY GROUNDED (pins 18–23)



CAUTION

The 5373A contains metric threaded HP-IB cable mounting studs as opposed to English threads. Metric threaded HP 10833A, B, C, or D HP-IB cable lockscrews must be used to secure the cable to the instrument. Identification of the two types of mounting studs and lockscrews is made by their color. English threaded fasteners are colored silver and metric threaded fasteners are colored black. DO NOT mate silver and black fasteners to each other or the threads of either or both will be destroyed.



Logic Levels
The Hewlett-Packard Interface Bus logic levels are TTL compatible, i.e., the true (1) state is 0.0V dc to +0.4V dc and the false (0) state is +2.5V dc to +5.0V dc.

Mating Connector
HP 1251-0293; Amphenol 57-30240.

Mating Cables Available
HP 10631A, 1 metre (3.3 ft.), HP 10631B, 2 metres (6.6 ft.)
HP 10631C, 4 metres (13.2 ft.), HP 10631D, 1/2 metre (1.6 ft.)

Cabling Restrictions

1. A Hewlett-Packard Interface Bus System may contain no more than 2 metres (6.6 ft.) of connecting cable per instrument.
2. The maximum accumulative length of connecting cable for any Hewlett-Packard Interface Bus System is 20.0 metres (65.6 ft.).
3. The maximum number of instruments in one system is fifteen.

Figure A-1. Hewlett-Packard Interface Bus Connection (2 of 2)

CONNECTING TO A PRINTER OR PLOTTER

You can connect the HP 5373A to a printer or a plotter, either with or without a controller. If you connect a controller, you can operate the printer or plotter under program control. In the absence of a controller, the HP 5373A, when in Talk Only, becomes the controller for the printer or plotter.

When operating a printer or plotter without a controller, set the HP 5373A HP-IB mode to Talk Only. When operating in Talk Only mode, the printer or plotter must be in Listen Only or Listen Always mode, and may be set to output a service request (SRQ). Refer to the respective printer or plotter manual for these settings.

The Print/Plot program in *Appendix E Program Examples* controls printing and plotting features of the HP 5373A (when using a controller) This program takes advantage of the Hardware Status Register and Service Request Enable (SRE) register to determine when a hardcopy is complete.

B DEFAULT MEASUREMENT SETUPS

APPENDIX

B

DEFAULT MEASUREMENT SETUPS

DEFAULT MEASUREMENT SETUPS

The Default Measurement Setup function (DMS command) sets a configuration most likely to give valid results for the currently chosen measurement function. When the DMS function is selected using the DMS command or the front panel SHIFT, PRESET keys), certain values are preset to default conditions. These conditions include:

- source channel
- sample size
- arming (and the associated parameters)
- input channel specifications (trigger modes, levels).

Math modifiers are disabled, Statistics is enabled, and the Numeric (Results + Statistics) screen is displayed. With two keypresses (Shift, Preset) you can set up a measurement likely to obtain valid measurement results, plus statistical analysis results.

General Conditions

When going into DMS mode, the following conditions are true for all measurement types, with an HP 53702A Envelope Detector Pod in CHANNEL A and an HP 54002A 50Ω Input Pod in CHANNEL B:

General (applies to all functions)

Prior setup stored in register 0

Acquisition mode	Repetitive
Numeric Display Mode	Split (results and statistics)
Displayed Menu	Numeric Screen
Math Channel A,B,C	Off
Statistics Channel A,B,C	On
Limit Test Channel A,B,C	Off
Reference value A,B,C	0

Block count	1
Bias, channel A	AC
Bias, channel B	GND
Attenuation A,B	X1
Pre-trigger	Off
Carrier	Pulse
Inhibit	Off
Hysteresis A,B	Min
Attenuation C	0%
Trigger Mode A,B	Repetitive Auto Trigger
Ext arm trigger level	0 V

Function Specific:

1. Time Interval, Continuous Time Interval, Time Deviation, Phase Deviation, PRF, Frequency, PRI, Period

Block size	100
Slope A,B	Positive
Input Mode	Separate
Measurement channel	A
Arming	Automatic
Channel A Auto trig lvl	50%
Channel B Auto trig lvl	50%

2. Histogram Time Interval, Histogram Continuous Time Interval:
   Same as 1, except

Block size	1,000,000

3. Envelope Power, Amplitude Modulation:
   Same as 1, except

Block size	1

4. ±Time Interval:
   Same as 1, except

Measurement channel	A→B

5. Histogram ±Time Interval:
   Same as 1, except

Measurement channel	A→B
Block size	1,000,000

6. Totalize:
   Same as 1, except

Arming	Interval Sampling

7. Rise Time: (with HP 53702A pod installed)

Block size	100
Slope A,B	Negative

Input Mode	Common
Measurement channel	A
Arming	Automatic
Channel A Auto trig lvl	80%
Channel B Auto trig lvl	20%

8. Fall Time: (with HP 53702A pod installed)

Block size	100
Slope A,B	Positive
Input Mode	Common
Measurement channel	A
Arming	Automatic
Channel A Auto trig lvl	20%
Channel B Auto trig lvl	80%

9. Pulse Width (with HP 53702A pod installed), Duty Cycle (with HP 53702A pod installed):

Block size	100
Slope A	Negative
Slope B	Positive
Input Mode	Common
Measurement channel	A
Arming	Automatic
Channel A Auto trig lvl	50%
Channel B Auto trig lvl	50%

10. Pulse Offtime :

Same as 7, except:

Slope A	Positive
Slope B	Negative

11. Phase:

Block size	100
Slope A	Negative
Slope B	Positive
Input Mode	Separate
Measurement channel	A relative to B
Arming	Edge Holdoff
Start channel	A
Start channel slope	Positive
Channel A Auto trig lvl	50%
Channel B Auto trig lvl	50%

C STATUS AND ERROR MESSAGES

APPENDIX
C

STATUS AND ERROR MESSAGES

INTRODUCTION

Five types of messages appear on the CRT Status Line of the HP 5373A to indicate errors, failures, and general information. This appendix contains a complete list of all messages in alphabetical order. The messages have a priority order; from the lowest to the highest priority, they are:

- Static Status Messages
- Momentary Status Messages
- Momentary Warning Messages
- Static Error Messages
- Static Failure Messages

These messages are acknowledged in different ways, depending on whether the instrument can continue operating and making measurements. Some messages are only warnings, and normal operation can continue without user response. Others are intended to notify the user that operation has been suspended until the error condition is acknowledged and/or corrected.

Displayed messages are replaced with ones of higher priority. For example, a Static Error Message will overwrite a Momentary Warning Message. If another message occurs with the same priority as the current message, overwriting occurs only if both are Status or if both are Momentary messages. Otherwise, the second message will be ignored.

STATIC STATUS MESSAGES

These are information messages to inform you of the condition of the instrument. Static Status Messages are cleared when the condition in the instrument changes, or when the RESTART key is pressed.

MOMENTARY STATUS MESSAGES

These are information messages to alert you to a particular condition in the instrument. They will clear automatically after three seconds.

MOMENTARY WARNING MESSAGES

These are warning messages to alert you to an illegal operation that was attempted. They will clear automatically after three seconds. Examples of actions that generate a Momentary Warning Message are: pressing an undefined softkey, or pressing a non-numeric key while entering a numeric value. An error number will be placed in the Error Queue for each Momentary Warning Message.

STATIC ERROR MESSAGES

Static errors convey improper operating states or errors generated by HP-IB. Pressing a valid key clears these errors. HP-IB errors are cleared on the transition from REMOTE to LOCAL. Examples of Static Errors are: sending an invalid HP-IB command, or attempting to query the instrument while it is in Binary output mode. An error number will be placed in the Error Queue for each Static Error Message.

STATIC FAILURE MESSAGES

These are failures which prevent the instrument from operating properly. They convey "catastrophic" hardware-related failure conditions. Static failures must be acknowledged by pressing the RESTART key or by correcting the failure condition. Examples of Static Failures are: applying too much voltage to the Channel A or B input pods, or neglecting to power-down the instrument before removing one of the input pods. An error number will be placed in the Error Queue for each Static Failure Message.

ERROR QUEUE QUERY COMMAND

The Error Queue query command ("ERR?") allows an HP-IB system controller to request the contents of the HP 5373A's Error Queue. The Error Queue contains a maximum of 16 error numbers, represented by integer values, which identify operator or hardware errors. If more than 16 errors have been queued but not queried, then the 16th one is replaced with Error -350 (which is the HP standard error number indicating that "too many errors have occurred"). Successively sending the query "ERR?" returns error numbers in the order that they occurred, until the queue is empty. Additional queries return an error of "0" until another error condition is generated. Only messages of the type Momentary Warning, Static Error and Static Failure have error numbers that are entered into the Error Queue.

STATUS AND ERROR MESSAGE DESCRIPTIONS

The following list describes the HP 5373A system-wide status and error messages in alphabetical order. HP standard numbers, which are consistently defined for all HP instruments, are listed under "Error", and are preceded with a minus sign (for example, "Error -120: Numeric argument error"). All of the other error messages with positive numbers are unique to the HP 5373A. Messages with numbers are used to indicate actual events which have occurred which should be logged in the Error Queue. Messages without error numbers are intended for status information only.

Note that there are also some localized messages in the Graphics screens which are not covered here in detail. Those messages are intended to give the user feedback on the current Graph status, but are not generally considered errors of global concern, and do not generate error number entries in the Error Queue. Examples are: the number of measurements between the markers, the status indication while the graph display is being recalculated (due to a GRAPhic command), or an indication that some action has completed (such as a Graph copy to memory). These messages are considered to be self-explanatory and therefore are not listed here.

Abort only allowed in Single

Type: Momentary Status *Associated With:* Measurement

This message occurs when the "ABORT" HP-IB command is received or the Abort (MANUAL ARM) key is pressed while the HP 5373A is in the Repetitive Sample mode. The Abort command is valid only when the HP 5373A is in Single Sample mode, so no action takes place.

Acquiring measurement data

Type: Static Status *Associated With:* System Operation

This message occurs while the HP 5373A is acquiring measurement data. It is erased when the acquisition process is complete.

Alternate Timebase selected. Press RESTART.

Type: Static Failure *Associated With:* Rear Panel External Reference Input
Error Number: 105

The HP 5373A will display this message and stop the measurement process when an external timebase reference is connected to, or disconnected from, the rear panel EXTERNAL

INPUT connector. Press the front panel RESTART key or send the HP-IB "RESTART" command to restart measurements. If the message was caused by connecting the external reference, the restarted measurements will be based on that external reference, otherwise, they will be based on the Internal Timebase.

Arming has changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when the Arming selection has been changed to resolve conflict with another parameter that has been entered (such as changing Measurement Function).

Arming parameter changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when an Arming parameter has been changed to resolve conflict with another parameter that has been entered. Examples of Arming parameters that might be changed are: Arming channel or delay value.

Arming value changed by Fast Meas mode.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when an Arming value has been changed to accomodate the Fast Measurement mode. This mode restricts arming to occur within 131 μsec.

Arming, input parameters changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when both the Arming selection and one or more Input menu parameters have been changed to resolve conflict with another parameter that has been entered. Examples of Input menu parameters that might change are: Trigger Mode, Trigger Slope or Trigger Level.

Arming, measurement source have changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when both the Arming selection and Measurement Source channel have been changed to resolve conflict with the Measurement Function that has been entered.

Binary output turned off.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when the instrument is in the Binary output mode, and the Envelope Power or Amplitude Modulation function has been selected. Binary output mode is not supported for Envelope Power or Amplitude Modulation measurements. The instrument defaults to ASCII output mode.

Block or Measurement size changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs if a Block Size or Measurement Size is entered which causes the total number of measurements to exceed 2E+15. This is applicable to Histogram TI modes in particular. When this happens, the entered parameter is allowed, but the other is defaulted to keep the total acquisition size less than 2E+15 measurements. For example, if the Measurement Size is 20,000,000 measurements, the maximum Block Size enterable is 99,999,999 (to insure that the total is less than 2E+15). If the Measurement Size is increased (so that it is greater than 20,000,000), the corresponding Block Size is decreased to keep the total less than 2E+15 measurements.

Calculating measurements.

Type: Static Status *Associated With:* System Operation

This message is displayed while the HP 5373A is calculating the measurement results. It is erased when the calculation process is complete.

Decimal point entry disallowed.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when a decimal point is not allowed at this point in the current numeric entry sequence, because the exponent value has already been specified (e.g. the value currently being entered is "1.2E+01").

Decimal point previously entered.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when a decimal point is not allowed at this point in the current numeric entry sequence, because a decimal point has already been entered (e.g. the value currently being entered is "1.2").

Enter register number.

Type: Static Status *Associated With:* Save/Recall

This message appears after pressing the SAVE or RECALL keys, prompting the user to select one of the saved configuration registers.

Error -100: Unrecognized command.

Type: Static Error *Associated With:* Standard HP Error

This message occurs when an invalid command has been sent via HP-IB. Examples are: commands not valid for the currently specified subsystem, commands not allowed for the current measurement setup, or commands containing syntax errors.

Error -120: Numeric Argument error.

Type: Static Error *Associated With:* Standard HP Error

This message occurs when an attempt has been made to enter a Stop arming value less than the corresponding Start arming value in TIME/TIME or EVENT/EVENT Arming modes. Examples are: entering a Start time greater than a Stop time, or entering a Start event count greater than a Stop event count.

Error -151: Query not allowed. Binary format.

Type: Static Error *Associated With:* Standard HP Error

This message occurs when output data is requested from the HP 5373A while it is in the Binary output mode. The HP 5373A cannot be queried or send formatted numeric data via HP-IB while in Binary output mode. To process queries, switch to ASCII or Floating Point output modes.

Error 100: No Listeners on bus.

Type: Momentary Warning *Associated With:* HP-IB

This message occurs when there are no listeners present on the bus, and an attempt has been made to have the HP 5373A send output. This is specific to the Talk/Listen mode of operation.

Error 101: Talker, no listeners.

Type: Momentary Warning *Associated With:* HP-IB

This message occurs when the HP 5373A is addressed to talk, but there are no listeners present on the bus. This is specific to the Talk-only mode of operation.

Error 102: Bus conflict: Talk-only.

Type: Static Error *Associated With:* HP-IB

This message occurs when an attempt is made to send HP-IB commands to the HP 5373A while it is in Talk-Only mode. The instrument cannot accept commands via HP-IB while in this mode. To allow the HP 5373A to accept commands, return to Talk/Listen mode.

Error 103: Key ignored in Remote.

Type: Momentary Warning *Associated With:* HP-IB

This message occurs when a front panel key is pressed while the HP 5373A is in Remote mode. While in Remote, all front panel keys except LOCAL are disabled.

Error 104: Key ignored in LLO.

Type: Momentary Warning *Associated With:* HP-IB

This message occurs when the LOCAL key is pressed while the HP 5373A is in Remote, and in Local Lockout mode. In Local Lockout mode, the LOCAL key is disabled.

Error 107: Timebase unlocked. Press RESTART.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when the oscillator is out of lock. Any measurements made while this message is on the screen may not be accurate.

Error 108: Ch A and B Overvoltage.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when an overvoltage condition is present on both Input Channels A and B. To correct this error, change the signal Trigger Level, Bias or Attenuation on the Input menu.

Error 109: Ch A Overvoltage.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when an overvoltage condition is present on Input Channel A. To correct this error, change the signal Bias or Attenuation on the Input menu.

Error 110:
Ch B Overvoltage.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when an overvoltage condition is present on Input Channel B. To correct this error, change the signal Bias or Attenuation on the Input menu.

Error 111:
Power-down before removing pods.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when an input pod is removed while the HP 5373A is powered on. The instrument should be powered down before input pods are removed.

Error 130:
Only graphs can be plotted.

Type: Momentary Warning *Associated With:* Plot

This message occurs when an attempt is made to plot a display other than Graphics. Plots are only allowed of Graphic displays. To get a hardcopy of any other menu, use the PRINT function, which outputs the current display to an attached printer.

Error 131:
Plot/meas data conflict.

Type: Momentary Warning *Associated With:* Plot

This message occurs when the PLOT key is pressed while Graphics are available, but the output source chosen on the System menu is Measurement instead of Display. In this case, the HP 5373A has been configured to output measurement results instead of display data. Set the output source to Display before plotting.

Error 140:
Register protected.

Type: Momentary Warning *Associated With:* Save/Recall

This message occurs when an attempt was made to save the current instrument configuration to a register that is protected. Save to another register, or unprotect the desired register on the Instrument State Menu.

Error 141:
Register not saved yet.

Type: Momentary Warning *Associated With:* Save/Recall

This message occurs when an attempt has been made to recall a saved instrument configuration from a register that has not yet been saved.

Error 142: Register out of range.

Type: Momentary Warning *Associated With:* Save/Recall

This message occurs when a register number outside the valid range has been entered. Valid register numbers are 0-9 for RECALL, and 1-9 for SAVE or ERASE.

Error 150: Parameter conflict.

Type: Static Error *Associated With:* HP-IB

This message occurs when an HP-IB command was sent which conflicts with the current instrument configuration.

Error 160: Out of sensitivity cal.

Type: Static Failure *Associated With:* Hardware Error

This message occurs when the HP 5373A has lost its battery-stored memory and the sensitivity calibration factors have been lost. When this occurs, the factors are set to defaults, and the HP 5373A needs calibration.

Events occurred which were not timed.

Type: Momentary Status *Associated With:* Measurement

This message occurs when the measurement acquisition process is not able to timestamp every event sample individually. All measurement results displayed are still valid in this case; no user-intervention is required.

Exponent entry disallowed.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs for one of two reasons: an attempt was made to enter an integer parameter using exponent format, but the menu field is too small to adequately handle that format (not enough space to specify digits plus the "E+00" notation), or the EXP key was pressed without having entered any digits in the current numeric entry sequence.

Exponent disallowed due to mantissa.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs if there is not enough space left in the numeric entry field to show "E+00" when the EXP key is pressed. In this case, the exponent entry mode is valid for the parameter in question, but there is not enough space left in the field because too many digits have already been entered.

There must be at least four character spaces available to show "E+xx" in the field. The BACKSPACE key may be used to clear enough character spaces to allow exponent entry.

Gate open.

Type: Static Status *Associated With:* Manual Totalize Measurement

This message appears while the manually controlled gate is open during a Manually armed Totalize measurement. When the gate is closed to complete the measurement, the message is erased and the measurement result is displayed.

Graphics not allowed for this meas.

Type: Momentary Status *Associated With:* Manual Totalize and Envelope Power Measurements

Graphics displays are not allowed when the HP 5373A is making Envelope Power, Amplitude Modulation, or Manually armed Totalize measurements.

HP 5373A Graphics command no longer used.

Type: Momentary Warning *Associated With:* HP-IB *Error Number:* 120

This message occurs if an unsupported HP 5371A Graphics command is sent via HP-IB. See Appendix F for details on how HP 5371A commands may be translated into equivalent HP 5373A commands.

Inhibit usage may distort results.

Type: Momentary Status *Associated With:* Inhibit function

When the Inhibit function is activated, the requested number of measurements will be made, but not all results may be valid. The Numeric screen and Graphics displays indicate which measurements have been inhibited during the measurement sequence.

Input line truncated to first 80 chars.

Type: Momentary Status *Associated With:* HP-IB

This message occurs when an HP-IB string of more than 80 characters is entered from the controller. The parser truncates the string to the first 80 characters, and continues processing.

Input parameters may have changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when parameters on the Input menu have been changed to resolve conflict with another parameter that has been entered. In this case, no Arming mode or Arming parameter changes have occurred.

Interval sample value changed to 131 μs.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when Fast Measurement mode is selected, causing an out-of-limits condition for the interval sample value. 131 μsec is the maximum interval sample value allowed in Fast Measurement mode.

Measurement Aborted.

Type: Momentary Status *Associated With:* Measurement

This message occurs when the "ABORT" HP-IB command is received or the Abort (MANUAL ARM) key is pressed while a measurement is in progress, and enough samples have been taken to give at least one valid measurement result. The Abort command is valid only when the HP 5373A is in Single Sample mode.

Measurement Inhibited.

Type: Static Status *Associated With:* Measurement

This message occurs when Inhibit mode is activated, and at least one measurement in the last acquisition sequence was inhibited.

Measurement terminated, no data.

Type: Momentary Status *Associated With:* Measurement

This message occurs when the "ABORT" HP-IB command is received or the Abort (MANUAL ARM) key is pressed while a measurement is in progress, and not enough samples have been taken to give at least one valid measurement result. The Abort command is valid only when the HP 5373A is in Single mode.

No digits specified, entry aborted.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when the ENTER key is pressed, without having specified a numeric value containing any digits.

Non-numeric key ignored.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs in numeric entry when pressing any non-numeric key before pressing ENTER or LAST VALUE. Non-numeric keys are keys other than 0 to 9, . (decimal point), EXP, +/-, or BACKSPACE.

Not in Talk-only.

Type: Momentary Status *Associated With:* Print/Plot Graph keys

This message occurs when an attempt is made to print a screen or plot a graph without first setting the HP 5373A to Talk-Only mode on the System menu screen.

Number must be positive.

Type: Momentary Status *Associated With:* Numeric Entry

This mesage occurs when an attempt is made to change an enterable parameter to a negative value, and that parameter is only allowed to be positive. Examples are: Measurement Size, or Arming on event or time values.

Numeric entry aborted.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when the LAST VALUE key is pressed. The parameter that was being entered is restored to its previous value.

Out of Range: see Meas mode on System menu.

Type: Momentary Status *Associated With:* Parameter Conflict

This message occurs when a parameter value is entered which cnflicts with the limitations of Fast Measurement mode. However, the parameter value may be within the valid range for Normal Measurement mode (Measurement mode is selectable on the System menu).

Plot/Print aborted.

Type: Momentary Status *Associated With:* Plot/Print

The current plot or print output action in progress has been canceled (at user request).

Pre-trigger precedes data.

Type: Momentary Status *Associated With:* Measurement

This message appears when a block of data has been captured due to a Pre-trigger but the point where the Pre-trigger occurred precedes the block of data shown.

Response timeout occurred.

Type: Momentary Warning *Associated With:* System Operation *Error Number:* -303

This message occurs when the Response Timeout feature is enabled and a measurement is in progress but has not completed within the specified time period. The instrument will proceed with the measurement acquisition to completion if possible.

Result format must be ASCII, see System menu.

Type: Momentary Status *Associated With:* HP-IB

This message occurs when an attempt is made to send data out on the HP-IB while in Talk Only, and the output format is binary or floating point. The output format should be changed to ASCII on the System menu.

Sending output to plotter...

Type: Static Status *Associated With:* Plot

This message occurs while the current Graphics screen display is being output to the attached plotter.

Sending output to printer...

Type: Static Status *Associated With:* Print

This message occurs while the current screen display is being output to the attached printer.

Source channel has changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when the Measurement Source channel has been changed to resolve conflict with another parameter that has been entered (such as changing the Measurement Function).

Source, input parameters changed.

Type: Momentary Status *Associated With:* Parameter Coupling

This message occurs when the Measurement Source channel and one or more Input menu parameters have been changed to resolve conflict with another parameter that has been entered (such as changing the Measurement Function).

Stop Arming precedes Start Arming.

Type: Momentary Status *Associated With:* Parameter Validation

This message appears when the Stop Arming value is less than the Start Arming value. The stop arm will thus occur before the start arm. If you are making ±Time Interval measurements, expect negative results.

Undefined key.

Type: Momentary Status *Associated With:* Key entries

This message occurs when an invalid or undefined key is pressed. An example is an undefined softkey.

Value out of range: set to limit.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when parameter values have been altered to resolve conflict with the selection of Fast Measurement mode.

Value out of range: set to maximum.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when the entered parameter value is above the maximum allowable value. The parameter is defaulted to that maximum value.

Value out of range: set to minimum.

Type: Momentary Status *Associated With:* Numeric Entry

This message occurs when the entered parameter value is below the minimum allowable value. The parameter is defaulted to that minimum value.

Waiting for arming...

Type: Static Status *Associated With:* Measurement Status

This message occurs when the HP 5373A is waiting for the specified arming event to occur, before making the first measurement.

Waiting for input signal...

Type: Static Status *Associated With:* Measurement Status

This message occurs when the HP 5373A has met the specified arming condition, and is waiting for measurements to begin (no input signal has been detected).

Waiting for Manual Arm...

Type: Static Status *Associated With:* Measurement Status

This message appears when a Totalize measurement with Manual arming is stared. The HP 5373A is waiting for the MANUAL ARM key to be pressed which will open the gate.

Waiting for Pre-trigger...

Type: Static Status *Associated With:* Measurement Status

This message occurs when the HP 5373A is waiting for the specified Pre-trigger condition to occur, before making a block of measurements.

Waiting for Start Arming...

Type: Static Status *Associated With:* Measurement Status

This message appears when the measurement in progress is waiting for the start arm to occur.

Waiting for Stop Arming...

Type: Static Status — *Associated With:* Measurement Status

This message appears when the measurement in progress is waiting for the stop arm to occur.

WARNING: Both frequencies out of auto-trigger range.

Type: Momentary Warning — *Associated With:* System Operation — *Error Number:* 182

This message occurs when the instrument is in Auto-trigger mode, and the input signals on both Channel A and B are outside the Auto-trigger frequency range. One input signal is below 1 kHz and one signal is above 200 MHz.

WARNING: Frequency too high for auto-trigger.

Type: Momentary Warning — *Associated With:* System Operation — *Error Number:* 180

This message occurs when the instrument is in Auto-trigger mode, and the input signal is above 200 MHz. The Auto-trigger frequency range is 1 kHz to 200 MHz.

WARNING: Frequency too low for auto-trigger.

Type: Momentary Warning — *Associated With:* System Operation — *Error Number:* 181

This message occurs when the instrument is in Auto-trigger mode, and the input signal is below 1 kHz. The Auto-trigger frequency range is 1 kHz to 200 MHz.

D

PROGRAMMING THE ARMING MODES

APPENDIX
D

PROGRAMMING THE ARMING MODES

INTRODUCTION

This appendix describes how to select an arming mode, then choose the appropriate commands and options. Four examples show representative program lines and screen displays for four different arming modes.

AVAILABLE ARMING MODES

Table D-1 lists the arming modes available for each measurement function. The arming modes are shown in the left-most column of the table, and the measurements are listed across the top. To use *Table D-1*:

1. Pick a measurement column at the top of the table.
2. Scan down that column until you come to a box with an entry.
3. Each box with an entry indicates an arming mode (on the left) that you can use to make the selected measurement.
4. Use the Key at the end of the table for an explanation of the entry.

For information about measurements and arming modes, refer to the operating manual.

Table D-1. Function and Arming Summary

ARMING MODE	MEASUREMENT FUNCTION												
	TIME INTERVAL OR HISTOGRAM TI		CONTINUOUS TIME INTERVAL OR HISTOGRAM CTI	± TIME INTERVAL OR HISTOGRAM ±TI		PRF, FREQUENCY, PRI, PERIOD		TOTALIZE		PULSE WIDTH, OFFTIME, RISE TIME, FALL TIME, DUTY CYCLE	PHASE	ENVELOPE POWER, AMPLITUDE MODULATION	PHASE DEVIATION, TIME DEVIATION, FREQUENCY DEVIATION
	A	A → B	A	A	A → B	A	DUAL[1]	A	DUAL[1]	A	A rel B	A	A
	B	B → A	B	B	B → A	B	RATIO[2]	B	RATIO[2]		B rel A	B	B
						C	SUM[3]		SUM[3]				
							DIFF[4]		DIFF[4]				
AUTOMATIC													
AUTOMATIC	C*	C*	C*		C*	C*	C*			C*	C*	N*	C*
HOLDOFF													
EDGE HOLDOFF	C	C	C		C	C					C		C
TIME HOLDOFF	C	C	C			C							
EVENT HOLDOFF	C	C	C			C							
SAMPLING													
INTERVAL SAMPLING	C	C	C		C	C	C	C*	C*		C		C
TIME SAMPLING						N							
CYCLE SAMPLING						C							
EDGE SAMPLING						C	C	C	C				
PARITY SAMPLING					C								
REPET EDGE SAMPLING	C	C	C		C								
REPET EDGE-PARITY SAMPLING					C								
RANDOM SAMPLING	C	C			C								
HOLDOFF/SAMPLING													
EDGE/INTERVAL	C	C	C		C	C	C	C	C		C		C
EDGE/TIME						N							
EDGE/EDGE						C		C	C				
EDGE/CYCLE						C							
EDGE/EVENT				N	N	N							
EDGE/PARITY					C								
EDGE/RANDOM	C	C			C								
TIME/INTERVAL						C		C					
TIME/TIME				N	N	N							
EVENT/INTERVAL						C							
EVENT/EVENT				N*	N	N							
EXTERNALLY GATED						C		C	C				
MANUAL								N	N				

Symbol C or N indicates that a measurement can be made using the corresponding combination of Function, Channel, and Arming selections.

C = Continuous Arming, (Block/Sample Arming)

N = Non-Continuous arming, (Start/Stop Arming), setups are limited to M blocks of 1 measurement.

1. DUAL. Simultaneous Dual-channel, (2 results). Frequency and Period options are: A&B, A&C, B&C. Totalize option is: A&B.
2. RATIO. Frequency and Period ratio options are: A/B, A/C, B/A, B/C, C/A, C/B. Totalize ratio options are: A/B, B/A.
3. SUM. Frequency and Period sum options are: A+B, A+C, B+C. Totalize sum option is: A+B.
4. DIFFERENCE. Frequency and Period difference options are: A–B, A–C, B–A, B–C, C–A, C–B. Totalize difference options are: A–B, B–A.

* = Default Arming

ARMING CATEGORIES

Category	Continuous Arming Modes	Non-Continuous Arming Modes
Automatic	Block Holdoff is Automatic Sample Arm is Automatic	none
Holdoff Modes	Block Holdoff is User-defined Sample Arm is Automatic	none
Sampling Modes	Block Holdoff is Automatic Sample Arm is User-defined	Start Arm is Automatic Stop Arm is User-defined
Holdoff/Sampling Modes	Block Holdoff is User-defined Sample Arm is User-defined	Start Arm is User-defined Stop Arm is User-defined

ARMING MODE PROGRAMMING COMMANDS

Table D-2 summarizes the programming commands for the default settings of each arming mode. The arming modes are in the left-most column of the table, and the programming commands are listed across the top.

Here is an example of how to use *Table D-2* to program an arming mode:

1. Specify the measurement and source channel. A frequency measurement on Channel B is used here.

   OUTPUT 703; "MEAS; FUNC, FREQ; SOUR, B"

2. Find a supported arming mode for frequency from *Table D-1*. Event/Event is used here.

3. Specify the arming mode from *Table D-2*.

   OUTPUT 703; "ARM, EVEV"

4. Specify the start arm sublevel commands by adding to the previous program line.

   OUTPUT 703; "ARM, EVEV; STAR; SLOP, POS; CHAN, A; DEL, 5; DCH, B"

   This program line now sets the measurement to begin after a positive edge on Channel A, followed by 5 events on Channel B.

5. Specify the stop arm sublevel commands.

   OUTPUT 703; "SAMP; DEL, 15; DCH, B"

   This program line sets the measurement to end after 15 events on Channel B. Note that the start and stop delay events are both referenced to the positive edge on Channel A specified in Step 4. The measurement result will be the frequency of 10 events measured on Channel B.

Table D-2. Arming Mode Programming Commands

Arming Mode	Arming Sublevel Command	SLOPe	CHANnel	DELay	Delay CHannel
1. Automatic AUT	——	——	——	——	——
2. Edge Holdoff EDH	STAR	(POS), NEG	(A), B, X	——	——
3. Time Holdoff THOL	STAR	(POS), NEG	(A), B, X	(2 ns)	——
4. Event Holdoff EVH	STAR	(POS), NEG	(A), B, X	(1 event)	(A), B
5. Interval Sampling ISAM	SAMP	——	——	(10 μs)	——
6. Time Sampling TSAM	SAMP	——	——	(1 s)	——
7. Cycle Sampling CSAM	SAMP	——	——	(16 cycles)	(A), B, O*
8. Edge Sampling ESAM	SAMP	(POS), NEG	(A), B, X	——	——
9. Parity Sampling PSAM	——	——	——	——	——
10. Repetitive Edge REDG	STAR	(POS), NEG	(A), B, X	——	——
11. Repetitive Edge/ Parity RPAR	STAR	(POS), NEG	(A), B, X	——	——
12. Random Sampling RSAM	——	——	——	——	——
13. Edge/Interval EDIN	STAR	(POS), NEG	(A), B, X	——	——
	SAMP	——	——	(10 μs)	——
14. Edge/Time EDT	STAR	(POS), NEG	(A), B, X	——	——
	SAMP	——	——	(1 s)	——
15. Edge/Edge EDED	STAR	(POS), NEG	(A), B, X	——	——
	SAMP	(POS), NEG	(A), B, X	——	——
16. Externally Gated EGAT	STAR	(POS), NEG	(A), B, X	——	——
17. Edge/Cycle EDCY	STAR	(POS), NEG	(A), B, X	——	——
	SAMP	——	——	(16 cycles)	(A), B, O*
18. Edge/Event EDEV	STAR	(POS), NEG	(A), B, X	——	——
	SAMP	——	——	(1 event)	(A), B

*O or OSC is an abbreviation for the internal timebase oscillator. The internal timebase cycles at a rate of 2 ns/cycle.

Table D-2. Arming Mode Programming Commands (Continued)

Arming Mode	Arming Sublevel Command	SLOPe	CHANnel	DELay	Delay CHannel
19. Edge/Parity EDP	STAR	(POS), NEG	(A), B, X	——	——
20. Edge/Random EDR	STAR	(POS), NEG	(A), B, X	——	——
21. Time/Interval TINT	STAR	(POS), NEG	(A), B, X	(2 ns)	——
	SAMP	——	——	(10 μs)	——
22. Time/Time TTIM	STAR	(POS), NEG	(A), B, X	(2 ns)	——
	SAMP	——	——	(1 s)	——
23. Event/Interval EVIN	STAR	(POS), NEG	(A), B, X	(1 event)	(A), B
	SAMP	——	——	(10 μs)	——
24. Event/Event EVEV	STAR	(POS), NEG	(A), B, X	(1 event)	(A), B
	SAMP	——	——	(10 events)	(A), B
25. Manual MAN	——	——	——	——	——

Conventions used in Table D-2

Table D-2 shows the commands to program the arming modes. This list of items explains the terms used in the table.

1. The programming shortform for each of the arming modes is listed in the left column.

2. In the column called, "Arming Sublevel Command," STAR is the shortform for START, and SAMP is the shortform for SAMPLE.

3. The programming commands are listed across the top of the table in the same order as the menu fields programmed by these commands are displayed on the Function menu.

4. Lines in the boxes indicate that the command is not recognized for that particular arming mode/sublevel command combination.

5. All choices are listed. Choices in parentheses ( ) are default choices. These default choices are set by the HP-IB PRESet command or by pressing the Preset key on the front panel, when in local.

6. X = External Arm input.

Example Program Arming Excerpts

Four examples are included here to demonstrate the use of *Table D-2* to program the arming modes. Compare the arming commands used in these examples with the command sequence as listed in the table.

NOTE

The HP 5373A will accept a maximum program line length of 80 characters, including delimiters. The programming examples below are shown with shorter line lengths to make them easier to describe.

Automatic Arming Mode —

OUTPUT 703; "MEAS; FUNC, FREQ; SOUR, A; ARM, AUT"

This program line selects a Channel A frequency measurement with Automatic arming. *Figure D-1* shows the HP 5373A as it would be programmed.

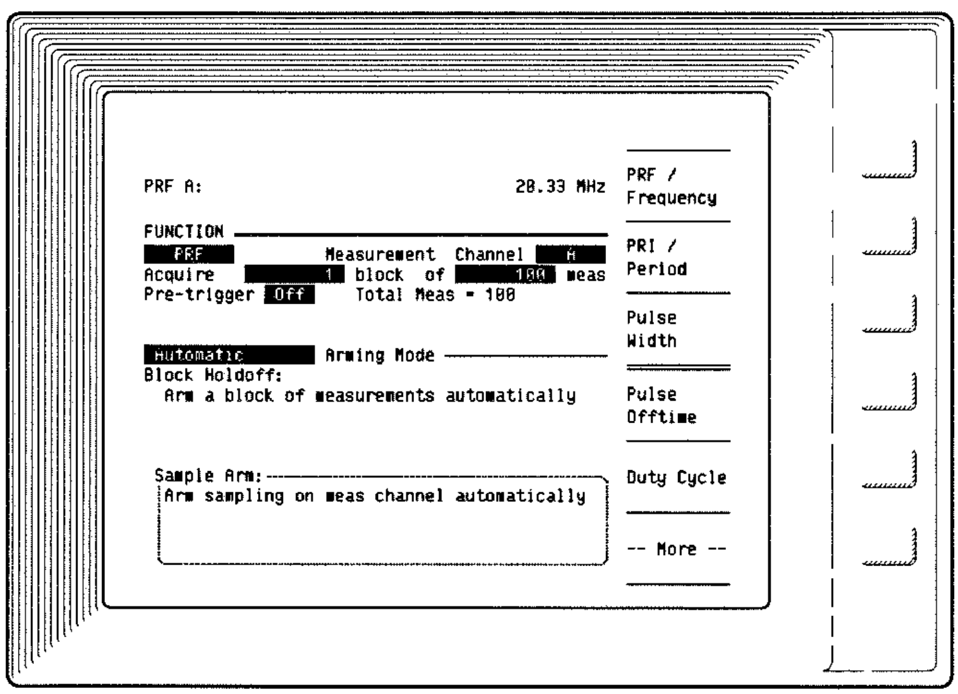


Figure D-1. Automatic Arming Mode

Event Holdoff Arming Mode —

OUTPUT 703; "MEAS; FUNC, TINT; SOUR, (A>B); ARM, EVH"

OUTPUT 703; "STAR; SLOP, NEG; CHAN, A; DEL, 25; DCH, B"

The first program line selects a Time Interval measurement, Channel A to Channel B, with Event Holdoff arming.

The second line sets the measurement to begin after a negative edge on Channel A, followed by 25 events on Channel B. *Figure D-2* shows the HP 5373A as it would be programmed.

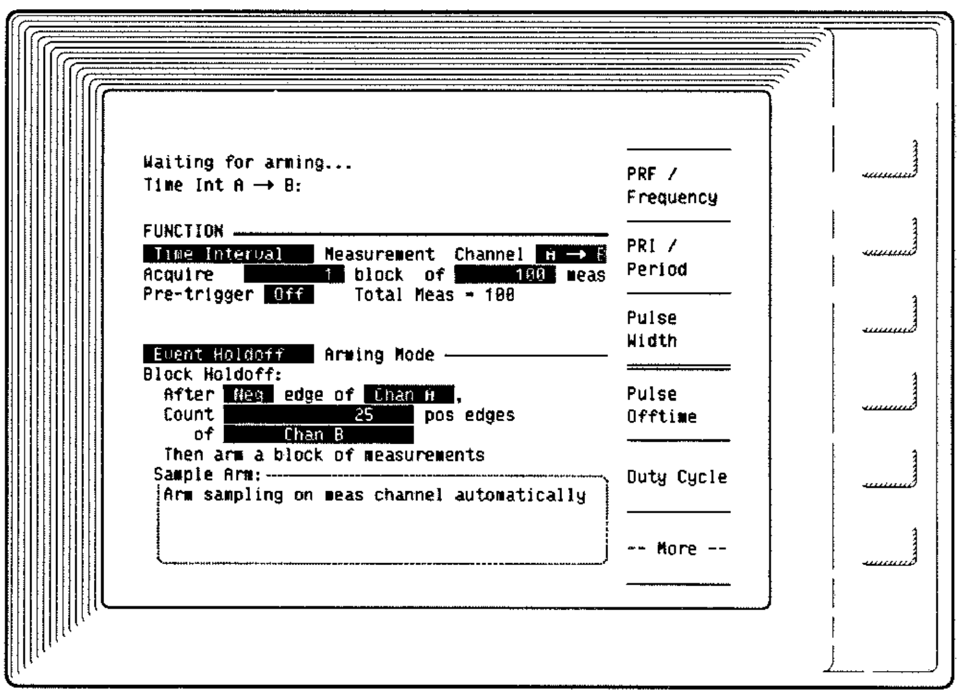


Figure D-2. Event Holdoff Arming Mode

Interval Sampling Arming Mode —

OUTPUT 703; "MEAS; FUNC, FREQ; SOUR, A; ARM, ISAM; SAMP; DEL, 100E-6"

This program line selects a Channel A Frequency measurement with Interval Sampling arming. The interval is set to 100 microseconds. *Figure D-3* shows the HP 5373A as it would be programmed.

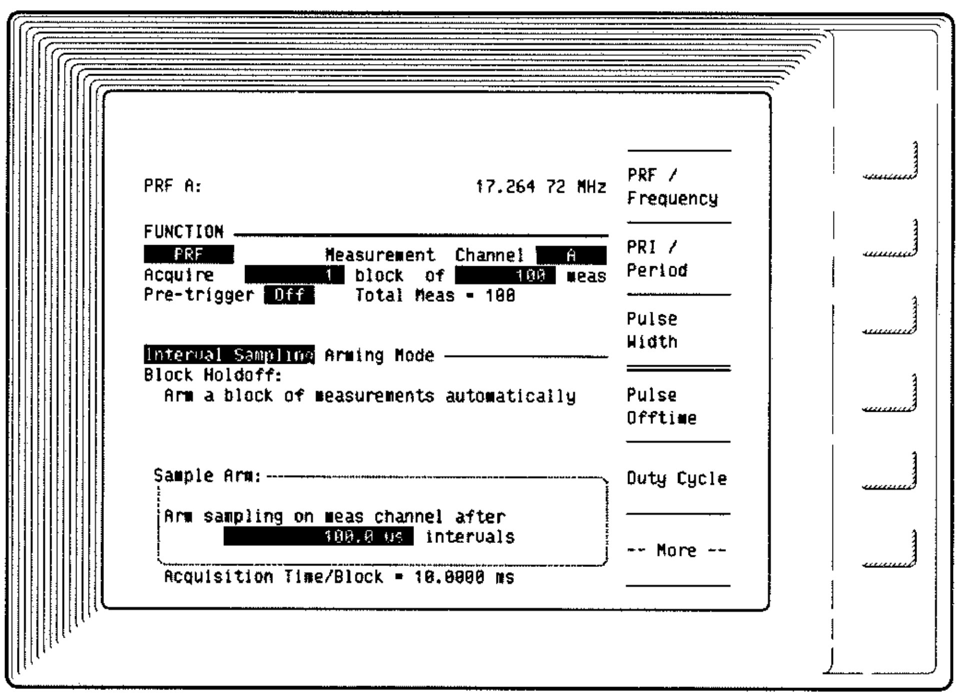


Figure D-3. Interval Sampling Arming Mode

Event/Interval Arming Mode —

OUTPUT 703; "MEAS; FUNC, FREQ; SOUR, A; BLOC, 1; MSIZ, 164"

OUTPUT 703; "ARM, EVIN; STAR; SLOP, NEG; CHAN, B; DEL, 40; DCH, A"

OUTPUT 703; "SAMP; DEL, 10E-3"

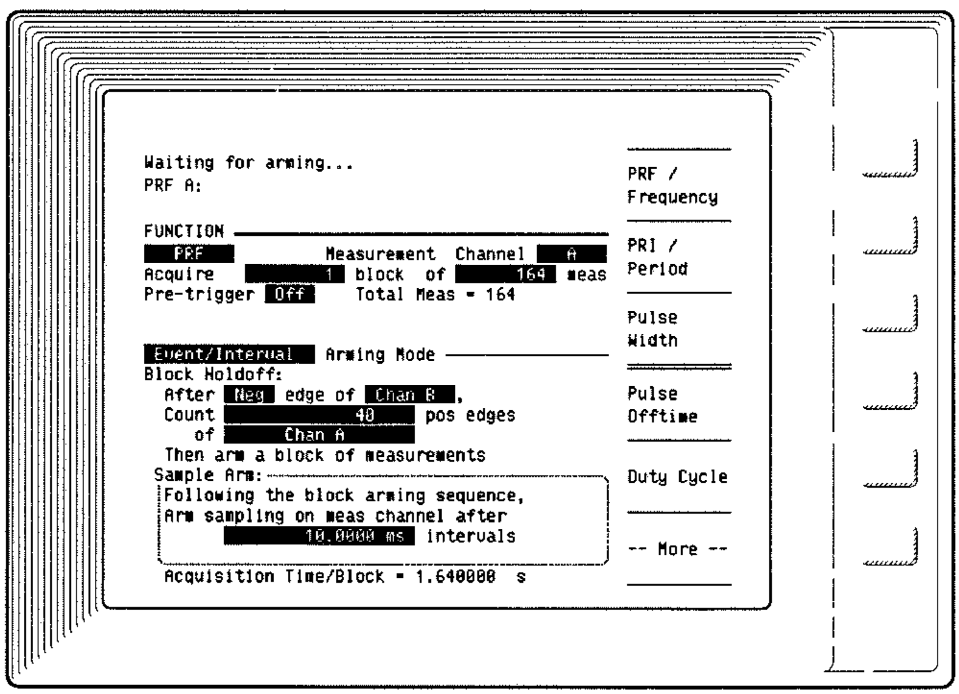
The first program line selects a Frequency measurement on Channel A. One block of 164 measurements will be collected.

The second line selects the Event/Interval arming mode. The measurement begins after a negative edge on Channel B, followed by 40 events on Channel A.

The third line sets measurement samples to be taken on the trigger event after every 10 milliseconds. *Figure D-4* shows the HP 5373A as it would be programmed.

For a detailed explanation of all the arming modes, refer to the Operating Manual.

Figure D-4. Event/Interval Arming Mode



E PROGRAM EXAMPLES

APPENDIX

E

PROGRAM EXAMPLES

INTRODUCTION

Refer to the sample programs for guidance in programming the HP 5373A. Sample programs in this appendix include:

- Abort
- Status Byte
- Print/Plot
- Teach/Learn
- Binary
- ASCII
- Floating point

NOTE

All the program examples in this Appendix require that an HP 54002A 50Ω Input Pod be installed as the HP 5373A CHANNEL A Input.

ABORT

This example program uses the INTERVAL sampling mode of the HP 5373A to make a frequency measurement. The measurement can be completed in 3 ways. It can end when all of the samples (determined by the variable Block_size) are taken, when the user presses the Abort softkey on the computer, or when a measurement takes longer than specified in the variable Time_out. You specify which way to end a measurement.

The program consists of a main program and 3 subprograms. In the main program, variables are assigned to set up the HP 5373A. Block_size indicates the number of measurements you want taken, if an abort does not occur. Samp_time defines the sample interval. Time_out controls how long the program waits before issuing the ABORT command. As the example program is listed, the HP 5373A aborts a measurement three seconds after a measurement starts. For this setup, about 750 measurements are taken.

The subprogram Transfer_data sets up the HP 5373A to make the measurement. It also defines a key label that allows the user to abort a measurement at any time by pressing the soft key on the computer display. Lines 900 – 950 determine how many bytes the HP 5373A transfers, and initiates the transfer.

Example Program

```
10    !Program name "ABORT". Requires BASIC 5.0 or later
20    !For earlier versions of BASIC, the SUB Sort_data must be changed
30    !to not use the MAT command. Use a FOR/NEXT loop to get the data into
40    !Freq(*) and Seconds(*).
50    !Expanded data is turned on for this example. This means the HP 5373A
60    !will send over Frequency AND Time information.Data_pts(*) is set up with
70    !2 dimensions to handle the data. The SUB Sort_data takes the data
80    !from the buffer and puts it in Freq(*) and Seconds(*).
90    !
100   !
110   COM /Measurements/ INTEGER Block_size,Num_meas,REAL Samp_time,Time_out
120   COM /Instruments/ @Hp5373a,@Counter
130   DIM Data_pts(1:1000,1:2),Freq(1:1000),Seconds(1:1000)
140   CLEAR SCREEN
150   GRAPHICS OFF
160   Block_size=1000     !1000 measurements
170   Samp_time=.004      !Sample interval in seconds
180   Time_out=3.0        !Time_out in seconds, 3 seconds will abort measurement
190    !
200    !
210    !This demo can stop in one of 3 different ways.
220    !1) The measurement can be completed normally. That is, a total of
230    !   Block_size measurements are made.
240    !2) The measurement can end because it takes longer than the Time_out.
250    !   For this to happen, Block_size*Samp_time > Time_out.
260    !   For a Block_size of 1000 and a Samp_time of .004, Time_out should
270    !   be less than 1000*.004 or 4 seconds.
280    !3) The user hits the ABORT softkey on the computer.
290    !
300    !
310   CALL Transfer_data(Data_pts(*),Freq(*),Seconds(*))
320   CALL Sort_data(Data_pts(*),Freq(*),Seconds(*))
330   CALL Plot_it(Data_pts(*),Freq(*),Seconds(*))
340   END    !End of main program
350    !
360    !
370    !
380    !
390   SUB Transfer_data(Data_pts(*),Freq(*),Seconds(*))
400     COM /Measurements/ INTEGER Block_size,Num_meas,REAL Samp_time,Time_out
410     COM /Instruments/ @Hp5373a,@Counter
420     DIM Header$[10],Disp$[80]
```

```
430     FOR I=0 TO 23              !Blank all the softkeys
440        ON KEY I LABEL "" GOSUB Waiting
450     NEXT I
460 Transfer_data:! Sets up and transfers a block of data from the HP5373A.
470     DISP "Setting up measurement."
480     ASSIGN @Counter TO 703;FORMAT OFF !FORMAT OFF uses the internal data
490                                       !representation of the computer
500                                       !Necessary for Floating Point Output
510     ASSIGN @Hp5373a TO 703
520     CLEAR @Hp5373a
530     Clear_all=SPOLL(@Hp5373a) !Make sure nothing is pending in the HP 5373A
540     OUTPUT @Hp5373a;"PRES"    ! Preset the HP 5373A
550     OUTPUT @Hp5373a;"MEN,INF" ! Bring up information menu
560     !
570     OUTPUT @Hp5373a;"*SRE 16" ! Set Service Request Enable mask to pull SRQ
580                               ! on the Measurement Available bit of
590                               ! the HP 5373A
600     !
610     !
620     !
630     OUTPUT @Hp5373a;"SMODE,SING"   ! Put HP 5373A in single measurement mode
640     OUTPUT @Hp5373a;"INT;OUTP,FPO" ! Set data output to Floating Point
650     OUTPUT @Hp5373a;"MEAS;FUNC,FREQ"! Measure Frequency
660     OUTPUT @Hp5373a;"MSIZE,"&VAL$(Block_size)
670     OUTPUT @Hp5373a;"ARM,ISAM"     ! Interval Sampling
680     OUTPUT @Hp5373a;"SAMP;DEL,"&VAL$(Samp_time)  ! Set sample interval
690     OUTPUT @Hp5373a;"NUM;EXP ON"   ! Send over gate time with frequency
700     OUTPUT @Hp5373a;"RESTART"      ! Start measurement
710     !
720     !
730     !
740     ON KEY 8 LABEL "ABORT",15 GOTO Timed_out      !User initiated abort
750     DISP "Waiting for completion of measurement."
760     ON INTR 7 GOTO Get_data   !Set up interrupt to enable transfer of data
770     ENABLE INTR 7;2
780     ON DELAY Time_out GOTO Timed_out       !Timeout if data goes away
790     LOOP                                   !Loop here until measurement is
800     END LOOP                               !completed or aborted
810     !
820     !
830 Timed_out:  !Timeout or user aborted measurement
840             !Branch here if measurement is aborted
850             !and then continue with the transfer process
860     OUTPUT @Hp5373a;"ABORT"  !Abort the HP 5373A measurement
870     !
880     !
890 Get_data:    ! Start the transfer process here
900              ! Branch here if measurement completes normally
910     OFF INTR 7  !Disable interrupt
920     OFF DELAY   !Disable delay interrupt
930     !
940     LOOP
950     EXIT IF BIT(SPOLL(@Hp5373a),4) !Wait here until HP 5373A processes data
960     END LOOP
970     !
980     ENTER @Hp5373a USING "#,8A";Header$  !Get information about measurement
```

```
990     Number_bytes=VAL(Header$[3])          !Number of bytes to be transferred
1000                                          !Used by @Buff(*)
1010    Num_meas=Number_bytes/16              !Number of measurements
1020                                          !Used by Data_pts(*)
1030    REDIM Data_pts(1:Num_meas,1:2)        !Size Data according to Num_meas
1040    ENTER @Counter;Data_pts(*)            !Enter Data
1050    IF Num_meas<Block_size THEN
1060       Disp$="Measurement aborted. Number of measurements = "&VAL$(Num_meas)
1070    ELSE
1080       Disp$="Measurement complete. Number of measurements = "&VAL$(Num_meas
)
1090    END IF
1100    OUTPUT @Hp5373a;"LOC"          ! Put in local
1110    DISP Disp$
1120    SUBEXIT
1130    !
1140 Waiting:! If a non-assigned softkey is pressed just RETURN
1150    RETURN
1160    !
1170 SUBEND  ! Transfer_data
1180  !
1190  !
1200  !
1210  !
1220 SUB Plot_it(Data_pts(*),Freq(*),Seconds(*))
1230    COM /Measurements/ INTEGER Block_size,Num_meas,REAL Samp_time,Time_out
1240    INTEGER I
1250    GINIT
1260    GCLEAR
1270    GRAPHICS ON
1280    VIEWPORT 25,100*RATIO,30,98
1290    CSIZE 3,.5
1300    PEN 7
1310    LINE TYPE 3
1320    FRAME
1330    PEN 1
1340    LINE TYPE 1
1350    CLIP OFF
1360    Min_freq=MIN(Freq(*))
1370    Max_freq=MAX(Freq(*))
1380    IF Max_freq-Min_freq<1 THEN
1390       Max_freq=Max_freq+1.E-5*ABS(Max_freq)
1400       Min_freq=Min_freq-1.E-5*ABS(Min_freq)
1410    END IF
1420    WINDOW 0,1,0,1
1430    MOVE 0,1
1440    LORG 8
1450    LABEL VAL$(DROUND(Max_freq,10))&"-"
1460    MOVE 0,.5
1470    LABEL "Hz  "
1480    MOVE 0,0
1490    LABEL VAL$(DROUND(Min_freq,10))&"-"
1500    MOVE 0,0
1510    LORG 6
1520    LABEL "|"
1530    MOVE 0,0
```

```
1540    LORG 3
1550    LABEL 0
1560    MOVE .5,0
1570    LORG 6
1580    LABEL "seconds"
1590    MOVE 1,0
1600    LORG 9
1610    Total_time=SUM(Seconds)
1620    LABEL PROUND(Total_time,-12)
1630    CLIP ON
1640    PENUP
1650    PEN 3
1660    Prior_time=0
1670    WINDOW 0,Total_time,Min_freq,Max_freq
1680    FOR I=1 TO Num_meas
1690       PLOT Seconds(I)+Prior_time,Freq(I)
1700       Prior_time=Seconds(I)+Prior_time
1710    NEXT I
1720 SUBEND  ! Plot_it
1730  !
1740  !
1750  !
1760 SUB Sort_data(Data_pts(*),Freq(*),Seconds(*))
1770    COM /Measurements/ INTEGER Block_size,Num_meas,REAL Samp_time,Time_out
1780    MAT Freq= Data_pts(*,1)    !Get Freq(*) data from @Buff
1790    MAT Seconds= Data_pts(*,2) !Get Seconds(*) data from @Buff(*)
1800    REDIM Freq(1:Num_meas),Seconds(1:Num_meas)
1810 SUBEND  ! Sort_data
```

The subprogram Sort_data sorts the data from the buffer into two arrays: Freq(*) and Seconds(*). This makes it easier to examine and plot the data. BASIC 5.0 is required for this subprogram. If you have an earlier version of HP BASIC, use a FOR/NEXT loop to separate the data.

The subprogram Plot_it creates and labels a graph and then plots all the available points.

STATUS BYTE

HP 5373A Status Register Programming Example

The program listed on the following pages demonstrates how to use the Hardware Status Register to effectively control the interactions between the HP 5373A and a computer. Although only the Hardware Status Register is shown, the same techniques apply to the Event Status Register.

The HP 5373A has a Limit Testing feature that indicates when a measurement falls outside of user defined test limits. The Hardware Status Register can be used to enable SRQ and alert the computer that a measurement is out of limits. Refer to chapter 2 for a description of the Hardware and Event Status registers.

To use this program, do the following:

1. Install an an HP 54002A 50Ω Input Pod as the HP 5373A CHANNEL A Input.

2. Connect the 10 MHz Timebase output from the rear of the HP 5373A to the Channel A input.

3. Load and RUN the program.

4. As the program runs, the HP 5373A will make a block of measurements and check the measurements against the test limits. If the Timebase output is left connected, the measurements will always pass. The computer CRT will indicate the progress of the measurements.

5. Watch the Gate LED on the HP 5373A. When the Gate LED is on, the HP 5373A is making measurements. When the Gate LED is on, disconnect and then quickly re-connect the BNC cable. This action simulates a change in frequency that is out of limits.

6. The computer CRT shows the results from the SRQ.

The structure of the program is straightforward. The first part sets up the HP 5373A to make a frequency measurement using Interval sampling. The measurement is designed to last one second, which makes it easy to disconnect the BNC while the Gate LED is on.

A loop is set up to make measurements continuously, as long as the measurements are within the Low_limit and High_limit. If a measurement falls out of limit, an SRQ is generated, causing a branch to the label Srq.

The SRQ is then processed. It is cleared by a Serial Poll. The Status Byte is then examined to determine what caused the SRQ. From there, the Hardware Status Register is queried to determine what bit was set.Finally, the program reports on results of the process.

Example Program

```
10    !This program demonstrates the use of the Hardware Status Register.
20    !The program is set up to cause the HP 5373A to enable SRQ when
30    !a measurement block contains a frequency out of the user defined
40    !test limits. Techniques shown also apply to the Event Status Register.
50    !
60    !
70    !To run this demo, the 10 MHz Timebase Output from the rear of the
80    !HP 5373A must be connected to the Channel A input via a BNC cable.
90    !When the program is running, the computer CRT will show the number
100   !of completed measurements. To show how the Hardware Status Register
110   !can be used to pull SRQ, do the following:
120   ! 1) RUN the program.
130   ! 2) As measurements are made, the computer CRT updates the number of
140   !    measurement blocks completed.
150   ! 3) After several measurement blocks are made, do the following:
160   !     When the Gate LED is on, disconnect and then quickly re-connect
170   !     the BNC at Channel A. This will simulate a change in frequency
180   !     that will be out of limits, causing the HP 5373A to signal SRQ.
190   ! 4) The computer CRT will then show the results from the SRQ.
200   !
210   !
220 CLEAR SCREEN                         !Clear the CRT
230 Low_limit=9.999E+6                   !Low limit 9.999 MHz
240 High_limit=1.0001E+7                 !High limit 10.001 MHz
250                                      !The HP 5373A Timebase is 10.000 MHz
260 ASSIGN @Hp5373a TO 703               !Set up path for HP 5373A
270 CLEAR @Hp5373a                       !Clear any messages
280 OUTPUT @Hp5373a;"PRES"               !Preset the HP 5373A
290 OUTPUT @Hp5373a;"SMOD,SING"          !Put it in single measurement mode
300 OUTPUT @Hp5373a;"MEAS;FUNC,FREQ"     !Make a frequency measurement
310 OUTPUT @Hp5373a;"MSIZ,250"           !Set the measurement size to 250
320 OUTPUT @Hp5373a;"SOUR,A"             !Use Channel A
330 OUTPUT @Hp5373a;"ARM,ISAM"           !Interval Sampling
340 OUTPUT @Hp5373a;"SAMP;DEL,.004"      !4 msec sample intervals
350 OUTPUT @Hp5373a;"PROC;SOUR,A;LIM,ON"   !Turn test limits on
360 OUTPUT @Hp5373a;"LLIM";Low_limit       !Set the lower limit
370 OUTPUT @Hp5373a;"HLIM";High_limit      !Set the upper limit
380 OUTPUT @Hp5373a;"NUM;DISP,LIM"         !Enable the limit display
390 OUTPUT @Hp5373a;"MEN,NUM"              !Turn on the Numeric display
400                                        !The HP 5373A CRT will show the
410                                        !results of the limit test.
420   !
430 OUTPUT @Hp5373a;"*HSE 128"     !Set the Hardware Status Enable Register to
440                                !enable the measurement limit bit
450 OUTPUT @Hp5373a;"*SRE 1"       !Set the Service Request Enable register to
```

```
460                               !pull SRQ when Hardware Status Bit is set
470   !
480 OUTPUT @Hp5373a;"REST"        !Start a new measurement
490 ON INTR 7 GOTO Srq            !If SRQ then go to Srq and process it
500 ENABLE INTR 7;2               !Enable the interrupt
510 DISP "Measurement in progress."
520   !
530   !
540 LOOP                          !Stay in this loop as long as each block
550                               !of measurements passes the test limits.
560                               !An SRQ will exit this loop.
570   Test_one=SPOLL(@Hp5373a)
580   IF BIT(Test_one,4) THEN
590     N=N+1
600     DISP "Measurements in block";N;"passed. Measurements continuing..."
610     OUTPUT @Hp5373a;"REST"   !Start a new measurement
620   END IF
630 END LOOP
640   !
650   !
660 Srq:  ! An SRQ caused the program to branch here.
670 OFF INTR 7
680 PRINT "SRQ generated by HP 5373A."
690 PRINT
700 Test_bit=SPOLL(@Hp5373a)   !Serial Poll to clear SRQ
710 IF BIT(Test_bit,0) THEN    !Bit 0 goes high because of Hardware Status
720                            !Register activity.
730   DISP                     !Clear display line
740   PRINT "Hardware Status Register enabled SRQ."
750   PRINT
760   OUTPUT @Hp5373a;"*HSR?"  !Query the Hardware Status Register contents.
770   ENTER @Hp5373a;Hsr_value !Put the contents in Hsr_value
780   PRINT "Hardware Status Register returned a value of ";Hsr_value
790   IF BIT(Hsr_value,7) THEN !If bit 7 then measurement is out of limit.
800                             !Note: Other bits may also be set. Consult
810                             !the manual for details.
820     PRINT
830     PRINT "Hardware Status Register Bit 7 set (See manual for other Bits)."
840     PRINT
850     PRINT "Measurement out of limit!"
860   END IF
870 ELSE
880   DISP "Measurement error."
890 END IF
900 END
```

PRINT/PLOT

Example Program

```
10   ! This program demonstrates using a computer to access
20   ! the printer and plotter dump features of the HP 5373A
30   ! Modulation Domain Pulse Analyzer.
40   ! The program also demonstrates the use of the
50   ! Hardware Status Register and Service Request Enable Register
60   ! to determine when a print or plot is done; the HP 5373A
70   ! communicates this fact via SRQ.
80   !
90   ! Make a measurement, using the front panel of the instrument,
100  ! set the instrument's display to be as you want to see it, then
110  ! RUN this program.
120  !
130  !
140  Hpib=7                            ! SELECT CODE OF HP-IB INTERFACE
150  Printer=1                         ! HP-IB ADDRESS OF PRINTER
160  Hp5373a=3                         ! HP-IB ADDRESS OF HP5373A
170  Plotter=5                         ! HP-IB ADDRESS OF PLOTTER
180  ASSIGN @Hp5373a TO Hpib*100+Hp5373a
190  !
200  !
210  CLEAR Hpib
220  OUTPUT @Hp5373a;"SMOD,SING"       ! SET THE SAMPLING MODE TO SINGLE
230                                    ! (MAKE SURE NO MEASUREMENT IS MADE
240                                    !  BETWEEN PRINTING AND PLOTTING)
250  OUTPUT @Hp5373a;"*HSR?"           ! CLEAR THE HARDWARE STATUS REGISTER
260  ENTER @Hp5373a;Hsr
270  Temp=SPOLL(@Hp5373a)              ! MAKE SURE THE STATUS BYTE IS CLEARED
280  OUTPUT @Hp5373a;"*HSE,512;*SRE,1"  ! SET UP FOR SRQ ON HARD COPY COMPLETE
290  OUTPUT @Hp5373a;"INTE;PSOUR,DISP"  ! SET PRINTER DUMP MODE TO "DISPLAY"
300  !
310  !
320  !*******************************************************************
330  ! THE FOLLOWING TELLS THE HP 5373A TO DO A PRINTER DUMP, AND DOES THE
340  ! NECESSARY HP-IB ADDRESSING
350  SEND Hpib;UNL MTA LISTEN Hp5373a DATA "PRINT",CHR$(10) ! CHR$(10)=LINEFEED
360  SEND Hpib;UNL TALK Hp5373a LISTEN Printer DATA
370  !*******************************************************************
380  !
390  !
400  !
410  ! WAIT FOR AN SRQ (WHICH HAPPENS WHEN THE HP 5373A IS FINISHED PRINTING)
420  ON INTR Hpib GOTO Plot
430  ENABLE INTR Hpib;2
440 Wait1:GOTO Wait1
450  !
460  !
470 Plot:!
480  OUTPUT @Hp5373a;"*HSR?"          ! AGAIN, CLEAR OUT HSR
```

```
490  ENTER @Hp5373a;Hsr
500  Temp=SPOLL(@Hp5373a)              ! AGAIN, CLEAR THE STATUS BYTE
510  !
520  !
530  ! ONLY THE GRAPHICS DISPLAY CAN BE PLOTTED, SO THE FOLLOWING
540  ! LINE FORCES THE DISPLAY TO THIS MODE
550  OUTPUT @Hp5373a;"MENU GRAPHICS"
560  !
570  !
580  !*************************************************************************
590  ! THE FOLLOWING TELLS THE HP 5373A TO DO A PLOTTER DUMP, AND DOES THE
600  ! NECESSARY HP-IB ADDRESSING
610  SEND Hpib;UNL MTA LISTEN Hp5373a DATA "PLOT",CHR$(10) ! CHR$(10)=LINEFEED
620  SEND Hpib;UNL TALK Hp5373a LISTEN Plotter DATA
630  !*************************************************************************
640  !
650  !
660  ! WAIT FOR ANOTHER SRQ (WHEN THE PLOT IS DONE)
670  ON INTR Hpib GOTO Plot_done
680  ENABLE INTR Hpib;2
690 Wait2:GOTO Wait2
700  !
710  !
720 Plot_done:!
730  OUTPUT @Hp5373a;"*HSR?"             ! CLEAR OUT HSR, TO CLEAN UP
740  ENTER @Hp5373a;Hsr
750  Temp=SPOLL(@Hp5373a)                ! CLEAR STATUS BYTE, TO CLEAN UP
760  !
770  LOCAL @Hp5373a                      ! RETURN HP 5373A TO FRONT PANEL USE
780  !
790  END
```

Teach/Learn Programming Example

The following program demonstrates how to use the "SET?" query and the "SET" command to save and recall front panel setups, a technique known as the Teach/Learn feature. As the name implies, Teach/Learn is a two step process. First the Teach/Learn program stores a desired front panel setup in a string variable. Later, you select a stored string variable (the learned settings) to retrieve the corresponding front panel setup. You can create an unlimited number of such variables to manage diverse measurement regimes.

Example Program

```
10     !
20     !This program demonstrates the use of the "SET" command and the "SET?"
30     !query. The program is broken down into three sections,"Setup_query",
40     !"Store_setup", and "Retrieve_setup". "Setup_query" reads front panel
50     !setup information to Header$ and Setup_string$. "Store_setup" creates
60     !a binary file called "SETUP_FILE" on the selected disk drive,
70     !and stores Header$ and Setup_string$ in the file.
80     !"Retrieve_setup" retrieves the front panel setup information from
90     !"SETUP_FILE" and uses the "SET" command to program the HP 5373A's
100    !front panel setup.
110    CLEAR SCREEN                         !THIS IS AN HP BASIC 5.0 COMMAND ONLY
120    GCLEAR                               !CLEARS GRAPHICS ON CRT
130    ASSIGN @Hp5373a TO 703               !ASSIGNS @Hp5373a TO INSTRUMENT'S
140                                         !ADDRESS
150                                         !
160    DIM Header$[5]                       !DIMENSIONS Header$ TO 5
170    OUTPUT @Hp5373a;"PRES"               !PRESETS INSTRUMENT
180    OUTPUT @Hp5373a;"LOCAL"              !SET 5373A FOR LOCAL OPERATION
190    PRINT "MANUALLY SET THE HP 5373A TO THE DESIRED SETTINGS AND,"
200    PRINT "PRESS CONTINUE TO STORE THE HP 5373A FRONT PANEL SETUP....."
210    PAUSE
220    CLEAR SCREEN
230    !************************************************************************
240    ! Setup_query: Reads the HP 5373A's front panel setup information into
250    !              Header$ and Setup_string$.
260    !************************************************************************
270  Setup_query:  !
280                !
290      OUTPUT @Hp5373a;"SET?"                  !THE SET QUERY RETURNS A STRING
300                                              !CONTAINING THE 5373A'S FRONT
310                                              !PANEL SETUP INFORMATION
320                                              !
330      ENTER @Hp5373a USING "#,5A";Header$     !ENTER THE "Header" STRING WHICH
340                                              !IS THE FIRST 5 ALPHA CHARACTER
350                                              !OF THE SETUP STRING
360                                              !
370      String_length=(VAL(Header$[3]))         !DIMENSION String_length TO BE
380                                              !LENGTH OF SETUP STRING
390                                              !
```

```
400     ALLOCATE Setup_string$[String_length]  !DIMENSION Setup_string TO
410                                            !String_length
420                                            !
430     ENTER @Hp5373a USING "-K";Setup_string$!TRANSFER SETUP STRING
440                                            !FROM HP 5373A USING "-K"
450                                            !SO THAT TRANSFER DOES NOT
460                                            !TERMINATE ON RANDOM LINE FEED
470    !*********************************************************************
480    !Store_setup: Creates a binary file called "SETUP_FILE" on the selected
490    !             disk drive and stores Header$ and Setup_string$ in the
500    !             file.
510    !*********************************************************************
520 Store_setup:   !
530                !
540   CREATE BDAT "SETUP_FILE",1,(String_length+5+4+4) !CREATE A BDAT FILE WITH
550                                                    !ONE RECORD THAT IS LONG
560                                                    !ENOUGH TO HOLD Header$
570                                                    !AND Setup_string$
580                                                    !
590   ASSIGN @Path_1 TO "SETUP_FILE"          !OPEN AN I/O PATH TO "SETUP_FILE"
600                                           !
610   OUTPUT @Path_1;Header$,Setup_string$    !STORE Header$ and Setup_string$
620                                           !IN "SETUP_FILE"
630                                           !
640   !**********************************************************************
650   !Retrieve_setup: Retrieves the front panel setup information from
660   !                "SETUP_FILE" and uses the "SET" command to program
670   !                the 5373A's original front panel setup.
680   !**********************************************************************
690 Retrieve_setup:  !
700                  !
710   OUTPUT @Hp5373a;"LOCAL"                    !SET 5373A FOR LOCAL OPERATION
720   PRINT "MANUALLY CHANGE THE HP 5373A'S FRONT PANEL SETTINGS,"
730   PRINT "THEN PRESS CONTINUE AND CHECK FOR THE ORIGINAL FRONT"
740   PRINT "PANEL SETUP."
750   PAUSE
760   !
770   !
780   CLEAR SCREEN
790   ALLOCATE Retrieved_headr$[5]             !CREATE NEW STRING VARIABLE FOR
800                                            !THE HEADER INFORMATION
810                                            !
820   ALLOCATE Retrieved_setup$[String_length] !CREATE NEW STRING VARIABE FOR
830                                            !SETUP INFORMATION
840                                            !
850   ENTER @Path_1,1;Retrieved_headr$;Retrieved_setup$ !READ SETUP INFORMATION
860                                                     !FROM "SETUP_FILE"
870                                                     !
880   OUTPUT @Hp5373a;"SET"                    !THE "SET" COMMAND IS USED TO
890                                            !SEND SETUP INFORMATION TO THE
900                                            !HP 5373A
910                                            !
920   OUTPUT @Hp5373a USING "#,K";Retrieved_headr$,Retrieved_setup$
930   !SEND setup information back to the, HP 5373A
940   !
950   ASSIGN @Path_1 TO *                      !CLOSE I/O PATH
```

```
960   PURGE "SETUP_FILE"                        !PURGE "SETUP_FILE"
970                                             !SETUP_FILE CONTAINS FRONT PANEL
980                                             !SETUP INFORMATION, AND WOULD
990                                             !NOT NORMALLY BE PURGED.
1000                                            !THIS EXAMPLE PROGRAM PURGES
1010                                            !THE FILE SO IT IS NOT LEFT  ON
1020                                            !THE USER'S DISK DRIVE.
1030  END
```

BINARY DATA OUTPUT EXAMPLES

Below are three program listings that demonstrate the use of the binary output of the HP 5373A. The programs illustrate the setup, capture, transfer, and results of measurements made at the Channel A input of the HP 5373A. Refer to chapter 3 of this programming manual for detailed explanations of binary programming and data conversion techniques.

Binary Example #1

This program sets up the HP 5373A to take 4096 Time Interval measurements using Automatic arming. For this measurement the HP 5373A uses output format 4A (see Chapter 3, Binary Output, for details on format types). The time interval measurements, which are sent over the HP-IB as binary data, are converted in the controller, and plotted versus time. This example program contains three subroutines.

1. **Transfer_data** This subroutine sets up the HP 5373A to take 4096 time interval measurements. It then initiates the measurements, and uses the TRANSFER statement to read the measurement data from the HP 5373A.
2. **Convert_bin72** This subroutine converts the binary Time and Interpolator data into time stamps which are then used to calculate the time interval measurements.
3. **Plot_data** This subroutine plots the 4096 time interval measurements versus time.

To use this program:

1. Install an an HP 54002A 50Ω Input Pod as the HP 5373A CHANNEL A Input.
2. Connect a 10 kHz signal to Channel A.
3. LOAD and RUN the program.

Binary #1 Program Example

```
10    INTEGER Buff(0:8192,1:3) BUFFER,Block_size
20    !Dimension Buff to hold Time data for 4096 Time Interval measurements.
30    !Because this measurement uses Format 4A output mode there are 2N Time
40    !Stamps. This gives a total of 8192 Time Stamps each being 3 bytes long.
50    !
60    REAL Time_interval(1:4096),Time_stamp(1:8192)!Dimension Time_interval and
Time
70                                          !arrays to hold 4096 measurements
80    COM /Constants/ Format_bytes,Two_exp16,Two_exp32
```

```
90    Block_size=4096! Block size specifies the number of measurements to be
100                  ! taken. 4096 is the maximum number of measurements for
110                  ! Time Interval measurements using Automatic arming.
120                  !
130   Two_exp16=2^16  ! SET UP CONSTANTS
140   Two_exp32=2^32
150   Format_bytes=6
160   !This measurement uses Format 4A. The output therefore is represented as:
170   !Time (4 BYTES), INTERPOLATOR (2 BYTES)
180   !
190   Transfer_data(Block_size,Buff(*))
200   Convert_bin72(Block_size,Buff(*),Time_interval(*),Time_stamp(*))
210   Plot_data(Time_interval(*),Time_stamp(*))
220   END
230   SUB Transfer_data(INTEGER Block_size,Data_buff(*) BUFFER)
240 Transfer_data:! Sets up and transfers a block of data from the HP 5373A.
250     DISP "Setting up measurement."
260     Format_bytes=3
270     ASSIGN @Hp5373a TO 703         ! HP 5373A's ADDRESS
280     OUTPUT @Hp5373a;"PRES"         ! PRESET
290     OUTPUT @Hp5373a;"MEN,INF"      ! STATE MENU
300     OUTPUT @Hp5373a;"SMODE,SING"   ! SINGLE MEASUREMENT MODE
310     OUTPUT @Hp5373a;"INT;OUTP,BIN" ! BINARY OUTPUT
320     OUTPUT @Hp5373a;"MEAS;FUNC,TINT"! MEASURE TIME INTERVAL
330     OUTPUT @Hp5373a;"MEAS;SOUR,(A>B)"! CHANNELS A>B
340     OUTPUT @Hp5373a;"MSIZE,"&VAL$(Block_size)! MEASUREMENT SIZE = 4096
350     OUTPUT @Hp5373a;"MEAS;ARM,AUTO"  ! AUTOMATIC ARMING
360     OUTPUT @Hp5373a;"INP;MOD,COM"  !COMMON INPUT MODE
370     OUTPUT @Hp5373a;"REST"         ! START MEASUREMENT
380     ASSIGN @Buff TO BUFFER Data_buff(*)
390     DISP "Waiting for completion of measurement."
400     ENTER @Hp5373a USING "#,8A";Header$ ! Read Header$
410     Tot_byte_count=VAL(Header$[3]) !Total number of bytes to be transferred
420                                    !is represented by the last 6 digits of
430                                    !Header$
440     TRANSFER @Hp5373a TO @Buff;COUNT Tot_byte_count,WAIT
450     OUTPUT @Hp5373a;"LOC"          ! PUT IN LOCAL
460     OUTPUT @Hp5373a;"INT;OUTP,ASCII"   ! RETURN TO ASCII OUTPUT MODE
470     DISP
480   SUBEND ! Transfer_data
490   !
500   !
510   SUB Convert_bin72(INTEGER Block_size,Buff(*) BUFFER,REAL Time_interval(*),
Time_stamp(*))
520 Convert_bin72: !Converts binary 5373A data to Time_interval and Time arrays
530     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
540     INTEGER I,Format_words
550     REAL Start_time,Stop_time,Time_ovfl,Offset
560     Format_words=Format_bytes DIV 2 ! FORMAT IN 16 BIT WORDS
570     Offset=-4.E-10 ! This offset corrects for differences in the electronic
580                    ! paths lengths between measurement channels.
590     Time_ovfl=0
600     IF VAL(SYSTEM$("VERSION:CLOCK")) THEN ON CYCLE 1 GOSUB Disp_update ! Do
this only if CLOCK binary is loaded.
610     GOSUB Disp_update
620   ! REDIMENSION ARRAYS TO ALLOW FOR CONVENIENT INDEXING.
```

```
630   !
640     REDIM Buff(1:Block_size*2,1:Format_words),Time_interval(1:Block_size),Ti
me_stamp(1:Block_size*2)
650   ! Store Start and Stop time stamps in the Time_stamp array. Odd indices
660   ! correspond to Start times while even indices correspond to Stop times.
670   Time0=FNGet_4byte_val(1,1,Buff(*))
680   Time0_offset=Time0*2.E-9-Buff(1,3) MOD 32*1.E-10
690   Time_stamp(1)=0
700   FOR I=2 TO Block_size*2
710      Time1=FNGet_4byte_val(I,1,Buff(*))
720      IF Time1<Time0 THEN Time_ovfl=Time_ovfl+Two_exp32
730      Time_stamp(I)=((Time1+Time_ovfl)*2.E-9-Buff(I,3) MOD 32*1.E-10)-Time0_o
ffset
740      Time0=Time1
750   NEXT I
760   !
770   OFF CYCLE
780   !Calculate Time Interval measurements.
790   DISP "Calculating Time Interval Measurements . . ."
800   J=0
810   FOR N=1 TO Block_size*2 STEP 2
820     J=J+1
830     Time_interval(J)=(Time_stamp(N+1)+Offset)-Time_stamp(N)!
840     !Time_interval = (Stop time + Channel Offset) - Start time
850   NEXT N
860   SUBEXIT
870 Disp_update: DISP "Converting binary data.  ";I DIV 2;"of";Block_size;"compl
eted."
880     RETURN
890   SUBEND ! Convert_bin71
900   DEF FNGet_4byte_val(INTEGER Index1,Index2,Buff(*) BUFFER)
910 Get_4byte_val: ! Converts two BASIC INTEGER types into an unsigned 32 bit nu
mber.
920     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
930     RETURN (Buff(Index1,Index2)+(Buff(Index1,Index2)<0)*Two_exp16+(Buff(Inde
x1,Index2+1)<0))*Two_exp16+Buff(Index1,Index2+1)
940     !-------------------Analysis of the above Function---------------------
950     !This function converts the 32 bit unsigned binary Time data into 16
960     !bit signed binary data in order to simplify basic processing. To
970     !convert from unsigned to signed binary, the Function checks to see
980     !if either the upper or the lower 16 bits of the Time data is a
990     !negative number (if the most significant bit is a 1). If either of
1000    !the 16 bit numbers are negative, 2^16 is added to it in order to
1010    !convert it to its positive signed binary equivalent. The upper 16 bit
1020    !number is then multiplied by 2^16 and added to the lower 16 bit
1030    !number. The function can be rewritten as follows:
1040    !
1050    ! Return (Buff(Index1,Index2)*2^16) +
1060    ! + 2^32 if the msb of Buff(Index1,Index2) is a 1, (0 otherwise) +
1070    ! + Buff(Index1,Index2+1) +
1080    ! + 2^16 if the msb of Buff(Index1,Index2+1) is a 1, (0 otherwise)
1090    !
1100    !----------------------------------------------------------------------
1110  FNEND ! Get_4byte_val
1120  !
1130  !
```

```
1140  SUB Plot_data(Time_interval(*),Time_stamp(*))
1150 Plot_data:! Plots Time Interval vs. Time
1160    INTEGER I
1170    OUTPUT KBD;CHR$(255)&"K";! Clear ALPHA screen
1180    GINIT
1190    GCLEAR
1200    GRAPHICS ON
1210    VIEWPORT 15,90*RATIO,20,75
1220    CSIZE 3,.5
1230    PEN 7
1240    LINE TYPE 3
1250    FRAME
1260    LINE TYPE 1
1270    CLIP OFF
1280    Min_ti=MIN(Time_interval(*))
1290    Max_ti=MAX(Time_interval(*))
1300    IF Max_ti=Min_ti THEN
1310      Max_ti=Max_ti+1.E-5*ABS(Max_ti)
1320      Min_ti=Min_ti-1.E-5*ABS(Min_ti)
1330    END IF
1340    WINDOW 0,1,0,1
1350    MOVE 0,1
1360    LORG 8
1370    LABEL VAL$(DROUND(Max_ti,6))&"-"
1380    MOVE 0,.5
1390    LABEL "Hz  "
1400    MOVE 0,0
1410    LABEL VAL$(DROUND(Min_ti,6))&"-"
1420    MOVE 0,0
1430    LORG 6
1440    LABEL "|"
1450    MOVE 0,0
1460    LORG 3
1470    LABEL 0
1480    MOVE .5,0
1490    LORG 6
1500    LABEL "seconds"
1510    MOVE 1,0
1520    LABEL "|"
1530    MOVE 1,0
1540    LORG 9
1550    LABEL PROUND(Time_stamp(SIZE(Time_stamp,1)),-12)
1560    CLIP ON
1570    PENUP
1580    PEN 3
1590    MOVE 0,0
1600    WINDOW 0,Time_stamp(SIZE(Time_stamp,1)),Min_ti,Max_ti
1610    PENUP
1620    PLOT Time_stamp(1),Time_interval(1)
1630    FOR I=2 TO SIZE(Time_stamp,1) DIV 2-1
1640      PLOT Time_stamp(I*2),Time_interval(I)
1650    NEXT I
1660  SUBEND
```

Binary Example #2

This program sets up the HP 5373A to take 10 Frequency measurements using Time\Interval arming. For this measurement the HP 5373A uses output format 3 (see Chapter 3, Binary Output, for details on format types). The frequency measurements are sent over the HP-IB as binary data. The example program then converts the binary data in the controller, and displays the measurements. This example program contains three subroutines:

1. **Transfer_data** This subroutine sets up the HP 5373A to take 10 Frequecy measurements. It then initiates the measurements, and uses the TRANSFER statement to read the measurement data from the HP 5373A.

2. **Convert_bin72** This subroutine converts the binary Time, Event, and Interpolator data into Time and Event Stamps which are then used to calculate the Frequency measurements.

3. **Disp_data** This subroutine displays the value of the Time Stamp of the Block Arm and the 10 Frequency readings.

To use this program:

1. Install an an HP 54002A 50Ω Input Pod as the HP 5373A CHANNEL A Input.

2. Connect a 10 MHz signal to Channel A.

3. LOAD and RUN the program.

Binary #2 Program Listing

```
10    INTEGER Buff(0:12,1:7) BUFFER,Block_size
20    !Dimension Buff to hold Event,Time and Interpolator data for 10
30    !measurements. Because this measurement uses Format 3 output mode there
40    !are N+1 Time and Event Stamps plus 1 extra Time Stamp for the Time Stamp
50    !of the Block Arm. This gives a total of 12 Time and Event Stamp sets,
60    !each set being stored in 7 bytes.
70    REAL Freq(1:10),Time_stamp(0:10) !Dimension Freq and Time arrays to hold
80                               !measurements
90    COM /Constants/ Format_bytes,Two_exp16,Two_exp32
100   Block_size=10! Block_size specifies the number of measurements. The
110                ! maximum number of Frequency measurements using
120                ! Time\Interval arming is 8000.
130                  !
140   Two_exp16=2^16  ! SET UP CONSTANTS
150   Two_exp32=2^32
160   Format_bytes=14
170   ! This measurement uses Format 3. The output is therefore represented as:
```

```
180   ! EVENTS (4 BYTES), NONUSABLE (4 BYTES), TIME (4 BYTES), INTERP (2 BYTES)
190   !
200   Transfer_data(Block_size,Buff(*))
210   Convert_bin72(Block_size,Buff(*),Ts_block_arm,Time_stamp(*),Freq(*))
220   Disp_data(Block_size,Ts_block_arm,Time_stamp(*),Freq(*))
230   END
240   SUB Transfer_data(INTEGER Block_size,Data_buff(*) BUFFER)
250 Transfer_data:! Sets up and transfers a block of data from the HP 5373A.
260     DISP "Setting up measurement."
270     Format_bytes=14    !Because this measurement uses Format 3, 14 bytes
280                        !are tranferred for each set of Time, Event and
290                        !Interpolator data.
300     ASSIGN @Hp5373a TO 703          !HP 5373A's ADDRESS
310     OUTPUT @Hp5373a;"PRES"          ! PRESET
320     OUTPUT @Hp5373a;"MEN,INF"       ! STATE MENU
330     OUTPUT @Hp5373a;"SMODE,SING"    ! SINGLE MEASUREMENT MODE
340     OUTPUT @Hp5373a;"INT;OUTP,BIN" ! BINARY OUTPUT
350     OUTPUT @Hp5373a;"MEAS;FUNC,FREQ"! MEASURE FREQUENCY
360     OUTPUT @Hp5373a;"MSIZE,"&VAL$(Block_size) !MEASUREMENT SIZE = 10
370     OUTPUT @Hp5373a;"MEAS;ARM,TINT"  ! TIME/INTERVAL ARMING
380     OUTPUT @Hp5373a;"MEAS;STAR;SLOP,NEG;CHAN,A;DEL,3E-6"
390     ! BLOCK HOLDOFF AFTER NEGATIVE EDGE OF CHANNEL 'A' THEN DELAY 3.0 us
400     OUTPUT @Hp5373a;"MEAS;SAMP;DEL, 1E-6"  ! 1 us SAMPLING
410     OUTPUT @Hp5373a;"REST"          ! START MEASUREMENT
420     ASSIGN @Buff TO BUFFER Data_buff(*)
430     DISP "Waiting for completion of measurement."
440     ENTER @Hp5373a USING "#,8A";Header$ ! Read Header$
450     Tot_byte_count=VAL(Header$[3]) !Total number of bytes to be transferred
460                                     !is represented by the last 6 digits of
470                                     !Header$
480     TRANSFER @Hp5373a TO @Buff;COUNT Tot_byte_count,WAIT
490     OUTPUT @Hp5373a;"LOC"              ! PUT IN LOCAL
500     OUTPUT @Hp5373a;"INT;OUTP,ASCII"   ! RETURN TO ASCII OUTPUT MODE
510     DISP
520   SUBEND ! Transfer_data
530   !
540   !
550   SUB Convert_bin72(INTEGER Block_size,Buff(*) BUFFER,REAL Ts_block_arm,Time
_stamp(*),Freq(*))
560 Convert_bin72:! Converts binary HP 5373A data to Frequency and Time arrays.
570     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
580     INTEGER I,Format_words
590     REAL Time0,Time1,Event0,Event1,Time_ovfl,Time0_offset,Offset
600     Format_words=Format_bytes DIV 2 ! FORMAT IN 16 BIT WORDS
610     Offset=6.00E-10 ! This offset corrects for the difference in
620                     ! electronic path lengths between the block
630                     ! arming channel and the measurement channel.
640     Time_ovfl=0
650   !
660   ! REDIMENSION ARRAYS TO ALLOW FOR CONVENIENT INDEXING.
670     REDIM Buff(0:Block_size+1,1:Format_words),Time_stamp(0:Block_size),Freq(
1:Block_size)
680   !
690   ! GET THE TIME STAMP OF THE BLOCK ARM, AND THE FIRST EVENT AND TIME
700   ! VALUES. THESE ARE ASSOCIATED WITH TIME=0.
710     Ts_block_arm=FNGet_4byte_val(0,5,Buff(*))
```

```
720     Ts_offset=(Ts_block_arm*2.E-9)-(Buff(0,7) MOD 32*1.E-10)
730     Event0=FNGet_4byte_val(1,1,Buff(*))
740     Time0=FNGet_4byte_val(1,5,Buff(*))
750     IF Time0<Ts_block_arm THEN Time_ovfl=Time_ovfl+Two_exp32!ROLLOVER CHECK
760     Ts_block_arm=(Time0+Time_ovfl)*2.E-9-Buff(1,7) MOD 32*1.E-10-Ts_offset-O
ffset
770     Time0_offset=Time0*2.E-9-Buff(1,7) MOD 32*1.E-10
780   !
790     Time_stamp(0)=0
800     FOR I=2 TO Block_size+1
810       Event1=FNGet_4byte_val(I,1,Buff(*))
820       Time1=FNGet_4byte_val(I,5,Buff(*))
830       IF Event1<Event0 THEN Event1=Event1+Two_exp32      ! ROLLOVER CHECK
840       IF Time1<Time0 THEN Time_ovfl=Time_ovfl+Two_exp32  !     "
850       Time_stamp(I-1)=(Time1+Time_ovfl)*2.E-9-Buff(I,7) MOD 32*1.E-10-Time0_
offset
860       Freq(I-1)=(Event1-Event0)/(Time_stamp(I-1)-Time_stamp(I-2))
870       Event0=Event1
880       Time0=Time1
890     NEXT I
900     SUBEND
910   DEF FNGet_4byte_val(INTEGER Index1,Index2,Buff(*) BUFFER)
920 Get_4byte_val: !
930     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
940     RETURN (Buff(Index1,Index2)+(Buff(Index1,Index2)<0)*Two_exp16+(Buff(Inde
x1,Index2+1)<0))*Two_exp16+Buff(Index1,Index2+1)
950     !------------------Analysis of the above function---------------------
960     !This function converts the 32 bit unsigned binary Time and Event data
970     !into 16 bit signed binary data in order to simplify BASIC processing.
980     !To convert from unsigned to signed binary, the function checks to see
990     !if either the upper or the lower 16 bits of the Time or Event data is
1000    !a negative number (if the most significant bit is a 1). If either of
1010    !the 16 bit numbers are negative, 2^16 is added to it in order to
1020    !convert it to its positive signed binary equivalent. The upper 16 bit
1030    !number is then multiplied by 2^16 and added to the lower 16 bit
1040    !number. The function can be rewritten as follows:
1050    !
1060    ! Return (Buff(Index1,Index2)*2^16) +
1070    ! + 2^32 if the msb of Buff(Index1,Index2) is a 1, (0 otherwise) +
1080    ! + Buff(Index1,Index2+1) +
1090    ! + 2^16 if the msb of Buff(Index1,Index2+1) is a 1, (0 otherwise)
1100    !
1110    !------------------------------------------------------------------
1120  FNEND ! Get_4byte_val
1130  SUB Disp_data(INTEGER Block_size,REAL Ts_block_arm,Time_stamp(*),Freq(*))
1140 Disp_data: !DISPLAY FREQUENCY MEASUREMENTS
1150    PRINT CHR$(12);
1160    PRINT "Meas#              Measurement"
1170    PRINT
1180    PRINT "Timestamp";Ts_block_arm
1190    PRINT
1200    FOR X=1 TO Block_size
1210    PRINT X;
1220    PRINT TAB(20);Freq(X)
1230    NEXT X
1240    SUBEND
```

Binary Example #3

This binary output program is an example using the binary format output to make a single block of 4095 frequency measurements.

The following program makes frequency measurements and then displays a graph of the frequency vs. time results on the controller CRT (see *Figure E-1* for an example of the graphic display). The signal used for this measurement example was a 19 MHz input signal. A one millisecond gate time (sample interval) was selected.

For this measurement the HP 5373A uses output format 2A (see Chapter 3, Binary Output, for details on format types).

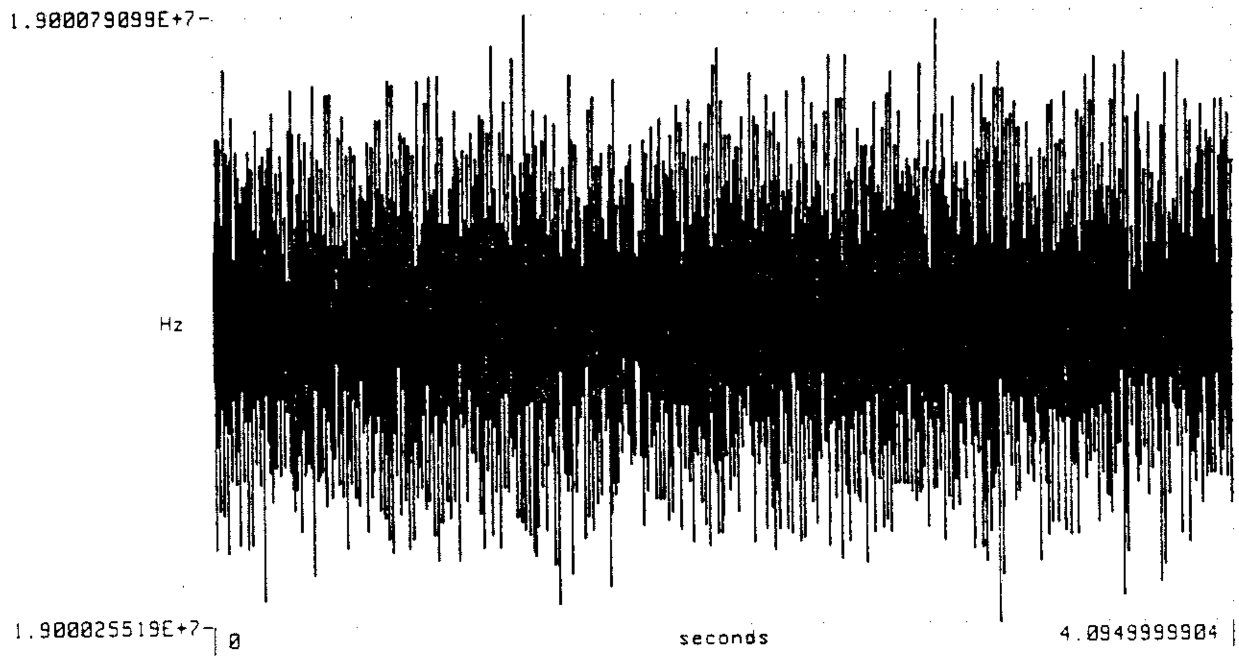


Figure E-1. Binary Example Graphic Display

Binary #3 Program Listing

```
10     INTEGER Buff(1:4096,1:5) BUFFER,Block_size
20     REAL Freq(1:4095),Time(1:4095)
30     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
40     Block_size=4095 ! As an input parameter, this specifies the number of
50                     ! MEASUREMENTS requested. As an output parameter, it
60                     ! Specifies the number of time stamps returned
70                     ! (usually MEASUREMENTS+1, unless system was aborted).
80                     !
90     Two_exp16=2^16  ! SET UP CONSTANTS
100    Two_exp32=2^32
110    Format_bytes=10 ! EVENTS (4 BYTES), TIME (4 BYTES), STATUS/INTERP
120                    !(2 BYTES)
130    Gate_time=0
140    PRINT CHR$(12)  !Clear alpha screen
150    INPUT "ENTER GATE TIME. (0 FOR AUTOMATIC ARMING)",Gate_time
160    Transfer_data(Gate_time,Block_size,Buff(*))
170    Convert_bin72(Block_size,Buff(*),Time(*),Freq(*))
180    Plot_data(Time(*),Freq(*))
190    END
200    SUB Transfer_data(Gate_time,INTEGER Block_size,Data_buff(*) BUFFER)
210  Transfer_data:! Sets up and transfers a block of data from the HP5373A.
220      DISP "Setting up measurement."
230      Format_bytes=10
240      ASSIGN @Hp5373a TO 703
250      OUTPUT @Hp5373a;"PRES"           ! PRESET
260      OUTPUT @Hp5373a;"REM"            ! REMOTE
270      OUTPUT @Hp5373a;"MEN,INF"        ! STATE MENU
280      OUTPUT @Hp5373a;"SMODE,SING"     ! SINGLE MEASUREMENT MODE
290      OUTPUT @Hp5373a;"INP;MOD,COM"    ! COMMON INPUTS
300      OUTPUT @Hp5373a;"INT;OUTP,BIN"   ! BINARY OUTPUT
310      OUTPUT @Hp5373a;"MEAS;FUNC,FREQ"! FREQUENCY
320      OUTPUT @Hp5373a;"MSIZE,"&VAL$(Block_size)
330      IF Gate_time THEN
340        OUTPUT @Hp5373a;"ARM,ISAM"     ! INTERVAL SAMPLING
350        OUTPUT @Hp5373a;"SAMP;DEL,"&VAL$(Gate_time)! DELAY GATE TIME
360      END IF
370      INPUT "Press <RETURN> to RE-START measurement.",A$
380      OUTPUT @Hp5373a;"REST"           ! START MEASUREMENT
390      ASSIGN @Buff TO BUFFER Data_buff(*)
400      DISP "Waiting for completion of measurement."
410      ENTER @Hp5373a USING "#,8A";Header$
420      Block_size=VAL(Header$[3]) DIV Format_bytes
430      Tot_byte_count=Format_bytes*Block_size
440      TRANSFER @Hp5373a TO @Buff;COUNT Tot_byte_count,WAIT
450      OUTPUT @Hp5373a;"LOC"            ! PUT IN LOCAL
460      OUTPUT @Hp5373a;"INT;OUTP ASC"   ! SET TO ASCII MODE TO ENABLE GRAPHICS
470      DISP
480    SUBEND ! Transfer_data
490    !
500    !
510    SUB Convert_bin72(INTEGER Block_size,Buff(*) BUFFER,REAL Time(*),Freq(*))
520  Convert_bin72:! Converts binary 5373A data to Frequency and Time arrays.
530      COM /Constants/ Format_bytes,Two_exp16,Two_exp32
```

```
540     INTEGER I,Format_words
550     REAL Time0,Time1,Event0,Event1,Time_ovfl,Time0_offset
560   !
570   ! Time0  is the previous 32-bit unsigned integer time count.
580   ! Event0 is the    "        "        "         "    event count.
590   ! Time1  is the current     "        "         "    time count.
600   ! Event1 is the    "        "        "         "    event count.
610   ! Format_words is the number of 16-bit words used for each measurement.
620   ! Time_ovfl increments by 2^32 every time the time counters overflow.
630   !           It essentially adds more bits to the time counter.
640   !           This measurement does not need to keep track of cumulative
650   !           event overflows, only relative ones.
660   ! Time0_offset is the value of the time counter at time t=0. It is
670   !           subtracted from every time value. This is calculated using
680   !           the interpolator data, which is the lower 5 bits of the
690   !           16-bit status/interp word. Note that the lower 5 bits of
700   !           the upper 8-bit byte is a duplication of the same data
710   !           for the FREQUENCY function.
720   !
730     Format_words=Format_bytes DIV 2 ! FORMAT IN 16 BIT WORDS
740     Time_ovfl=0
750   !
760   ! REDIMENSION ARRAYS TO ALLOW FOR CONVENIENT INDEXING.
770     REDIM Buff(0:Block_size-1,1:Format_words),Time(1:Block_size-1),Freq(1:Bl
ock_size-1)
780     GOSUB Disp_update
790   !
800   ! GET FIRST EVENT AND TIME VALUES. THESE ARE ASSOCIATED WITH TIME=0.
810     Event0=FNGet_4byte_val(0,1,Buff(*))
820     Time0=FNGet_4byte_val(0,3,Buff(*))
830     Time0_offset=Time0*2.E-9-Buff(0,5) MOD 32*1.E-10 ! Use low 5 bits of
840                                                      ! interpolator data.
850   ! GET FIRST MEASUREMENT. THIS IS OUTSIDE THE LOOP TO AVOID INDEXING
860   ! BEFORE THE FIRST MEASUREMENT.
870     Event1=FNGet_4byte_val(1,1,Buff(*))
880     Time1=FNGet_4byte_val(1,3,Buff(*))
890     IF Event1<Event0 THEN Event1=Event1+Two_exp32     ! Overflow occurred.
900     IF Time1<Time0 THEN Time_ovfl=Time_ovfl+Two_exp32 !    "        "
910     Time(1)=(Time1+Time_ovfl)*2.E-9-Buff(1,5) MOD 32*1.E-10-Time0_offset
920     Freq(1)=(Event1-Event0)/Time(1)
930     Event0=Event1
940     Time0=Time1
950     IF VAL(SYSTEM$("VERSION:CLOCK")) THEN ON CYCLE 1 GOSUB Disp_update
   ! Do this only if CLOCK binary is loaded.
960     FOR I=2 TO Block_size-1
970       Event1=FNGet_4byte_val(I,1,Buff(*))
980       Time1=FNGet_4byte_val(I,3,Buff(*))
990       IF Event1<Event0 THEN Event1=Event1+Two_exp32      ! Overflow
1000      IF Time1<Time0 THEN Time_ovfl=Time_ovfl+Two_exp32  !    "
1010      Time(I)=(Time1+Time_ovfl)*2.E-9-Buff(I,5) MOD 32*1.E-10-Time0_offset
1020      Freq(I)=(Event1-Event0)/(Time(I)-Time(I-1))
1030      Event0=Event1
1040      Time0=Time1
1050    NEXT I
1060    REDIM Buff(1:Block_size,1:Format_words),Time(1:Block_size-1),Freq(1:Bloc
k_size-1)
```

```
1070     OFF CYCLE
1080     SUBEXIT
1090 Disp_update:DISP "Converting binary data.  ";I;"of";Block_size;"completed."
1100     RETURN
1110  SUBEND ! Convert_bin72
1120  !
1130  DEF FNGet_4byte_val(INTEGER Index1,Index2,Buff(*) BUFFER)
1140 Get_4byte_val:! Converts two BASIC INTEGER types into an unsigned 32 bit nu
mber.
1150     COM /Constants/ Format_bytes,Two_exp16,Two_exp32
1160     RETURN (Buff(Index1,Index2)+(Buff(Index1,Index2)<0)*Two_exp16+(Buff(Inde
x1,Index2+1)<0))*Two_exp16+Buff(Index1,Index2+1)
1170  FNEND ! Get_4byte_val
1180  !
1190  !
1200  SUB Plot_data(Time(*),Freq(*))
1210 Plot_data:! Plots Frequency vs. Time
1220     INTEGER I
1230     DISP ! Clear ALPHA screen
1240     GINIT
1250     GCLEAR
1260     GRAPHICS ON
1270     VIEWPORT 25,100*RATIO,30,98
1280     CSIZE 3,.5
1290     PEN 7
1300     LINE TYPE 3
1310     FRAME
1320     LINE TYPE 1
1330     CLIP OFF
1340     Min_freq=MIN(Freq(*))
1350     Max_freq=MAX(Freq(*))
1360     IF Max_freq=Min_freq THEN
1370       Max_freq=Max_freq+1.E-5*ABS(Max_freq)
1380       Min_freq=Min_freq-1.E-5*ABS(Min_freq)
1390     END IF
1400     WINDOW 0,1,0,1
1410     MOVE 0,1
1420     LORG 8
1430     LABEL VAL$(DROUND(Max_freq,10))&"-"
1440     MOVE 0,.5
1450     LABEL "Hz  "
1460     MOVE 0,0
1470     LABEL VAL$(DROUND(Min_freq,10))&"-"
1480     MOVE 0,0
1490     LORG 6
1500     LABEL "|"
1510     MOVE 0,0
1520     LORG 3
1530     LABEL 0
1540     MOVE .5,0
1550     LORG 6
1560     LABEL "seconds"
1570     MOVE 1,0
1580     LABEL "|"
1590     MOVE 1,0
1600     LORG 9
```

```
1610    LABEL PROUND(Time(SIZE(Time,1)),-12)
1620    CLIP ON
1630    PENUP
1640    PEN 3
1650    MOVE 0,0
1660    WINDOW 0,Time(SIZE(Time,1)),Min_freq,Max_freq
1670    PLOT 0,Freq(1)
1680    FOR I=1 TO SIZE(Time,1)-1
1690      PLOT Time(I),Freq(I)
1700      PLOT Time(I),Freq(I+1)
1710    NEXT I
1720    PLOT Time(I),Freq(I)
1730  SUBEND
```

ASCII DATA OUTPUT EXAMPLES

This program shows three different result formats for a frequency measurement:

NOTE

Install an HP 54002A 50Ω Input Pod as the HP 5373A CHANNEL A Input.

- Measurement results for Channel A.
- Measurement results, limit test, and statistics for Channel A.
- Measurement results and gate data for Channels A and B.

The arrays for holding the data from the HP 5373A are dimensioned at the beginning of the program.

Example Program

```
10     !DIMENSION ARRAYS WHICH WILL BE READING DATA FROM THE HP 5373A
20     DIM Meas_a$(10)[22]              !MEASUREMENT DATA
30     DIM Meas_b$(10)[22]
40     DIM Gate_a$(10)[22]              !GATE DATA
50     DIM Gate_b$(10)[22]
60     DIM Limit_a$(10)[5]              !LIMIT TEST RESULTS
70     DIM Stats_a$(10)[22]             !STATISTICAL RESULTS
80     !--------------------------
90     !Initialize constants
100    Counter=703
110    Sample_size=10
120    Nr_of_stats=8
130    !========================================
140    CLEAR Counter
150    OUTPUT Counter;"PRESET"                   !PRESET THE HP 5373A
160    OUTPUT 703;"INP;MODE,COM"                 !COMMON INPUT CHANNELS
170    OUTPUT Counter;"SMOD SINGLE"              !SINGLE SAMPLE MODE
180    OUTPUT Counter;"MEAS;FUNC FREQ;ARM ISAM"  !FREQ, INTERVAL SAMPLING
190    OUTPUT Counter;"SSIZE ";Sample_size       !SET SAMPLE SIZE
200    OUTPUT Counter;"INT;OUTPUT ASCII"         !OUTPUT FORMAT = ASCII
210    OUTPUT Counter;"MENU NUM"                 !TURN OFF EXPANDED DATA
220    !
230    GOSUB Example_1                           !MEAS RESULTS
240    GOSUB Example_2                           !MEAS, LIMITS, STATS RESULTS
250    GOSUB Example_3                           !MEAS, GATE RESULTS
260    !
270    GOTO End_of_program
280    !==========================================
290 Example_1:  !
```

```
300   !Read just measurement results on Channel A
310   PRINT
320   PRINT "EXAMPLE 1; MEAS RESULTS FOR CHANNEL A"
330   TRIGGER Counter
340            !
350   FOR I=1 TO Sample_size                  !READ MEASURMENT RESULTS
360     ENTER Counter USING "%,K";Meas_a$(I)
370     PRINT I,Meas_a$(I)
380   NEXT I
390   CLEAR 703
400   RETURN
410            !===================================================
420 Example_2:!
430   !READ MEASUREMENT RESULTS,LIMIT TEST RESULTS AND STATISTICS ON CHANNEL A
440   PRINT
450   PRINT "EXAMPLE 2: MEAS,LIMIT, AND STATS RESULTS FOR CHANNEL A"
460   OUTPUT Counter;"PROC;SOUR A;LIM ON;STAT ON" !TURN LIMITS AND STATS ON
470   OUTPUT Counter;"MENU NUM"                   !DISPLAY RESULTS AND STATS
480   OUTPUT Counter;"NUM;DISP SPLIT;EXP OFF"
490    !
500   TRIGGER Counter
510   FOR I=1 TO Sample_size         !READ MEASUREMENT AND LIMIT TEST RESULTS
520     ENTER Counter USING "%,K";Meas_a$(I),Limit_a$(I)
530     PRINT I;Meas_a$(I),Limit_a$(I)
540   NEXT I
550   !
560   PRINT
570   PRINT "STATISTICS"
580   FOR I=1 TO Nr_of_stats         !READ STATISTICAL RESULTS
590     ENTER Counter USING "%,K";Stats_a$(I)
600     PRINT I;Stats_a$(I)
610   NEXT I
620   RETURN
630   !=============================================================
640 Example_3:!
650  !READ MEASUREMENT RESULTS AND GATE DATA FOR CHANNELS A AND B
660  PRINT
670  PRINT "EXAMPLE 3: MEASUREMENT AND GATE RESULTS FOR CHANNELS A AND B"
680  !
690  OUTPUT Counter;"MEAS;SOUR (A^B)"                 !SOURCE = CHANNELS A&B
700  OUTPUT Counter;"PROC;SOUR A; LIM OFF;STAT OFF" !TURN LIMITS AND STATS OFF
710  OUTPUT Counter;"NUM;DISP NUM;EXP ON"             !EXPANDED DATA ON
720  !
730  TRIGGER Counter
740  FOR I=1 TO Sample_size                  !READ MEASUREMENT AND GATE RESULTS
750    ENTER Counter USING "%,K";Meas_a$(I),Meas_b$(I),Gate_a$(I),Gate_b$(I)
760    PRINT
770    PRINT I;
780    PRINT TAB(7),"CHANNELS A AND B MEASUREMENTS";Meas_a$(I);Meas_b$(I)
790    PRINT TAB(7),"CHANNELS A AND B GATE TIMES  ";Gate_a$(I);Gate_b$(I)
800  NEXT I
810  RETURN
820  !===============================================================
830 End_of_program: !
840 END
```

FLOATING POINT DATA OUTPUT EXAMPLES

NOTE

Install an HP 54002A 50Ω Input Pod as the HP 5373A CHANNEL A Input.

This program demonstrates the same three result formats as the ASCII program, but the measurement type is time interval, instead of frequency. The three result formats are:

- Measurement results for Channel A to Channel B measurement.
- Measurement results, limit test, and statistics for a Channel A to Channel B measurement.
- Measurement results and event data for a Channel A to Channel B measurement.

The arrays for holding the data from the HP 5373A are dimensioned at the beginning of the program.

Example Program

```
10  !This program demonstrates several format combinations for FLOATING POINT
20  !data output on the HP 5373A.
30  !The data taken in this program uses Time Interval measurements, and
40  !needs input sources for both Channels A and B.
50  !
60  !This program covers 3 examples:
70  ! 1. Measurement results only
80  ! 2. Measurement and limit test results, with statistical results,
90  ! 3. Measurement and gate data results for both channels.
100 !
110 ! ----------------------------------------------------------
120   OPTION BASE 1
130   !DIMENSION ARRAYS WHICH WILL BE READING DATA FROM THE HP 5373A
140   DIM Meas_a(1000)     !MEASUREMENT DATA
150   DIM Meas_b(1000)
160   DIM Event_a(1000)    !GATE DATA
170   DIM Event_b(1000)
180   DIM Limit_a(1000)    !LIMIT TEST RESULTS
190   DIM Stats_a(1000)    !STATISTICAL RESULTS
200   DIM Buff(1000) BUFFER
210 !-----------------------------------------------------
220   !INITIALIZE CONSTANTS
230   Counter=703
240   ASSIGN @Counter TO 703;FORMAT OFF
250   ASSIGN @Controller_buf TO BUFFER Buff(*);FORMAT OFF
260   Sample_size=10
```

```
270 !
280 !
290   Hi_lim=4.40E-7                             !ENTER LIMIT VALUES HERE
300   Lo_lim=4.30E-7
310   CLEAR Counter
320   OUTPUT Counter;"PRESET"                    !PRESET THE HP 5373A
330   OUTPUT Counter;"SMOD SINGLE"               !SINGLE SAMPLE MODE
340   OUTPUT Counter;"MEAS;FUNC TINT;SOUR (AB)" !TI, AUTO arming,SOURCE A-B
350   OUTPUT Counter;"INP;MOD COM"               !Set Input Channels to COMmon
360   OUTPUT Counter;"SSIZE";Sample_size         !SET SAMPLE SIZE
370   OUTPUT Counter;"INT;OUTPUT FPO"            !OUTPUT FORMAT= FLOATING POINT
380   OUTPUT Counter;"MENU NUM"                  !GOTO NUMERIC MENU SCREEN
390   OUTPUT Counter;"NUM;DISP NUM;EXP OFF"      !TURN OFF EXPANDED DATA
400 !
410   GOSUB Example_1                            !MEAS RESULTS
420   GOSUB Example_2                            !MEAS, LIMITS, STATS RESULTS
430   GOSUB Example_3                            !MEAS, GATE RESULTS
440 !
450   GOTO End_of_program
460 !============================================================
470 Example_1: !
480   !READ JUST MEASUREMENT RESULTS
490   PRINT
500   PRINT "EXAMPLE 1: MEAS RESULTS"
510 !
520   TRIGGER Counter
530   GOSUB Read_header                           !READ BLOCK DATA HEADER
540   RESET @Controller_buf
550   TRANSFER @Counter TO @Controller_buf;COUNT Num_bytes
560   FOR I=1 TO Sample_size
570     PRINT I,Buff(I)
580   NEXT I
590   RETURN
600  !============================================================
610 Example_2:!
620   !READ MEASUREMENT RESULTS, LIMIT TEST RESULTS AND STATISTICS.
630   PRINT
640   PRINT "EXAMPLE 2:MEAS,LIMIT,AND STATS RESULTS"
650   OUTPUT Counter;"PROC;SOUR A; LIM ON; STAT ON" !TURN LIMITS AND STATS ON
660   OUTPUT Counter;"PROC;SOUR A;HLIM";Hi_lim      !SET HI LIMIT
670   OUTPUT Counter;"PROC;SOUR A;LLIM";Lo_lim      !SET LOW LIMIT
680   OUTPUT Counter;"NUM;DISP SPLIT"               !DISPLAY RESULTS AND STATS
690           !
700   TRIGGER Counter
710   GOSUB Read_header                             !READ BLOCK DATA HEADER
720   RESET @Controller_buf
730   TRANSFER @Counter TO @Controller_buf;COUNT Num_bytes !READ MEASUREMENT AND
740                                                        !LIMIT TEST RESULTS
750   J=1
760   PRINT "HIGH LIMIT= ";Hi_lim;"    ";"LOW LIMIT= ";Low_lim
770   PRINT
780   PRINT "MEAS #   MEAS RESULT       LIMIT RESULTS"
790   PRINT "                        1=HIGH,0=PASS,-1=LOW"
800   FOR I=1 TO Sample_size
810     PRINT I,Buff(J),"             ";Buff(J+1)!PRINT MEASUREMENT NUMBER,
820                                              !RESULT,AND LIMIT RESULTS
```

```
830     J=J+2
840   NEXT I
850   PRINT
860   PRINT "STATISTICS"                    !PRINT STATISTICS RESULTS
870   PRINT
880   PRINT "Mean ";Buff((Sample_size*2)+1)
890   PRINT "Std dev ";Buff((Sample_size*2)+2)
900   PRINT "Maximum ";Buff((Sample_size*2)+3)
910   PRINT "Minimum ";Buff((Sample_size*2)+4)
920   PRINT "Variance ";Buff((Sample_size*2)+5)
930   PRINT "Root Allen Variance  ";Buff((Sample_size*2)+6)
940   PRINT "RMS ";Buff((Sample_size*2)+7)
950   PRINT "Allan Variance ";Buff((Sample_size*2)+8)
960   RETURN
970          !=======================================================
980 Example_3:   !
990     !READ MEASUREMENT RESULTS AND GATE DATA.
1000  PRINT
1010  PRINT "EXAMPLE 3: MEASUREMENT AND EVENT RESULTS FOR CHANNELS A AND B"
1020    !
1030  OUTPUT Counter;"PROC;SOUR A;LIM OFF;STAT OFF" !TURN LIMITS, STATS OFF
1040  OUTPUT Counter;"NUM;DISP NUM;EXP ON"         !EXPANDED DATA ON
1050    !
1060  TRIGGER Counter
1070  GOSUB Read_header                            !READ BLOCK DATA HEADER
1080    !
1090    !***NOTE: FOR TI AND +\-TI, THERE IS NO EVENT DATA FOR LAST MEASUREMENT
1100    !***SINCE EVENT DATA NEEDS TWO "ENDPOINTS" FOR EACH MEASUREMENT.
1110    !
1120  RESET @Controller_buf
1130  TRANSFER @Counter TO @Controller_buf;COUNT Num_bytes !READ MEASUREMENT
1140                                                       !AND GATE RESULTS
1150  PRINT
1160  PRINT "MEAS #           MEAS RESULT           EVENT A          EVENT B"
1170  J=1
1180  FOR I=1 TO Sample_size-1        !READ MEASUREMENT AND GATE RESULTS
1190    Meas_a(I)=Buff(J)
1200    Event_a(I)=Buff(J+1)
1210    Event_b(I)=Buff(J+2)
1220    PRINT I,"     ",Meas_a(I),"    ",Event_a(I),"      ",Event_b(I)
1230    J=J+3
1240  NEXT I
1250  RETURN
1260     !========================================================
1270 Read_header:    !
1280  ENTER Counter USING "#,8A";Header$
1290  Num_bytes=VAL(Header$[3])
1300  PRINT Num_bytes;" BYTES EXPECTED"
1310  RETURN
1320      !========================================================
1330 End_of_program:     !
1340  END
```

F PROGRAM CONVERSION

APPENDIX

F

PROGRAM CONVERSION

HP 5371A to HP 5373A PROGRAM CONVERSION

The HP 5373A feature set is considerably enhanced over that of the HP 5371A. Thus, for the most part, the HP 5371A HP-IB command set is a subset of the HP 5373A HP-IB commands.

Those HP 5371A commands which have been changed in the HP 5373A are shown in *Table F-1,* along with their equivalent HP 5373A commands. When the HP 5373A is sent one of the HP 5371A commands from this table, it will translate to the equivalent new command, then execute the new command. If there is no equivalent command in the HP 5373A (for example: VAUTscale), the HP 5373A status line will display "HP5371A Graphics command no longer used". Whenever this message is displayed, an error number will be logged in the error queue so you may track occurrences of any non convertible commands. We recommend however, that you change HP 5371A commands to their HP 5373A equivalents so that your program may be more easily supported from this programming manual.

HP 5372A to HP 5373A PROGRAM CONVERSION

The HP 5373A feature set is similar to that of the HP 5372A. Thus, for the most part, the HP 5372A HP-IB command set is a subset of the HP 5373A HP-IB commands. Differences, and similarities, are discussed below.

- HP 5373A commands having the same names as HP 5372A commands are the same in both instruments.
- Even though HP 5373A commands may have the same names as HP 5372A commands, you may not find them grouped the same way in the HP 5373A displays as they were in the HP 5372A.
- The HP 5373A Preset measurement function is PRF or Frequency. The HP 5372A Preset measurement is Time Interval. Other HP 5373A Preset conditions also differ from those of the HP 5372A.

- Some HP 5373A default measurement setup conditions differ from those of the HP 5372A.
- Front-panel hardkey identification for the HP 5373A is the same as for the HP 5372A.
- The HP 5373A does not provide Window Margin Analysis; the HP 5372A does. HP 5372A Window Margin Analysis commands and queries are not recognized by the HP 5373A. These commands and queries are: CMARgin?, DWINdow, DWINdow(?), EMARgin?, ENOIse?, LMARgin?, LNOIse?, MLEVel, MLEVel?, MREad, MREad?, PSHift?, WALL, WHIStogram, WMULtiple, WOFFset, WTYPe, WTYPe?.
- The HP 5373A FUNCTION menu PRF/Frequency Softkey action depends on the pod installed in the selected channel. If the pod is an HP 53702A Envelope Detector, PRF will be chosen. If the pod is an HP 53702A 50Ω Input Pod, Frequency will be chosen. If a two-channel measurement has been specified, and either channel does not have an HP 53702A pod, Frequency will be chosen.

  Similar logic applies to the PRF/Period Softkey.
- The HP 5373A does not make Peak Amplitude measurements. The HP 5372A does.
- The HP 5373A Pulse Width measurement is the same as the HP 5372A Positive Pulse Width measurement.
- The HP 5373A Pulse Offtime measurement is the same as the HP 5372A Negative Pulse Width measurement.
- The HP 5373A can make Frequency Deviation measurements. The HP 5372A cannot.
- HP 5373A PROCess (MATH) menu CARRier parameters (except MANual) are different than those of the HP 5372A.

Table F-1. HP 5373A to HP 5371A Command Equivalents

HP 5371A COMMANDS	HP 5373A COMMANDS
MARKer,X	SMARker,BLACk
MARKer,O	SMARker,WHITe
MARKer,PEAK	MMAX
MARKer,CENTer	no equivalent
MARKer,RIGHt, #	MRIGht, #
MARKer,LEFT, #	MLEF, #
XMAR?	MDMode,MARKer;XVAL?
YMAR?	MDMode,MARKer;YVAL?
DELTA,ON	MDMode,DELTa
DELTA,OFF	MDMode,MARKer
XDEL?	MDMode,DELTa;XVAL?
YDEL?	MDMode,DELTa;YVAL?
STATistics	MDMode,STATistics
MIN?	HMINimum?
MAX?	HMAXimum?
MEAN?	HMEan?
SDEV?	HSDev?
RESCale	ZOOM,FULL
VAUToscale	no equivalent
XAUToscale,ON	XMSCale,ON
XAUToscale,OFF	XMSCale,OFF
YAUToscale,ON	YMSCale,ON
YAUToscale,OFF	YMSCale,OFF
CHANnel A/B	VCHannel A/B/C
UPDate,OFF	no equivalent
BINS, #	no equivalent
SCRoll,LEFT, #	no equivalent
SCRoll,RIGHt, #	no equivalent
DISPlay,ON	SGRaph,MAIN
MEMory,ON	SGRaph,MEMory
AEVents?	EVT1?
BEVents?	EVT2?
EVENts,ON/OFF	no equivalent
LIMit,ON/OFF	no equivalent

INDEX

Index

*Commands

A

B

C

D

H

I

K

L

M

N

O

P

R

S

T

U

W