HEWLETT PACKARD

HP 3245A
Universal
Source

HP 3245A Operating and Programming Manual

HP 3245A Universal Source

Operating and Programming Manual

HEWLETT® PACKARD

Manual Part Number: 03245-90001
Microfiche Part Number: 03245-99001

Printed SEPTEMBER 1988 Edition 1
Printed in U.S.A. U0394
Update 2, MARCH 1994

Notice

Hewlett-Packard to Agilent Technologies Transition

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. To reduce potential confusion, the only change to product numbers and names has been in the company name prefix: where a product name/number was HP XXXX the current name/number is now Agilent XXXX. For example, model number HP8648 is now model number Agilent 8648.

Contacting Agilent Sales and Service Offices

The sales and service contact information in this manual may be out of date. The latest service and contact information for your location can be found on the Web at:

`http://www.agilent.com/find/assist`

If you do not have access to the Internet, contact your field engineer or the nearest sales and service office listed below. In any correspondence or telephone conversation, refer to your instrument by its model number and full serial number.

United States
(tel) 1 800 452 4844
(fax) 1 800 829 4433

Canada
(tel) +1 877 894 4414
(fax) +1 888 900 8921

Europe
(tel) (31 20) 547 2323
(fax) (31 20) 547 2390

Latin America
(tel) (305) 269 7500
(fax) (305) 269 7599

Japan
(tel) (81) 426 56 7832
(fax) (81) 426 56 7840

Australia
(tel) 1 800 629 485
(fax) (61 3) 9210 5947

New Zealand
(tel) 0 800 738 378
(fax) 64 4 495 8950

Asia Pacific
(tel) (852) 3197 7777
(fax) (852) 2506 9284

Agilent Technologies

Printed in USA July 2004

Printing History

The Printing History shown below lists the printing dates of all Editions and Updates created for this manual. The Edition number changes as the manual undergoes subsequent revisions. Editions are numbered sequentially starting with Edition 1. Updates, which are issued between Editions, contain individual replacement pages which the customer uses to update the current Edition of the manual. Updates are numbered sequentially starting with Update 1. When a new Edition is created, all Updates associated with the previous Edition are merged into the manual. Each new Edition or Update also includes a revised copy of this printing history page.

Many product updates and revisions do not require manual changes and, conversely, manual corrections may be done without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

Edition 1 (Part Number 03245-90001) . . . . SEPTEMBER 1988
Update 1 (Part Number 03245-90091) . . . . MAY 1991
Update 2 (Part Number 03245-90092) . . . . MARCH 1994

Hewlett-Packard Company
3000 Hanover Street, Palo Alto, California 94304

HEWLETT PACKARD

Herstellerbescheinigung

Hiermit wird bescheinigt, daß das Gerät/System HP 3245A in Übereinstimmung mit den Bestimmungen von Postverfügung 1046/84 funkentstört ist.

Der Deutschen Bundespost wurde das Inverkehrbringen dieses Gerätes/Systems angezeigt und die Berechtigung zur Überprüfung der Serie auf Einhaltung der Bestimmungen eingeräumt.

Zusatzinformation fur Meß- und Testgeräte

Werden Meß- und Testgeräte mit ungeschirmten Kabeln und/oder in offenen Meßaufbauten verwendet, so ist vom Betreiber sicherzustellen, daß die Funk-Entstörbestimmungen unter Betriebsbedingungen an seiner Grundstücksgrenze eingehalten werden.

Manufacturer's declaration

This is to certify that the equipment HP 3245A is in accordance with the Radio Interference Requirements of Directive FTZ 1046/84. The German Bundespost was notified that this equipment was put into circulation, the right to check the series for compliance with the requirements was granted.

Additional Information for Test- and Measurement Equipment

If Test- and Measurement Equipment is operated with unscreened cables and/or used for measurements on open set-ups, the user has to assure that under operating conditions the Radio Interference Limits are still met at the border of his premises.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

HEWLETT PACKARD

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 National Institute of Standards and Technologies, to the extent allowed by the the Institute's calibration facility, and to the calibration facilities of other International Standards Organization members.

WARRANTY

This Hewlett-Packard instrument product is warranted against defects in materials 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.

Duration and conditions of warranty for this instrument may be superceded when the instrument is integrated into (becomes a part of) other -hp- instrument products.

Hewlett-Packard 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. Hewlett-Packard 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. HEWLETT-PACKARD 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. HEWLETT-PACKARD 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.

F

HEWLETT PACKARD

SAFETY SUMMARY

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the instrument. Hewlett-Packard Company assumes no liability for the customer's failure to comply with these requirements. This is a Safety Class 1 instrument.

GROUND THE INSTRUMENT

To minimize shock hazard, the instrument chassis and cabinet must be connected to an electrical ground. The instrument is equipped with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE

Do not operate the instrument in the presence of flammable gases or fumes. Operation of any electrical instrument in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS

Operating personnel must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

DO NOT SUBSTITUTE PARTS OR MODIFY INSTRUMENT

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification to the instrument. Return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DO NOT OPERATE A DAMAGED INSTRUMENT

Whenever it is possible that the safety protection features built into this instrument have been impaired, either through physical damage, excessive moisture, or any other reason, REMOVE POWER and do not use the instrument until safe operation can be verified by service-trained personnel. If necessary, return the instrument to a Hewlett-Packard Sales and Service Office for service and repair to ensure that safety features are maintained.

DANGEROUS PROCEDURE WARNINGS

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

WARNING

Dangerous voltages, capable of causing death, are present in this instrument. Use extreme caution when handling, testing, and adjusting.

A

Operating and Safety Symbols

Symbols Used On Products And In Manuals

~ LINE — AC line voltage input receptacle.

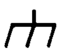
Instruction manual symbol affixed to product. Warns and cautions the user to refer to respective instruction manual procedures to avoid personal injury or possible damage to the product.

Indicates dangerous voltage – terminals connected to interior voltage exceeding 1000 volts.

 OR 

Protective conductor terminal. Indicates the field wiring terminal that must be connected to earth ground before operating equipment – protects against electrical shock in case of fault.

Clean ground (low-noise). Indicates terminal that must be connected to earth ground before operating equipment – for single common connections and protection against electrical shock in case of fault.

 OR 

Frame or chassis ground. Indicates equipment chassis ground terminal – normally connects to equipment frame and all metal parts.

ATTENTION
Static Sensitive

Affixed to product containing static sensitive devices – use anti-static handling procedures to prevent electrostatic discharge damage to components.

NOTE

NOTE

Calls attention to a procedure, practice, or condition that requires special attention by the reader.

CAUTION

CAUTION

Calls attention to a procedure, practice, or condition that could possibly cause damage to equipment or permanent loss of data.

WARNING

WARNING

Calls attention to a procedure, practice, or condition that could possibly cause bodily injury or death.

Table of Contents

PART I - GETTING STARTED

Chapter 1 - Using This Manual

Chapter 2 - Initial Operation

PART II - EXAMPLE WAVEFORM PROGRAMS

Chapter 3 - DC Voltage/Current Programs

Chapter 4 - Defined Waveforms Programs

Chapter 5 - Arbitrary Waveforms Programs

PART III - INSTRUMENT OPERATIONS

Chapter 6 - Front Panel Operation

Chapter 7 - Arrays and Subroutines

Chapter 8 - Input/Output Operations

Chapter 9 - Other Instrument Functions

PART IV - PROGRAMMING OUTPUTS/WAVEFORMS

Chapter 10 - Programming DC Outputs

Chapter 11 - Programming Defined Waveforms

Chapter 12 - Programming Arbitrary Waveforms

List of Examples

Chapter 2 - Initial Operation

Chapter 3 - DC Voltage/Current Programs

Chapter 4 - Defined Waveforms Programs

Chapter 5 - Arbitrary Waveforms Programs

Chapter 6 - Front Panel Operation

Chapter 7 - Arrays and Subroutines

Chapter 8 - Input/Output Operations

Chapter 9 - Other Instrument Functions

Chapter 10 - Programming DC Outputs

Chapter 10 - Programming DC Outputs (Cont'd)

Chapter 11 - Programming Defined Waveforms

Chapter 12 - Programming Arbitrary Waveforms

Chapter 13 - Triggering Outputs/Waveforms

Chapter 14 - Advanced Programming Topics

List of Figures

Chapter 7 - Arrays and Subroutines

Chapter 8 - Input/Output Operations

Chapter 9 - Other Instrument Functions

Chapter 10 - Programming DC Outputs

Chapter 11 - Programming Defined Waveforms

Chapter 12 - Programming Arbitrary Waveforms

Chapter 13 - Triggering Outputs/Waveforms

Chapter 14 - Advanced Programming Topics

List of Tables

Chapter 10 - Programming DC Outputs

Chapter 11 - Programming Defined Waveforms

Chapter 13 - Triggering Outputs/Waveforms

PART I
GETTING STARTED

Contents

Chapter 1
Using This Manual

Chapter 1

Using This Manual

Manual Organization

This HP 3245A Operating and Programming Manual shows how to install, operate, and program the HP 3245A Universal Source (HP 3245A). The manual has four parts: Part I - Getting Started; Part II - Example Waveforms Programs; Part III - Instrument Operations; and Part IV - Programming Outputs/Waveforms.

An example program disc is supplied with this manual. Many of the programs listed in this manual are stored on the disc so you can quickly and easily run programs of interest to you. Refer to "Example Programs Disc" for details on using the disc. A summary description of the manual follows.

Part I - Getting Started

Part I (Chapters 1 and 2) shows how to use this manual and how to set the HP 3245A for initial operation. Chapter 1 summarizes manual contents, describes initial inspection, and shows typical sequences to use the manual set for HP-IB* or bench operation of the instrument. Chapter 2 shows how to install and initially operate the instrument. Refer to the HP 3245A Calibration Manual for operational verification, performance tests, or adjustments to the HP 3245A.

Part II - Example Waveforms Programs

Part II (Chapters 3 through 5) lists several example programs to output DC voltages and currents, defined (sine, ramp, square), and arbitrary (user-defined) waveforms. You may be able to use these example programs directly to generate required outputs. If your application requires different parameters or features, refer to Part III - Instrument Operations or to Part IV - Programming Outputs/Waveforms for details.

Part III - Instrument Operations

Part III (Chapters 5 through 9) shows how to program the HP 3245A for its operational functions, including the front panel operation, using arrays and subroutines, input/output operations, and general instrument functions such as timing and memory operations. You can use these chapters to program the HP 3245A for your specific application requirements. Refer to the HP 3245A Command Reference Manual for command descriptions.

Part IV - Programming Outputs/Waveforms

Part IV (Chapters 10 through 14) provides a description of programming techniques and triggering methods required to generate DC outputs, defined waveforms, and arbitrary waveforms. Refer to the HP 3245A Command Reference Manual for command descriptions

* HP-IB (Hewlett-Packard Interface Bus) is Hewlett-Packard's implementation of IEEE 488-1978 and ANSI MC1.1

Example Programs Disc

A programs disc was shipped with this manual (Example Programs, Calibration and Performance Tests, 03245-10010) and contains listings for many example programs as well as performance and calibration routines. When a program is stored on the programs disc, the title listed in the manual shows the disc file name in parenthesis The disc is organized into a series of subdirectories. The ones of interest are EX_CH_2_13, containing examples from chapters 2 through 13, and directory EX_CH_14, containing examples from chapter 14.

For example, the first program in Chapter 3 of this manual is titled *Example 3-1: Power-On DC Voltage Outputs (DCVP3).* Since DCVP3 is enclosed in parentheses in the title, this program is stored on the program disk in directory EX_CH_2_13 under the file name DCVP3.

To run an example program from the disc, first insert the disc into your controller
Then, for HP 9000 Series 200/300 controllers, use MSI "*directory_name*" to change to the appropriate directory and then LOAD "*file_name*". Press RUN to run the program.
For example, to run the program DCVP3, use MSI "EX_CH_2_13" followed by LOAD "DCVP3" and then press RUN

Typical Programming Sequence

The steps to take to operate and/or program the HP 3245A depend on whether you will use the HP 3245A via HP-IB (HP-IB operation) or whether you will use the HP 3245A as a bench instrument (bench operation) and enter commands from the front panel only.

HP-IB Operation

With HP-IB operation, the full range of HP 3245A features is available. Figure 1-1 shows a suggested sequence to use the HP 3245A manual set for HP-IB operation. A summary of the steps follows.

Getting Started (Part I)

To get started, read Chapter 2 to install and initially operate the HP 3245A. If you need to do operational verification, performance testing, or adjustments for the instrument, refer to the HP 3245A Calibration Manual.

Example Waveform Programs (Part II)

By using the example programs in Chapters 3, 4, or 5, you may be able to generate the DC output or AC waveform required for your application without having to read any of the rest of the manual. Chapter 3 includes DC voltage/current programs, Chapter 4 includes defined (sine, ramp, and square) AC waveforms, and Chapter 5 includes arbitrary (user-defined) waveforms.

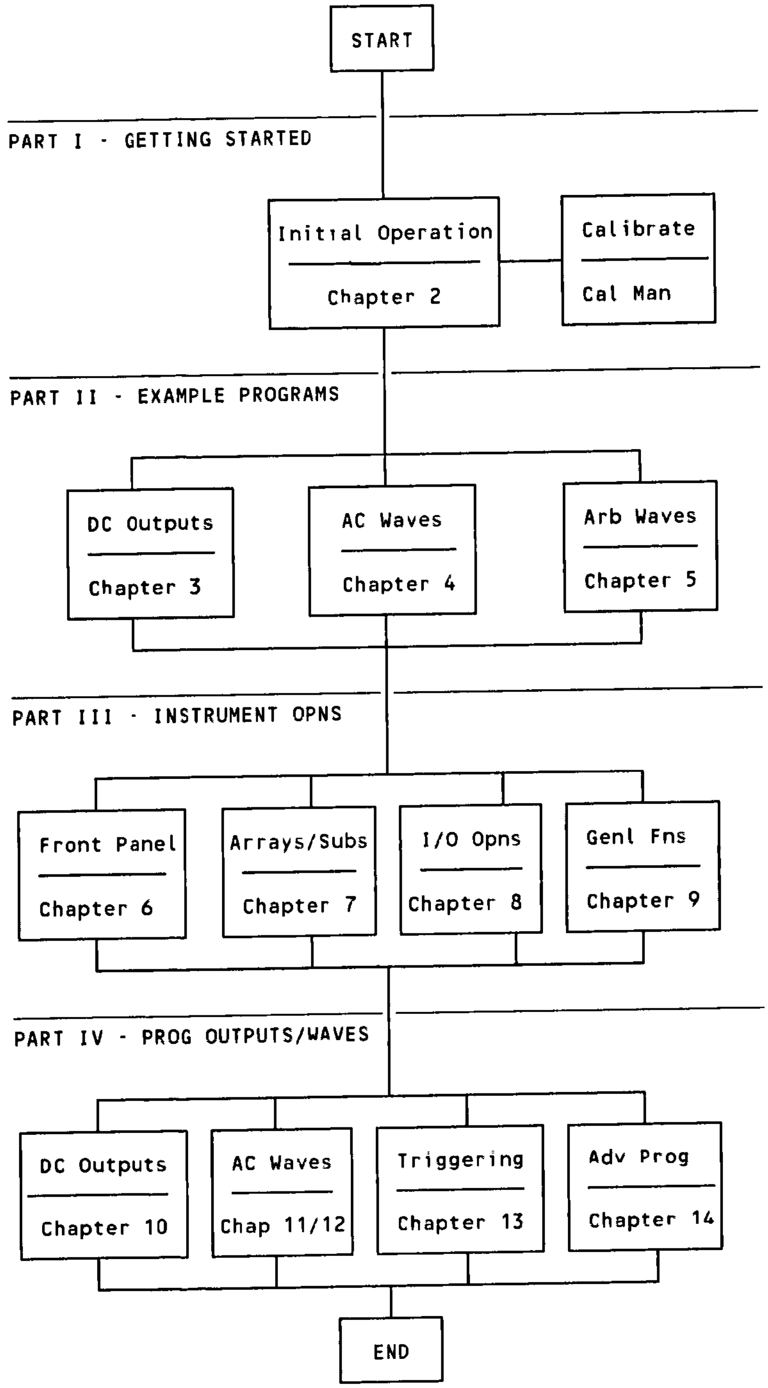


Figure 1-1. Typical Programming Sequence - HP-IB Operation

If you find an example program which meets your needs, simply load the appropriate Example Programs disc in your controller, use LOAD "file_name" and press RUN to run the program. If you don't find the specific program required, you can refer to Chapters 6 through 14 for details on commands or parameters to be changed.

Instrument Operations (Part III)

As required, refer to Chapters 6 through 9 for information on instrument operations needed for your application. Chapter 6 describes front panel operation, Chapter 7 shows how to use arrays and subroutines, Chapter 8 describes input/output operations, and Chapter 9 describes general instrument functions such as timing and memory operations.

Programming Outputs/Waveforms (Part IV)

As required, refer to Chapters 10 through 14 for information on programming DC outputs or AC waveforms. You may also want to refer to the HP 3245A Command Reference Manual for details on commands described.

Bench Operation

With bench operation, the HP 3245A is operated and programmed from the HP 3245A front panel with no HP-IB connection. Figure 1-2 shows a suggested sequence of steps to use the HP 3245A manual set for bench operation of the instrument. A summary of the steps follows.

Getting Started

To get started, read Chapter 2 - Initial Operation to install and initially operate the HP 3245A (skipping the HP-IB information). If you need to do operational verification, performance testing, or adjustments for the instrument, refer to the HP 3245A Calibration Manual.

Front Panel Operation

Then, go to Chapter 6 - Front Panel Operation for a description of the HP 3245A front panel. Chapter 6 describes front panel keys and shows how to use MENU operation to enter commands.

Instrument Operations

As required, refer to Chapters 7 through 9 for information on instrument operations needed for your application. In many cases, you can enter the commands and parameters needed by using the front panel keys and/or MENU operation.

Waveform Generation

As required, refer to Chapters 10 through 13 for information on generating DC outputs and AC waveforms. You may also want to refer to the HP 3245A Command Reference Manual for details on commands.

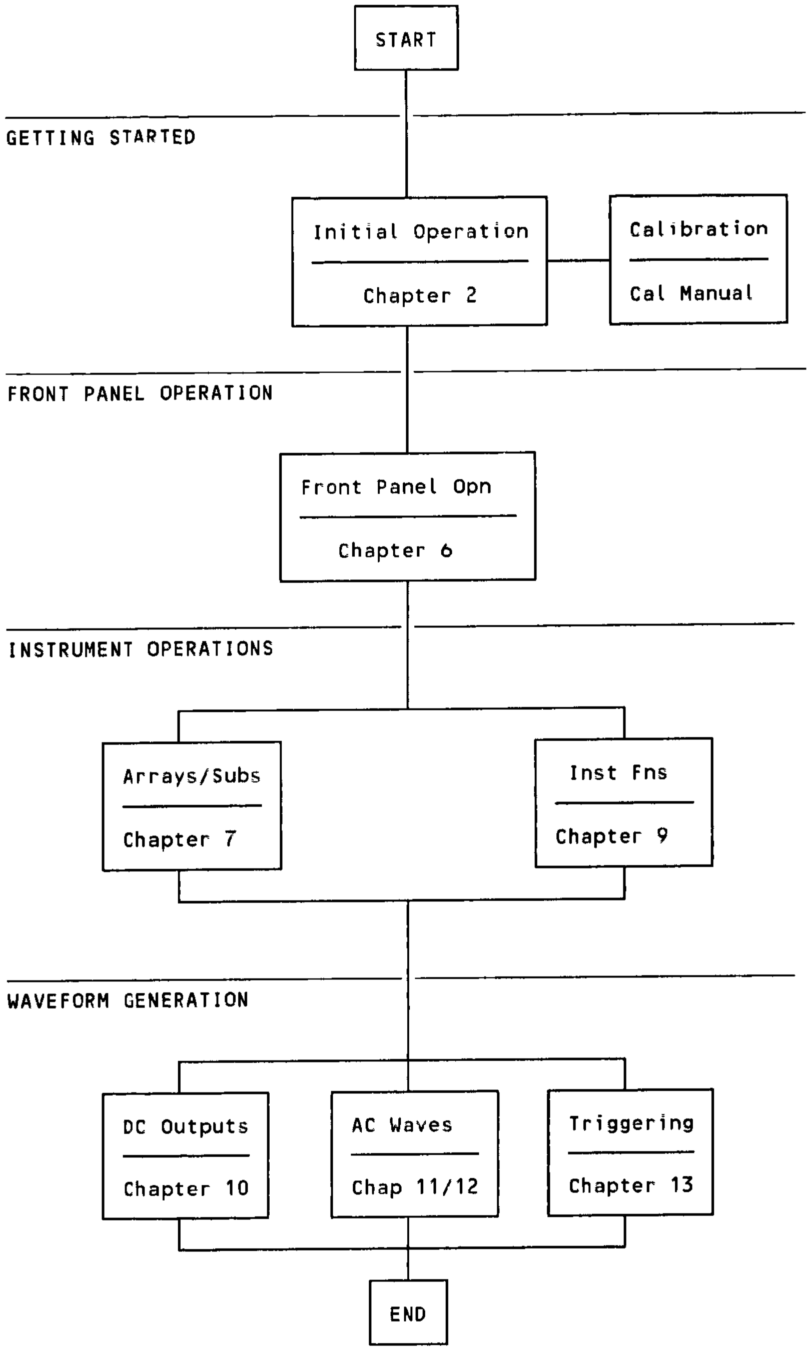


Figure 1-2. Typical Operating Sequence - Bench Operation

PART II
EXAMPLE WAVEFORM PROGRAMS

Contents

Chapter 2
Initial Operation

Chapter 2

Initial Operation

Chapter Contents

This chapter shows how to install and initially operate the HP 3245A Universal Source (HP 3245A). It includes:

- Initial inspection.
- Instrument features.
- Installing the instrument.
- Front panel operation.
- HP-IB programming.

Getting Started

This section shows how to get started using the HP 3245A. It includes initial inspection procedures and summarizes HP 3245A features.

Initial Inspection

If you have just received your HP 3245A, carefully inspect the instrument and verify that the shipment contents are complete and undamaged. The following steps show how to inspect the HP 3245A for damage, to verify that the shipment is complete, and to return the HP 3245A to Hewlett-Packard for repair service if required.

Inspect For Damage

Carefully inspect the HP 3245A as soon possible to ensure that it is not damaged. Keep the shipping container and shipping materials until the shipment contents have been checked and the HP 3245A has been checked for proper operation.

WARNING

If the HP 3245A appears to be damaged, has been subjected to severe transport stress, or has been stored for a long time under adverse conditions, do not use the instrument until safe operation can be verified by service-trained personnel.

Verify Shipment Contents

Verify that the following items are in the shipment. If the HP 3245A is damaged, or if the shipment is incomplete, promptly notify your nearest Hewlett-Packard Sales and Support Office.

Instrument/Accessories

- HP 3245A Universal Source (Std or Option 001).
  - -- AC line power fuses (one 500 mAT and one 1.5A NTD).
  - -- AC line power fuse cap (gray).
  - -- HP 3245A keyboard overlay.

Manuals/Example Program Discs

- HP 3245A Operating/Programming Manual (03245-90001)
  - -- 3.5" disc - Example Programs: Chapters 2-13.
  - -- 3.5" disc - Example Programs: Chapter 14.
  - -- 5.25" disc - Example Programs: Chapters 2-13.
  - -- 5.25" disc - Example Programs: Chapter 14.
- HP 3245A Command Reference Manual (03245-90002)
  - -- 3.5" disc - Example Programs: Command Reference.
  - -- 5.25" disc - Example Programs: Command Reference.
- HP 3245A Calibration Manual (03245-90003)
  - -- 3.5" disc - Automated Adjustments.
  - -- 5.25" disc - Automated Adjustments.

Return For Repair (as Required)

The HP 3245A may be repaired at an HP Service Center whether or not the instrument is under warranty. Contact your nearest Hewlett-Packard Sales and Support Office for shipping instructions prior to returning the instrument. If you need to return the HP 3245A for repairs, package the instrument in the original shipping material to prevent transit damage. (Transit damage is not covered by warranty.)

When returning the instrument, attach a tag to the shipping container identifying the owner and indicating the service or repair needed. Include the model number and serial number of the HP 3245A. The HP 3245A serial number has the form 0000A00000 and is located on the lower left corner of the rear panel.

HP 3245A Features

Figure 2-1 summarizes HP 3245A features when HP-IB operation is used. An overview of the instrument features follows.

DC Outputs/AC Waveforms

The HP 3245A can generate precise DC voltage outputs from -10.25 VDC to + 10.25 VDC with 6 digits of resolution (24 bits) in high resolution mode or 3.5 digits (12 bits) in low resolution mode. Also, in low resolution mode, outputs may be triggered. DC currents can be output in range of -0.1 A to +0.1 A. AC voltage and current outputs include sine, square, and arbitrary waveforms up to 1 MHz plus ramp waveforms up to 100 kHz. Variable duty cycles from 5% to 95% are selectable up to 100 kHz.

The HP 3245A has seven voltage ranges and four current ranges plus selectable 0 Ω or 50 Ω output impedance. Outputs can be generated from one or (optionally) two channels Each channel is independently programmable and outputs can be generated from front or rear panel terminals.

Triggering/Timing

The HP 3245A has selectable input trigger sources/events, including two internal trigger buses, external inputs, and software triggering. In addition, a Sync pulse output is available from front and/or rear panel ports. Gated, dual-frequency, and synchronized mode operation for output waveforms can be selected by the trigger mode.

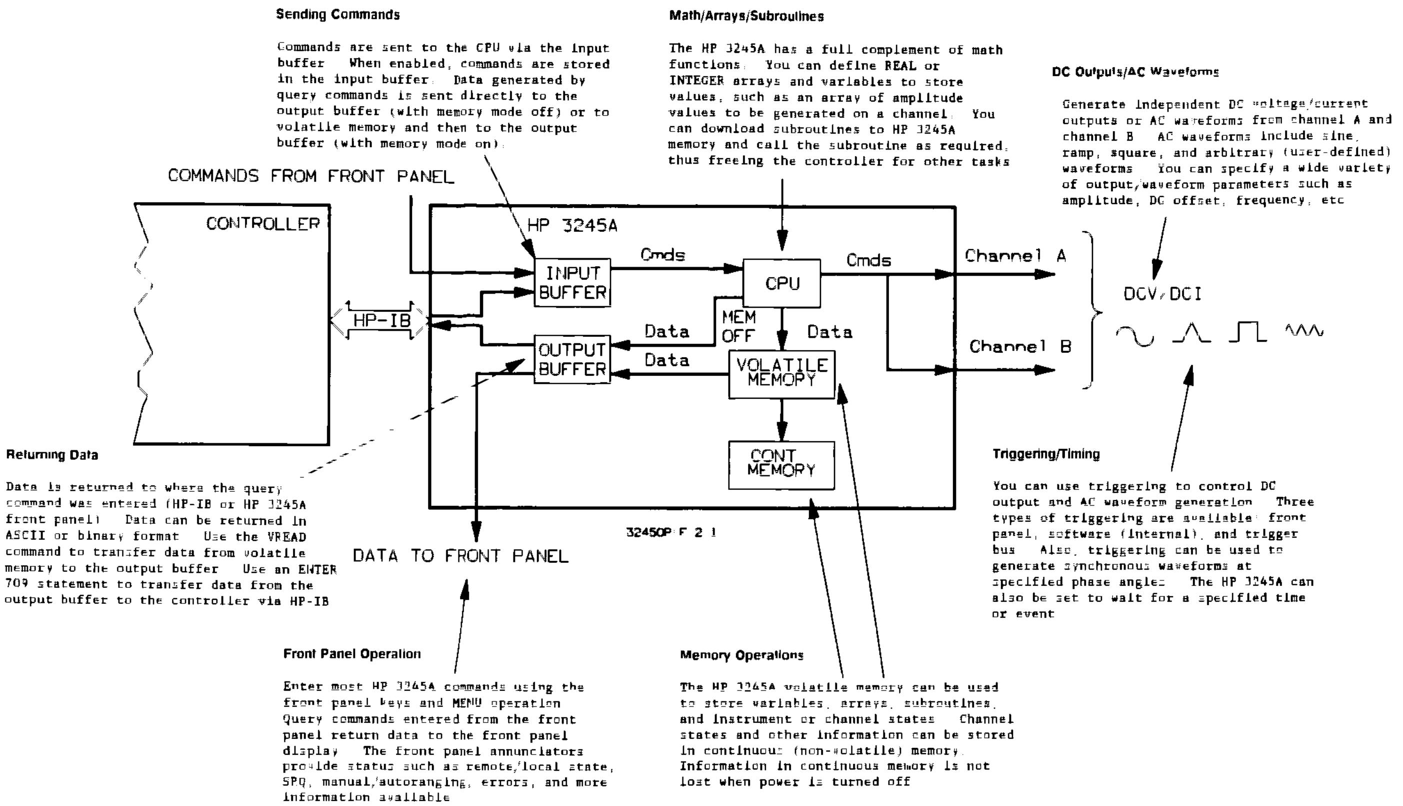


Figure 2-1. HP 3245A Features

For synchronized mode operation, a single reference frequency can be used for two-channel or multiple HP 3245A synchronized waveform outputs. In addition, the HP 3245A can be programmed to wait a specified time or to wait for a specified event.

Input/Output Operation

The HP 3245A can be programmed from the front panel or via HP-IB. ASCII or binary data formats are available and input/output buffering can be used for HP-IB operation. Waveforms can be stored in HP 3245A memory for future output or can be precomputed and then stored for faster output. Instrument and/or channel states can be stored in continuous (non volatile) memory and recalled in the event of a power failure.

Math/Arrays/Subroutines

The HP 3245A has a full range of math functions. Also, you can download data into subroutines, store and recall channel states, generate status and service requests, and set timing functions.

Front Panel Operation

You can enter most HP 3245A commands from the front panel, using the keys or using MENU operation. Commands entered from the front panel which return data (query commands) return the data to the front panel display.

Installing the HP 3245A

This section shows how to install and apply power to the HP 3245A.

Warnings and Cautions

WARNINGS and CAUTIONS which apply to operation and programming of the HP 3245A follow. Please review the WARNINGS and CAUTIONS before applying power to the instrument.

WARNING

SHOCK/FIRE HAZARD. Only qualified, service-trained personnel who are aware of the hazards involved should install or configure the HP 3245A. Turn off all power to the instrument before attempting repairs or connecting cables. For protection from electrical shock, the power cord ground must not be defeated. For continued fire protection, replace fuse only with one of the same type and rating.

CAUTION

POSSIBLE INSTRUMENT DAMAGE. Before connecting the HP 3245A to an AC power source, verify that the line voltage selector switch is set to match the AC line voltage and the proper line fuse is installed.

CAUTION

VOLTAGE/CURRENT LIMITS. Voltage/current limits for all BNC connectors on the HP 3245A are TTL-compatible 5.0 Vdc @ -5.2 mA (HIGH) and 0.4 Vdc @ 48 mA (LOW).

The HP 3245A's output BNCs are specified to source up to 100 mA and are current limited at approximately 120 mA. Application of voltages higher than ±15 Vpeak external to the output terminals may open the output relays. Each channel contains two output relays: one each for the front and rear panel BNC connectors. The relays are fused for additional protection.

Front/Rear Panel Description

A summary description of the HP 3245A front and rear panels follows. Please take a minute to get acquainted with the panel functions before beginning HP 3245A installation.

Front Panel Features

As shown in Figure 2-2, the HP 3245A front panel consists of an ON/OFF switch, a display, three groups of keys (FUNCTION/RANGE, MENU, and NUMERIC/USER), and a set of input/output connectors. The display shows the output voltage or current, commands to be entered, or can be used in monitor mode to check the channel state. The keys are used to enter commands, parameters, and numeric data. The input/output connectors provide input or output terminals for signal outputs and trigger pulses.

Rear Panel Features

As shown in Figure 2-3, the HP 3245A rear panel has five BNC connectors (Freq Ref, Channel A Output, Channel B Output, TB0, and TB1), an HP-IB connector, a line power connector, a fuse holder, and a line voltage selector switch. When rear panel operation is selected, the Output connectors provide the ports for signal outputs The Trigger Bus connectors provide input/output terminals for trigger signals, while the frequency reference port provides an input/output teminal for the reference frequency (used for multiple channel outputs)

Installation Steps

There are three steps to install an HP 3245A. First, select a location, then make rear panel connections, and then connect cables to input/output terminals.

Selecting Location

The HP 3245A has four feet and two tilt stands which allow it to be used as a bench instrument. The HP 3245A can also be mounted in a standard 19-inch rack using the optional rack mount kits shown in Table 2-1. (Order the rack mount kits from your nearest Hewlett-Packard Sales and Support Office.) Install the HP 3245A in a rack or place the instrument on a bench as required.

Display

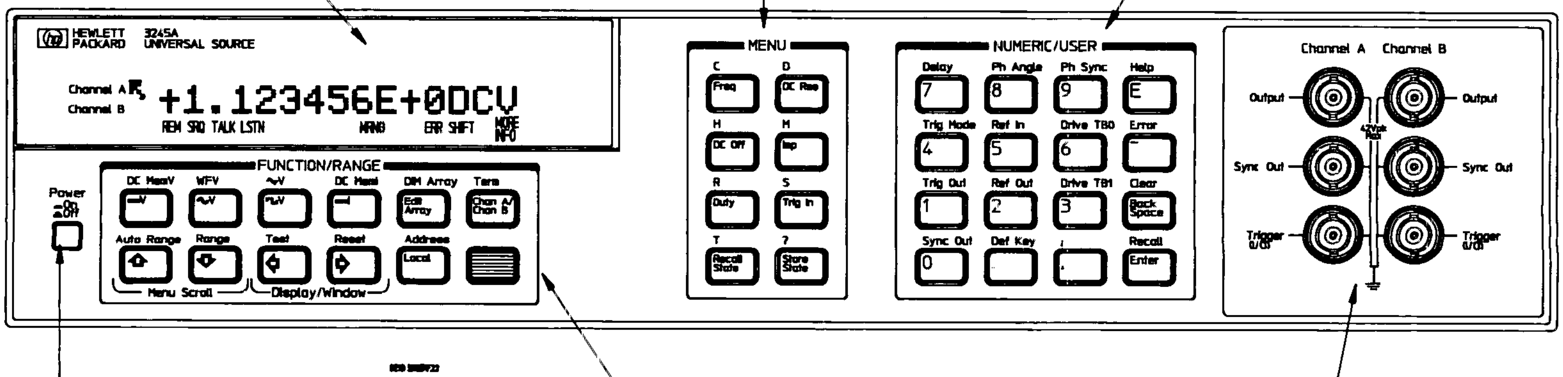
Displays output voltage/current or commands to be entered The annunciators indicate HP-IB status (REM, SRQ, TALK, and LISTEN), fixed range vs. autorange (MNRG), error condition (ERR), shifted keyboard (SHIFT), and additional information to be scrolled onto the display (MORE INFO). Also, the current USE channel (A or B) is indicated

MENU Keys

Use these keys to select waveform parameters such as frequency, DC resolution and offset, output impedance, and duty cycle Also, set the trigger input source and store and recall channel states In shifted mode, provide entry point for commands to be entered from the front panel

NUMERIC/USER Keys

In unshifted mode, use these keys to enter numeric data and/or punctuation as required and to enter commands into the HP 3245A. In shifted mode, enter various waveform parameters and trigger modes.



ON/OFF Switch

Press ON for power-on, press OFF for power off to the HP 3245A

FUNCTION/RANGE Keys

Use these keys to output AC or DC waveforms, edit stored arrays, select the channel to be used, set fixed range or autorange, test or reset the HP 3245A, or scroll commands or display information.

Input/Output Connectors

Each channel (A and B) has an Output connector to output the waveform, a Sync Out connector to output the sync pulse signal, and a Trigger (I/O) connector for input or output triggering.

Figure 2-2. Front Panel Functions

Freq Ref Connector

When synchronized waveform mode is set, output the reference frequency from this connector.

Channel Output Connectors

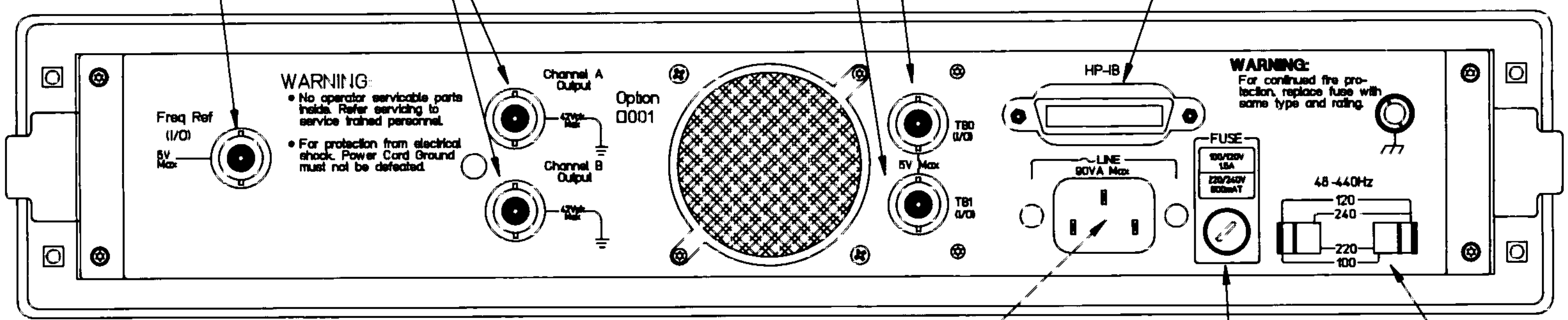
When rear panel output is selected, output the channel A waveform from the Channel A Output connector or output the channel B waveform from the Channel B Output connector. When rear panel output is selected, waveforms are not output from the front panel Output connectors.

Trigger Bus Connectors

The TB0 (I/O) Trigger Bus Connector provides an input/output port to trigger the HP 3245A via the TB0 trigger bus. The TB1 (I/O) Trigger Bus Connector provides an input/output port to trigger the HP 3245A via the TB1 trigger bus.

HP-IB Connector

Connection point for the HP-IB cable from the controller.



04E101 3245OPF 2.3

Line Power Connector

Connector for AC power cord. Use the proper cord.

Fuse Holder

Contains 1.5 A (@100V/120V) or 500mAT (@220V/240V) fuse

Line Selector Switch

Switch to set the HP 3245A for the appropriate line voltage (100 Vac, 120 Vac, 220 Vac, or 240 Vac).

Figure 2-3. Rear Panel Functions

Table 2-1. HP 3245A Rack Mount Kits

Description	Option Number	Part Number for Retrofit
Front Handle Kit	907	5061-9688
Rack Flange Kit	908	5061-9674
Rack Flange Kit/handles	909	5061-9675

Rear Panel Connections

As shown in Figure 2-4, the steps to make HP 3245A rear panel connections are:

- Set AC line selector switches.
- Install AC line fuse.
- Connect HP-IB cable.
- Connect AC power cord.

Set AC Line Selector Switches

The HP 3245A can be operated from a single phase power source delivering 100 Vac, 120 Vac, 220 Vac, or 240 Vac (all values RMS) at 48 to 440 Hz. Power line voltage can vary by ±10%, but cannot exceed 250 Vac RMS. Maximum power consumption is 90 VA (Volt-Amps).

Nominal line voltages and corresponding limits are shown in Figure 2-4 which also shows the location of the line selector switches, fuse holder, HP-IB connector, and power cord jack (all on the rear panel). To set the AC line selector switches to the proper position for your AC line voltage, first ensure that the AC line cord is removed. Then, use a small flatblade screwdriver to move the switches to the proper positions as shown in Figure 2-4.

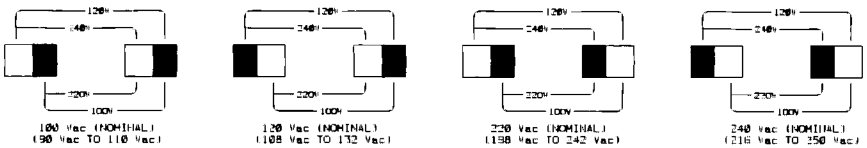
Install AC Line Fuse

When shipped, the HP 3245A does not have the AC line power fuse installed. Two line fuses and a fuse cap (packaged in a plastic bag) are provided with the HP 3245A. For 100 Vac or 120 Vac operation install the 1.5 A fuse and fuse cap. For 220 Vac or 240 Vac operation, install the 500 mAT fuse and fuse cap. To install a fuse, make sure the HP 3245A power cord is removed. Insert one end of the fuse into the fuse cap and then insert the fuse/cap assembly into the fuse holder. Using a small flatblade screwdriver, push in on the fuse cap and rotate it 1/4 turn clockwise to lock.

Connect HP-IB Cable

To connect your controller to the HP 3245A, attach an HP-IB cable from the controller to the HP-IB port on the HP 3245A rear panel. Finger tighten the two screws on the cable connector. Then, connect the other end of the HP-IB cable to your controller. A total of 15 devices can be connected on the same HP-IB bus. The cables have single male/female connectors on each end so that several cables can be stacked. However, the length of the HP-IB cables must not exceed 20 meters (65 feet) total or 2 meters (6.5 feet) per device, whichever is less.

① SET LINE SELECTOR SWITCHES

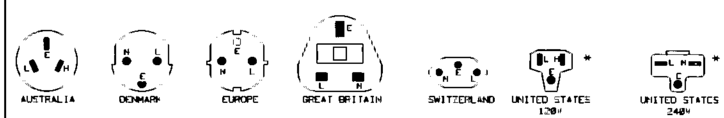


② INSTALL FUSE

LINE VOLTAGE	POWER LINE FUSE	FUSE CAP
100 or 120 Vac (Nominal)	1.5 AT - HP Part Number 2110-0043	Gray, HP Part Number 2110-0565
220 or 240 Vac (Nominal)	500 mAT - HP Part Number 2110-0202	

③ CONNECT POWER CORD

POWER CORDS



Country	Part Number	Opt	Voltage
Australia	8120-1369	901	250V 6A
Denmark	8120-2956	912	250V 6A
Europe	8120-1689	902	250V 6A
Great Britain	8120-1351	900	250V 6A
Switzerland	8120-2104	906	250V 6A
*United States	8120-1378	903	120V 10A
*United States	8120-0698	904	240V 10A

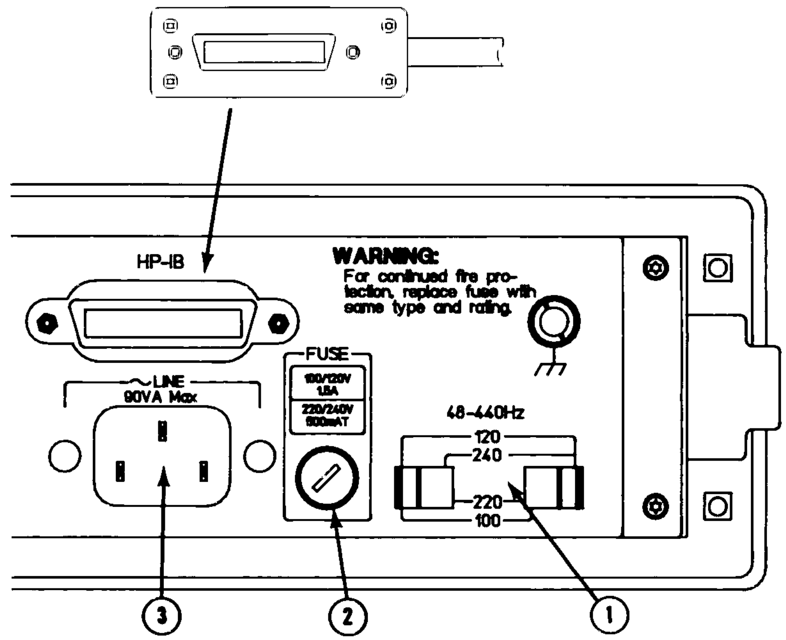
Power cords supplied by HP have polarities matched to the power input socket on the instrument

- L = Line or Active Conductor (also called "live" or "hot")
- N = Neutral or Identified Conductor
- E = Earth or Safety Ground

NOTE Plugs are viewed from connector end. Shape of molded plug may vary within country

* CSA certification includes only these Power Plugs

CONNECT HP-IB CABLE



HP 3245A REAR PANEL

Figure 2-4. Rear Panel Installation

Connect AC Power Cord

The HP 3245A is shipped with a three-conductor AC power cord (see Figure 2-4). This power cord must be connected to an approved three-contact electrical outlet which has its ground conductor connected to an electrical ground (safety ground).

The HP 3245A power jack and supplied power cords meet International Electrotechnical Commission (IEC) safety standards. If you received the wrong type of cord, please notify your nearest Hewlett-Packard Sales and Support Office for a replacement. When the proper fuse has been installed and the AC line selector switches have been set, connect the appropriate AC power cord to the AC line power connector.

Connecting BNC Cables

When the HP 3245A has been installed in the desired location and rear panel connections made, the next step is to connect BNC cables to the front and/or rear panel connectors as required. All BNC connectors on the HP 3245A are female connectors which accept standard (user-supplied) male BNC connectors. Figure 2-5 shows typical connections to the front and rear panel ports.

Connect Cables to Front Panel Ports

The front panel BNC connectors allow you to output voltage or current from either channel (Channel A and Channel B Output connectors), output a SYNC signal from either channel (Sync Out), or to accept or provide an external trigger signal to either channel (Trigger I/O connector).

You can connect a BNC cable from the Channel A or Channel B Output connector to the INPUT connector of an oscilloscope to monitor output waveforms or you can connect the cable to a high-resolution digital multimeter (DMM) such as the HP 3458A Multimeter (or equivalent). For triggering applications, you can also connect a BNC connector from the Channel A or Channel B Sync Out connector and/or from the Channel A or Channel B Trigger (I/O) connector.

Figure 2-5 (a) shows typical connections from the Channel A Output terminal to an oscilloscope, or to an HP 3458A DMM, or to a user application. When making connections to the HP 3458A, use a BNC-to-banana plug. Connect the banana plug to the HP 3458A HI and LO terminals for voltage outputs or to the I and LO terminals for current outputs.

Connect Cables to Rear Panel Ports

For some applications, you will need to connect BNC cables to the rear-panel ports. There are five rear panel BNC connectors (see Figure 2-5 (b). The Freq Ref (I/O) connector allows the HP 3245A to accept or provide an external reference frequency. The Channel A and Channel B Output connectors allow voltage or current outputs when rear panel output is selected. The TB0 and TB1 Trigger Bus connectors allow the HP 3245A to accept or provide an external trigger signal.

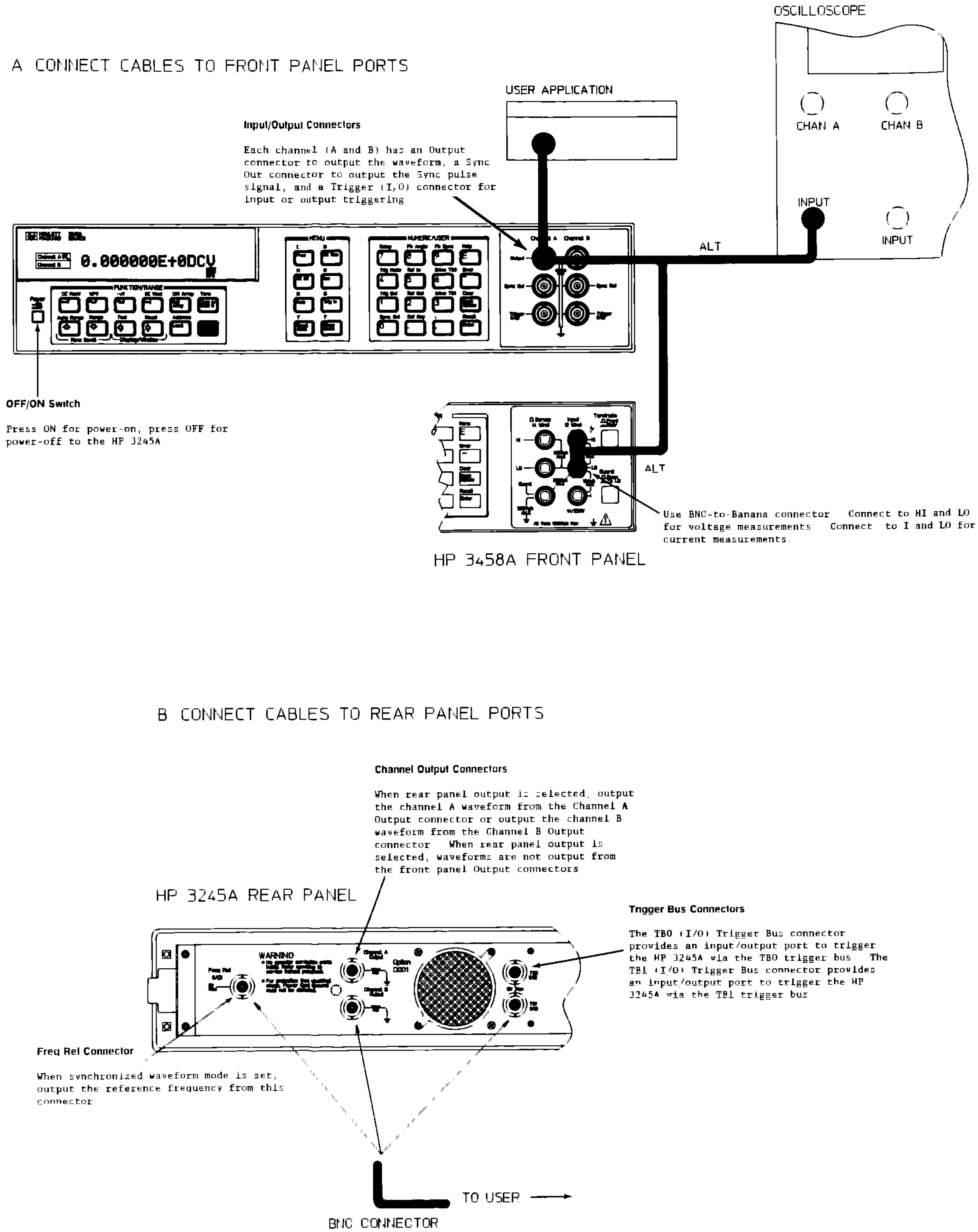


Figure 2-5. Typical Output Connections

All rear-panel BNC connectors are TTL-compatible. The Freq Ref, TB0, TB1, and front panel Trigger connectors can be set to act as inputs or outputs. Inputs on the TB0, TB1, and Trigger connectors are usable to 1.1 MHz with a minimum pulse width of 250 nsec. The input lock range for the Freq Ref connector is 1073741.824 Hz ±100 ppm. The output lock range for the Freq Ref connector is 1073741.824 Hz ±50 ppm.

Applying Power

When HP 3245A installation is complete, press the ON/OFF switch on the front panel (see Figure 2-2 for location) to ON to apply power to the HP 3245A The instrument will then cycle through its power-on sequence.

When the power-on sequence completes, the HP 3245A displays 0.000000E+0DCV on the front panel display and all display annunciators except MORE INFO are OFF. Also, the arrow on the display will point to Channel A showing that Channel A is the **USE** channel (the channel to receive commands). After power-on, the HP 3245A is operational and can be programmed via HP-IB or from the front panel.

NOTE

The front panel arrow points to the front-panel **USE** *channel. For an HP 3245A with two channels, one channel can be the* **USE** *channel for commands over HP-IB, while the other channel can be the* **USE** *channel for commands entered from the front panel. For example, channel A can be set as the* **USE** *channel for HP-IB operation and channel B as the* **USE** *channel for front panel operation.*

Front Panel Operation

This section shows how to use some front panel keys to generate outputs and to display data. It includes:

- Entering commands from the front panel.
- Displaying data on the front panel display.
- Example keystroke sequences.

NOTE

This section provides only an introduction to front panel operation. Refer to Chapter 6 - Front Panel Operation for detailed information on front panel operation.

Entering Commands

When power is applied and the power-on sequence completes, 0.000000E+0DCV is displayed on the front panel display and the MORE INFO annunciator is ON. To illustrate some ways to enter commands from the front panel, two example keystroke sequences follow. The first example generates a DC voltage output @ 1.2 VDC and the second generates a 1000 Hz AC sine waveform with 2.0 Vac PP amplitude. In the example sequences, the symbol ■ indicates a blinking cursor.

NOTE

If you make an error in entering keystrokes, or if you want to "start over", press the **Shift** *key followed by the* **Reset** *key to return the HP 3245A to a known state. All example sequences in this chapter assume a power-on state as the starting point.*

Example 2-1: DC Voltage Output From Front Panel

This keystroke sequence generates a DC voltage output @ 1.2 VDC from channel A. You may want to connect an HP 3458A DMM (or equivalent) to the Channel A Output connector (see Figure 2-5) so that you can verify the output value.

Key	Action	Display
⎓V	Set HP 3245A for DC voltage output.	APPLY DCV ■
1	Enters "1".	APPLY DCV 1■
.	Enters "."	APPLY DCV 1.■
2	Enters "2".	APPLY DCV 1.2■
Enter	Outputs 1.2 DCV	1.200000E+0DCV

Example 2-2: AC Sine Waveform From Front Panel

This keystroke sequence generates a 1000 Hz AC sine waveform from channel A with a peak-to-peak (PP) amplitude of 2.0 Vac. You may want to connect an oscilloscope to the Channel A Output connector (see Figure 2-5) so that you can observe the output.

Key	Action	Display
∿V	Sets HP 3245A to output AC sine waveform.	APPLY ACV ■
1	Sets output value to 1.0 Vac PP.	APPLY ACV 1■
Enter	Outputs 1.0 Vac PP sine waveform.	1.000000E+0ACV

Displaying Data

Commands which include a question mark (such as **ADDRESS?**) are called query commands. Query commands entered from the front panel return the result to the front panel display. To check initial operation of the front panel display, we will enter the **ADDRESS?** command which returns the HP-IB address of the HP 3245A to the display.

Example 2-3: Displaying HP-IB Address

This keystroke sequence queries the HP-IB address of the HP 3245A (see "HP-IB Programming" for a definition of the HP-IB address) and returns the address to the front panel display. Since the instrument is factory-set for address 09, "9" is shown on the display. The blue **Shift** key sets the front panel to shifted mode, so the function printed above any key is entered when the key is pressed. (In unshifted mode, the function printed on the key is entered.)

Key	Action	Display
	Shifts keyboard to shifted mode.	0.000000E+0DCV*
Address Local	Displays HP-IB address.	9
* = SHIFT annunciator is ON.		

HP-IB Programming

This section shows how to initially program the HP 3245A over the HP-IB interface It includes:

- Sending commands to the HP 3245A.
- Returning data to the controller.
- Example programs.

NOTE

This chapter includes only an introduction to HP-IB programming for the HP 3245A. Refer to Chapter 8 - Input/Output Operations for details on HP-IB operations.

Sending Commands

All programs in this manual show controller communication with an HP 3245A using BASIC language statements applicable to an HP 9000 Series 200/300 (or equivalent) controller. Commands are sent from the controller to the HP 3245A with an OUTPUT type statement and data is returned from the HP 3245A to the controller with an ENTER type statement.

Typical Controller OUTPUT Statement

Figure 2-6 shows a typical OUTPUT statement from a controller to send a command to the HP 3245A. In Figure 2-6, the word **OUTPUT** is a statement specific to HP 9000 Series 200/300 (and equivalent) controllers to send commands from the controller to the HP 3245A. The number 709 refers to the controller Interface Select Code (ISC) which is 7 in all program examples and the HP 3245A HP-IB address which is factory preset to 09.

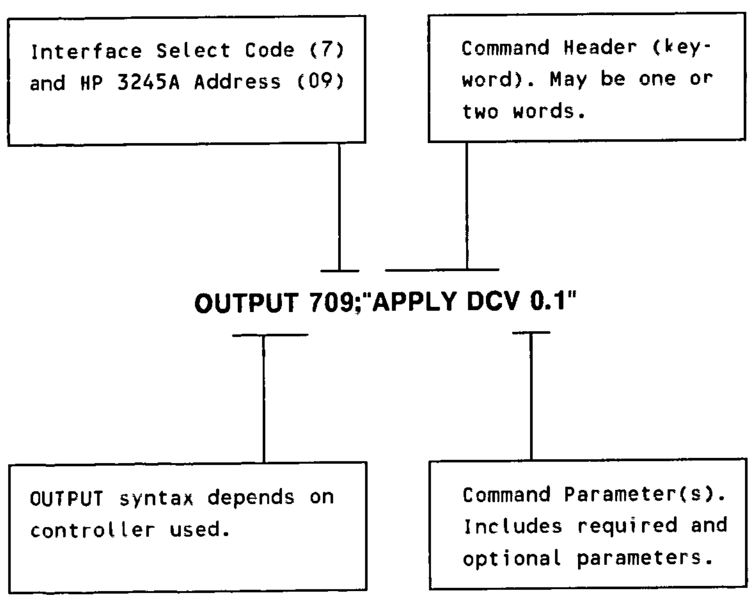


Figure 2-6. HP 3245A Command Format

HP 3245A Command Structure

HP 3245A commands consist of a command header (keyword) and zero, one, or two command parameters. The form is **KEYWORD** [*parameter* [,*parameter*]] where **KEYWORD** identifies the command function, and the parameter(s) identify the function values or settings.

When a parameter is enclosed in brackets ([ ]), the parameter is optional. A query command (a command which returns data) has the form **KEYWORD?** Either a comma or a space must separate command headers (keywords) from the first parameter and parameters must be separated by one or more spaces or one or more commas.

For example, the **ADDRESS** command has a keyword with no parameters, while **ADDRESS?** is a query command. **APPLY DCV** *volts* has a two-word keyword (**APPLY DCV**) and a single (required) parameter *volts*. **DELAY** [*time*] has a single (optional) parameter *time*. **APPLY WFV** *pp_amplitude* [,*array_name*] has both a required and an optional parameter.

As desired, multiple commands can be included in a single command string when the commands are separated by a semicolon (;). For example, the following program line generates a DC voltage output @ 1.5 DCV (using **APPLY DCV 1.5**) and then queries the output level (using **OUTPUT?**).

```
30  OUTPUT 709;"APPLY DCV 1.5;OUTPUT?"      !Output DCV, query value
```

HP 3245A Command Terminators

Valid command terminators are semicolons(;), carriage return (*cr*), and line feed (*lf*) or EOI asserted concurrent with the last character sent. The terminator initiates the execution of the command, so the next character in the command string is held off until the command finishes execution.

If the controller sends *cr lf* after each output string, sequential execution of the command string occurs. The HP 3245A executes commands in the sequence in which it receives them and finishes executing a command before executing the next command.

NOTE

If your controller does not use cr lf, refer to the **END** *command in the HP 3245A Command Reference Manual for details on asserting EOI.*

Returning Data

As noted, query commands (**ADDRESS?**, etc.) return data to the controller. To enter the data into the controller, use an ENTER 709;A type statement. Data returned by a query command from the controller is sent to the HP 3245A output buffer for entry into the controller via HP-IB. Data returned by a query command is sent to the front panel display when the command was initiated from the front panel.

Data can be returned to the controller in either ASCII or binary format, as shown in Table 2-2 (formats are IEEE-728 compatible). As desired, internal HP 3245A variables can be used in place of numeric parameters in all commands. Symbols not shown in Table 2-2 return an error message.

Table 2-2. HP 3245A Data Formats

Alphanumerics	Lower case is equivalent to upper case. User-defined HP 3245A variable names can be one to nine characters long. The first character must be a letter, but remaining characters can be either letters, numbers, the underline character (_), or a question mark (?).
Numbers	Numbers must be either integer, real, or scientific notation (+ddd.dddE+ddd).
Quoted Strings	Quoted strings include any characters surrounded by " or by '. Use either "" or '' to represent quotes within a command string. See the DISP command.
Mathematical Symbols	The symbols *,/,<,>,=,(,),+,-, and ^ are legal only within mathematical expressions or quoted strings.
Binary Output Format	Data can be output from the HP 3245A in ASCII or binary format. In binary format (with BLOCKOUT ON), the binary data format has the form #A, length byte 1, length byte 2, data bytes. See the OFORMAT command for examples.

Programming Examples

Three programming examples follow to illustrate HP 3245A programming over HP-IB. The first example uses **ADDRESS?** to return the HP 3245A HP-IB address; the second uses **USE?** to return the **USE** channel (the channel to receive output commands); and the third uses **OUTPUT?** to return the value of a DC voltage output.

To ensure that the HP 3245A is set to a known state, all example programs in this manual first clear the HP 3245A **CLEAR 709**, then reset the HP 3245A with **RST** and then delete all stored subroutines, variables, and arrays with **SCRATCH**.

NOTE

The HP 3245A is factory set for HP-IB address 09. If you need to change the address, refer to Chapter 6 - Front Panel Operation. All program listings must then be changed to reflect the new address.

Example 2-4: Reading HP-IB Address (ADDR2)

To program the HP 3245A over HP-IB, you must know the HP 3245A HP-IB address. You can read the HP-IB address with the **ADDRESS?** command. (The HP 3245A is factory-set for address 09). This program reads the HP-IB address of an HP 3245A and displays the result on the controller CRT.

```
10  !file ADDR2
20  !
30  CLEAR 709                    !Clear HP 3245A
40  OUTPUT 709;"RST"             !Reset HP 3245A
50  OUTPUT 709;"SCRATCH"         !Clear HP 3245A memory
60  OUTPUT 709;"ADDRESS?"        !Query HP-IB address
70  ENTER 709;A                  !Enter address
80  PRINT "Address =";A          !Display address
90  END
```

A typical return is:

```
Address = 9
```

Example 2-5: Reading Use Channel (USEQ2)

To execute some commands to the HP 3245A, you must specify the **USE** channel (the channel to receive the commands). If you have a one-channel version of the HP 3245A, the **USE** channel is always channel A (set with **USE 0** or **USE CHANA**). If you have a two-channel HP 3245A, the **USE** channel can be A or B. **USE 0** or **USE CHANA** sets channel A as the **USE** channel, while **USE 100** or **USE CHANB** sets channel B as the **USE** channel.

You can read the current **USE** channel by sending the **USE?** command. If channel A is the **USE** channel, "0" is returned. If channel B is the **USE** channel, "100" is returned. This program reads the current **USE** channel on the HP 3245A and displays the result (0) on the controller CRT to indicate that channel A is the **USE** channel.

```
10  !file USEQ2
20  !
30  CLEAR 709                    !Clear HP 3245A
40  OUTPUT 709;"RST"             !Reset HP 3245A
50  OUTPUT 709;"SCRATCH"         !Clear HP 3245A memory
60  OUTPUT 709;"USE?"            !Query use channel
70  ENTER 709;A                  !Enter result
80  PRINT "Use channel =";A      !Display result
90  END
```

Since **RST** sets channel A as the **USE** channel, a typical return is:

```
Use channel = 0
```

Example 2-6: Reading DC Output (DCVQ2)

A primary function of the HP 3245A is to output DC voltages. This program outputs 3.5 VDC from channel A and returns the output value to the controller CRT. You may want to connect an HP 3458A DMM (or equivalent) to the channel A output connector to verify that the output is 3.5 VDC.

```
10   !file DCVQ2
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"APPLY DCV 3.5"     !Output DC voltage @3.5 VDC
70   OUTPUT 709;"OUTPUT?"           !Query DC voltage output level
80   ENTER 709;A                    !Enter output level
90   PRINT "Output =";A;"VDC"       !Display output level
100  END
```

A typical return is:

```
Output = 3.5 VDC
```

Contents

Chapter 3
DC Voltage/Current Programs

Chapter 3

DC Voltage/Current Programs

Chapter Contents

This chapter shows several example programs to output DC voltages (DCV) or DC currents (DCI). The example programs are divided into three categories: power-on, modified, and triggered. Chapter contents are:

- **DC Outputs Overview** summarizes channel operation for DC voltage and current outputs.

- **DC Outputs Programs** provides several programs to generate DC voltage and current outputs. Programs are divided into three categories: power-on, modified, and triggered outputs.

- **Command Summary** shows a suggested command sequence to program DC outputs and summarizes commands which apply to DC voltages/current outputs.

NOTE

1. All programs in this chapter are stored on an Example Programs disc. (Refer to Chapter 1 - Using This Manual for details on using the disc.) Each program clears and resets the HP 3245A and clears the HP 3245A memory so that the instrument is in a known (power-on) state for program execution.

2. Table 3-2 at the end of this chapter lists commands which apply to defined waveforms and shows the power-on/default setting for each command. Refer to Chapter 10 - Programming DC Outputs for programming information on DC outputs and to the HP 3245A Command Reference Manual for command details. If you want to generate DC outputs using the HP 3245A front panel, refer to Chapter 6 - Front Panel Operation.

DC Outputs Overview

DC voltages or currents can be output from the use channel with an **APPLY DCV** or **APPLY DCI** command. For triggered DC outputs, you can use an **APPLY DCMEMV** or **APPLY DCMEMI** command. For DC voltage outputs, the range is -10.25 VDC to +10.25 VDC. For DC current outputs, the range is -0.1 A to +0.1 A. Figure 3-1 summarizes channel operation for DC outputs. If power-on conditions can be used, you only need to select the DC output function and level with an **APPLY DCV** *volts* or an **APPLY DCI** *amps* command. If power-on conditions need to be modified, you can change the channel parameters shown in Figure 3-1.

For triggered DC outputs, use **APPLY DCMEMV** or **APPLY DCMEMI**. With triggered outputs, 2 to 2048 output levels are defined by the user and stored in an array (defined with **DIM** or **REAL**) using **FILL** or the **Edit Array** key on the front panel. Then, when **APPLY DCMEMV** or **APPLY DCMEMI** is executed, the first value in the array is immediately output. The remaining values are output (one at a time) when a trigger from the source set by **TRIGIN** is received.

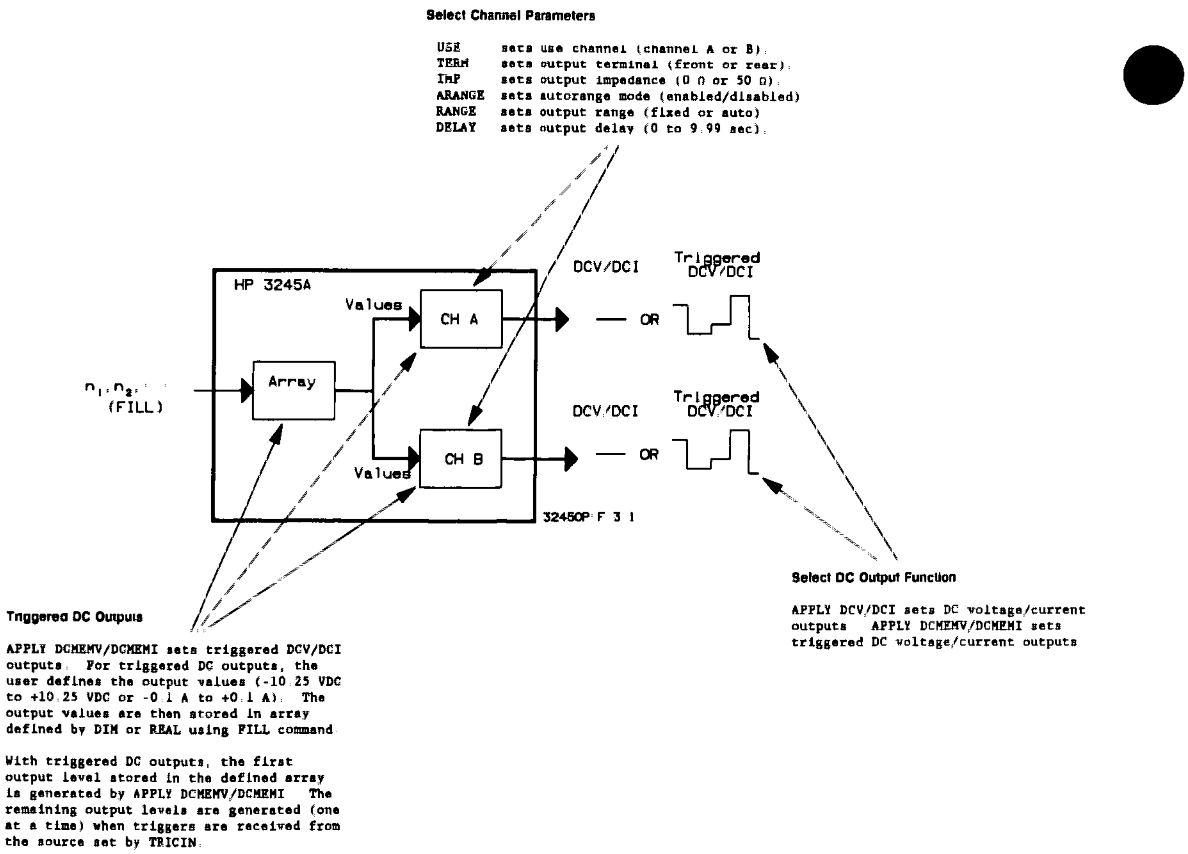


Figure 3-1. DC Outputs - Channel Operation

DC Outputs Programs

Example programs to output DC voltages and DC currents follow. All programs are stored on an Example Programs disc. The disc file name appears in parentheses in the program title.

Power-On DC Outputs Programs

With power-on/reset conditions (see Table 3-2), you can output DC voltages (constant voltage output) using **APPLY DCV** *volts*. Or, you can output DC currents (constant current output) with **APPLY DCI** *amps*. Two example programs for power-on DC outputs follow.

Example 3-1: Power-On DC Voltage Outputs (DCVP3)

This program outputs 3.5 VDC from channel A and returns the output function to the controller CRT. The HP 3245A is set for power-on/default conditions.

```
10   !file DCVP3
20   !
30   CLEAR 709                         !Clear HP 3245A
40   OUTPUT 709;"RST"                  !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"              !Clear HP 3245A memory
60   OUTPUT 709;"APPLY DCV 3.5"        !Output DC voltage @3.5 VDC
70   OUTPUT 709;"APPLY?"               !Query output function
80   ENTER 709;A$                      !Enter output function
90   PRINT "Function is";A$            !Display output function
100  END
```

A typical return is:

```
Function is DCV
```

Example 3-2: Power-On DC Current Outputs (DCIP3)

This program outputs 3.25 mA from channel A and returns the output value to the controller CRT. The HP 3245A is set for power-on/default conditions.

```
10   !file DCIP3
20   !
30   CLEAR 709                            !Clear HP 3245A
40   OUTPUT 709;"RST"                     !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                 !Clear HP 3245A memory
60   OUTPUT 709;"APPLY DCI 3.25E-3"       !Output DC current @ 3.25 mA
70   OUTPUT 709;"OUTPUT?"                 !Query output value
80   ENTER 709;A                          !Enter output value
90   PRINT "Output =";A;"Amps"            !Display output value
100  END
```

A typical return is:

```
Output = .00325 Amps
```

Modified DC Outputs Programs

As required, you can modify the power-on/default setting for your application. Table 3-1 lists output parameters which can be changed and shows power-on settings vs. alternate settings. Two example programs for modified DC outputs follow.

Table 3-1. DC Outputs - Parameter Changes

Function	Command	Power-On/ Default	Can Change Parameter to:
Use channel	USE	Channel A	Channel B
Output terminal	TERM	Front	Rear
Output impedance	IMP	0 Ohms	50 Ohms
Output delay	DELAY	0.04 sec	0 to 9.99 sec
DC resolution*	DCRES	High-res	Low-res
Autorange mode	ARANGE	Enabled	Disabled
Range	RANGE	Autorange	Fixed range

* = Low-resolution mode used for triggered DCV/DCI outputs and DCRES command has no effect.

Example 3-3: Modified DC Voltage Outputs (DCVM3)

This program outputs 3.5 VDC from channel B, with the output impedance set to 50 Ω and DC resolution set for low-resolution mode. The output level is returned to the controller CRT.

```
10   !file DCVM3
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"USE 100"           !Use channel B
70   OUTPUT 709;"IMP 50"            !Set 50 ohm output imped
80   OUTPUT 709;"DCRES LOW"         !Set low-res mode
90   OUTPUT 709;"APPLY DCV 3.5"     !Output DC voltage @3.5 VDC
100  OUTPUT 709;"OUTPUT?"           !Query DC voltage output level
110  ENTER 709;A                    !Enter output level
120  PRINT "Output =";A;"VDC"       !Display output level
130  END
```

A typical return (in Volts) is:

```
Output = 3.5 VDC
```

Example 3-4: Modified DC Current Outputs (DCIM3)

This program outputs 3.25 mA (0.00325 Amps) from channel A. Power-on/default conditions are set for the HP 3245A, except that the range is changed from autorange (set at power-on) to the 0.01A (fixed) range. The output value is returned to the controller CRT.

```
10   !file DCIM3
20   !
30   CLEAR 709                          !Clear HP 3245A
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60   OUTPUT 709;"APPLY DCI 3.25E-3"     !Output DC current @ 3.25 mA
70   OUTPUT 709;"RANGE 0.01"            !Set 0.01A range
80   OUTPUT 709;"OUTPUT?"               !Query output level
90   ENTER 709;A                        !Enter output level
100  PRINT "Output =";A;"Amps"          !Display output level
110  END
```

A typical return (in Amps DC) is:

```
Output = .00325 Amps
```

Triggered DC Outputs Programs

You can control the DCV or DCI output level by using external or internal triggering as set with the **TRIGIN** *event* command. To set the HP 3245A for triggered DC outputs, use **APPLY DCMEMV** for voltage outputs or use **APPLY DCMEMI** for current outputs With triggered outputs, you can change all output parameters shown in Table 3-1 except the DC resolution mode (triggered DC outputs use only low-resolution mode and **DCRES** has no effect). Two example programs for triggered DC outputs follow.

Example 3-5: Triggered DC Voltage Outputs (DCVT3)

This program outputs a triggered DC voltage sequence from channel A consisting of four voltages: -5.0, 0.0, 5.0, and +10.0 VDC. When the program executes, **APPLY DCMEMV** outputs the first value in the array (-5.0 VDC). The remaining voltages are output, one at a time, when **TRIGIN SGL** (a software trigger) is executed. Note that the first output (-5.0 VDC) is generated without a trigger.

```
10   !file DCVT3
20   !
30   CLEAR 709                        !Clear HP 3245A
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Clear HP 3245A memory
60   OUTPUT 709;"DIM VOUT(3)"         !Dimension 4-element array
70   OUTPUT 709;"FILL VOUT -5.0,0.0,5.0,10.0"  !Enter array values
80   OUTPUT 709;"APPLY DCMEMV 4,VOUT"  !Output = -5.0 V
90   OUTPUT 709;"TRIGIN SGL"          !Output =  0.0 V
```

```
100  OUTPUT 709;"TRIGIN SGL"              !Output =  5.0 V
110  OUTPUT 709;"TRIGIN SGL"              !Output = 10.0 V
120  OUTPUT 709;"TRIGIN SGL"              !Output = -5.0 V (wrap-around)
130  END
```

Example 3-6: Triggered DC Current Outputs (DCIT3)

This program outputs a triggered DC current sequence from channel A consisting of four current values: -0.05, +0.051, -0.052, and +0.053 Amps DC. When the program executes, **APPLY DCMEMI** outputs the first value in the array (-0.05 Amps DC). The remaining currents are output, one at a time, when **TRIGIN SGL** (a software trigger) is executed. Note that the first output (-0.05 Amps DC) is generated without a trigger.

```
10   !file DCIT3
20   !
30   CLEAR 709                             !Clear HP 3245A
40   OUTPUT 709;"RST"                      !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                  !Clear HP 3245A memory
60   OUTPUT 709;"DIM IOUT(3)"              !Define 4-element array
70   OUTPUT 709;"FILL IOUT -0.05,0.051,-0.052,0.053"    !Enter array values
80   OUTPUT 709;"APPLY DCMEMI 4,IOUT"      !Output = -0.05 A
90   OUTPUT 709;"TRIGIN SGL"               !Output = +0.051 A
100  OUTPUT 709;"TRIGIN SGL"               !Output = -0.052 A
110  OUTPUT 709;"TRIGIN SGL"               !Output = +0.053 A
120  OUTPUT 709;"TRIGIN SGL"               !Output = -0.05 A (wrap-around)
130  OUTPUT 709;"TRIGIN SGL"               !Output = +0.051 A
140  END
```

Command Summary

This section summarizes command and shows a suggested programming sequence to output DC voltages and currents. Figure 3-2 shows a typical sequence to program the use channel for power-on, modified, and triggered DC outputs and lists the related commands. Table 3-2 summarizes commands used to generate DC outputs. Refer to the HP 3245A Command Reference Manual for details on these commands.

Table 3-2. DC Outputs - Command Summary

Command	Description	Power-On/Default State
APPLY DCI amps	Output current	0 Amps DC
APPLY DCV volts	Output voltage	0 Volts DC
ARANGE [control]	Autorange mode	Autorange enabled
DCRES mode	DC resolution	High-resolution mode
DELAY [time]	Output delay	0.04 seconds
IMP mode	Output impedance	0 Ohms
RANGE [max_output]	Output range	Autorange (1 V)
TERM [mode]	Output terminal	Front
TRIGIN event	Triggering event	High
USE ch	Use channel	Channel A

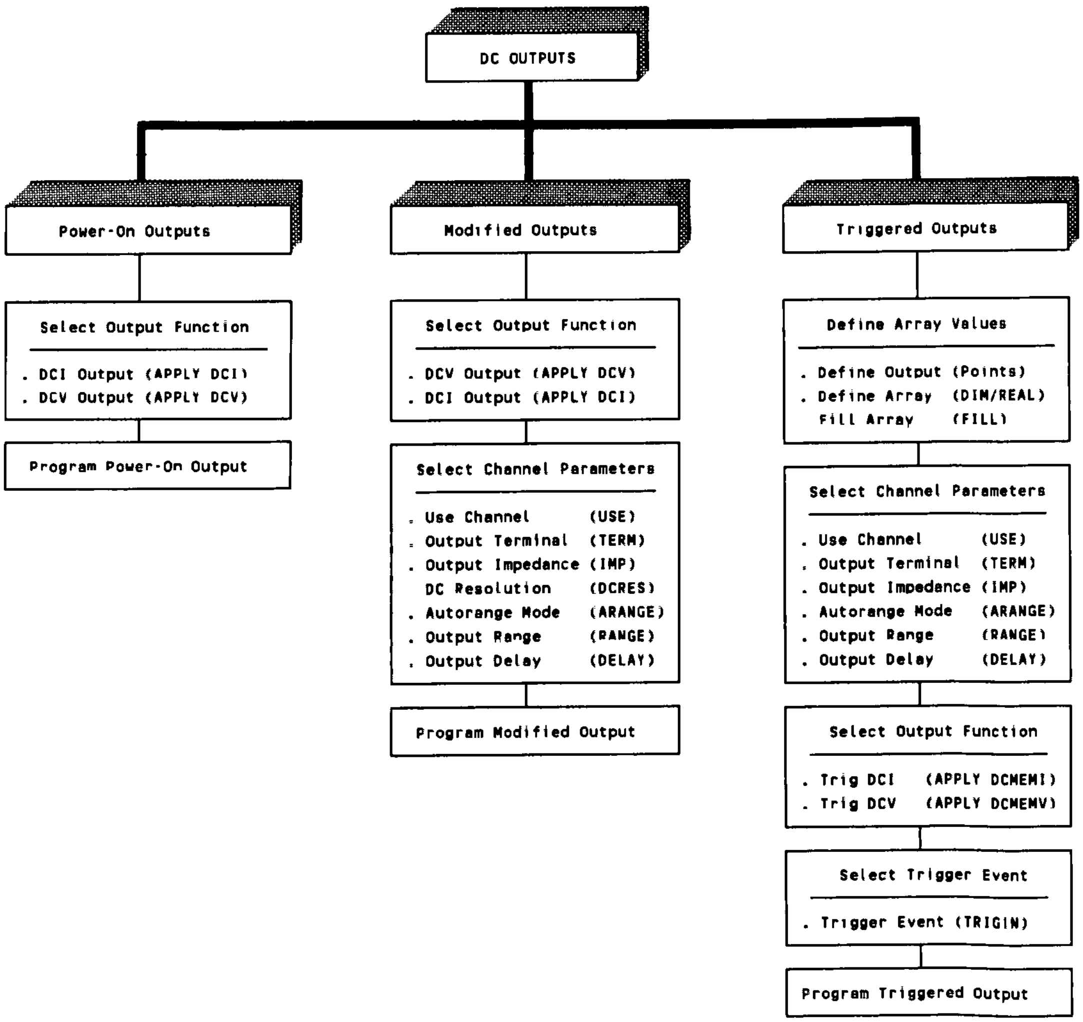


Figure 3-2. DC Outputs - Programming Steps

Contents

Chapter 4
Defined Waveforms Programs

Chapter 4
Defined Waveforms Programs

Chapter Contents

This chapter contains several example programs to generate defined (sine, ramp, and square wave) waveforms. The example programs are divided into three categories: power-on, modified, and triggered. Chapter contents are:

- **Defined Waveforms Overview** summarizes channel operation for defined AC waveform generation.

- **Defined Waveforms Programs** provides several programs to generate defined waveforms. Programs are divided into three categories: power-on waveforms, modified waveforms, and triggered waveforms.

- **Command Summary** shows a suggested command sequence to program defined waveforms and summarizes commands which apply to defined waveform generation.

NOTE

1. All programs in this chapter are stored on an Example Programs disc. Refer to Chapter 1 - Using This Manual for details on using the discs. Each program clears and resets the HP 3245A and clears the HP 3245A memory so that the instrument is in a known (power-on) state for program execution. Table 4-2 at the end of this chapter lists commands which apply to defined waveforms and shows the power-on/default setting for each command.

2. This chapter does not include arbitrary (user-defined) waveforms. Refer to Chapter 5 - Arbitrary Waveforms Programs for arbitrary waveform examples. Refer to Chapter 11 - Programming Defined Waveforms for programming information on defined AC waveforms and to the HP 3245A Command Reference Manual for command details. If you want to generate defined waveforms using the HP 3245A front panel, refer to Chapter 6 - Front Panel Operation.

Defined Waveforms Overview

A defined AC waveform is a sine, ramp, or square wave waveform which is output from the use channel with an **APPLY** command. Defined waveform voltage range is from 0.03125 Vac PP to 20 Vac PP with 0Ω output impedance, or from 0.015625 Vac PP to 10 Vac PP with 50Ω output impedance (when terminated in a 50 Ω load). The output current range is 0.00001 A PP to 0.2 A PP.

Figure 4-1 summarizes channel operation to generate defined waveforms. The user selects the waveform function (sine, ramp, or square wave) with the appropriate **APPLY** command. **APPLY ACV/ACI** selects sine wave voltage/current output, **APPLY RPV/RPI** selects ramp wave voltage/current output, and **APPLY SQV/SQI** selects square wave output.

If power-on conditions can be used, only an **APPLY** command is required to generate the output. As required, the user can specify one or more channel and/or waveform parameters and can select various trigger modes. When triggering is not used, when an **APPLY** command is executed, the specified waveform is immediately output. When triggering is used, the waveform is not generated until the specified trigger event occurs.

Defined Waveform Programs

Example programs follow to output defined waveforms. All programs are stored on an Example Programs disc and the file name appears in parentheses in the program title

Power-On Waveform Programs

With power-on/default conditions (refer to Table 4-2), only an **APPLY** command is required to output a sine, ramp, or square wave. You can output voltage sine waveforms (constant voltage) using **APPLY ACV** *volts* or output current sine waveforms (constant current) using **APPLY ACI** *amps*. Similarly, use **APPLY RPV** *volts* or **APPLY RPI** *amps* to output ramp waveforms or use **APPLY SQV** *volts* or **APPLY SQI** *amps* to output square waveforms.

The output voltage range for sine, ramp, and square waveforms is 20 Vac PP when the output impedance is 0 Ω or 10 Vac PP when the output impedance is 50 Ω (and the channel is terminated with a 50 Ω load). The output current range is 0.00001 Amps AC PP to 0.2 Amps AC PP for either impedance. At power-on, the duty cycle for ramp and square waveforms is 50% and output frequency for all waveforms is 1000 Hz. Example programs for power-on defined waveforms follow.

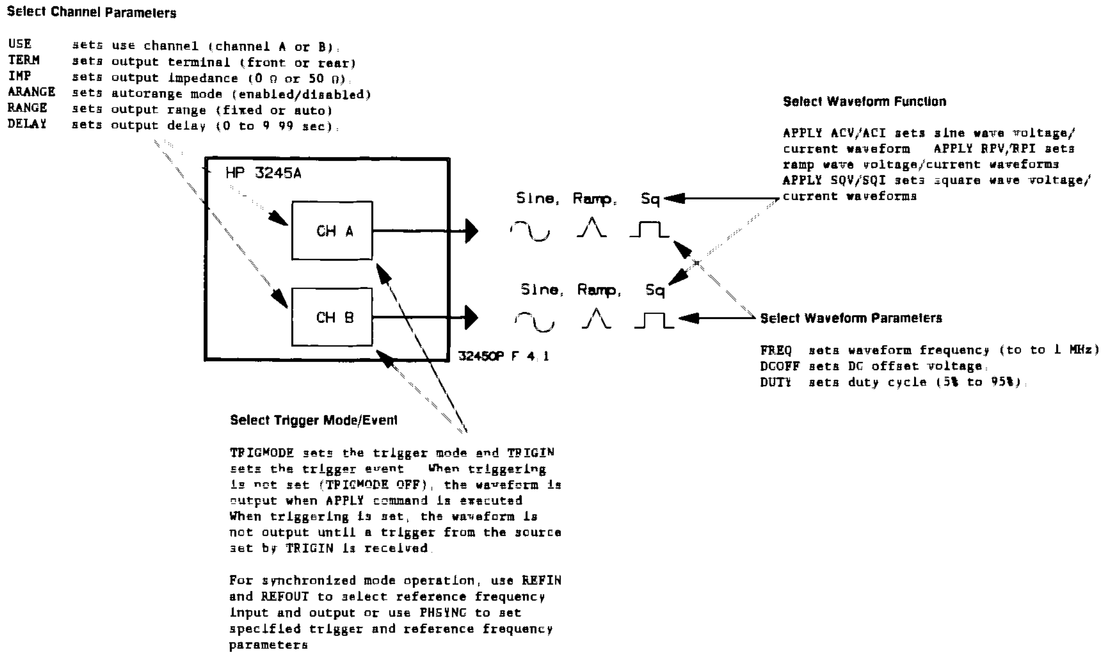


Figure 4-1. Defined Waveforms - Channel Operation

Example 4-1: Power-On Sine Waveform (SINPO4)

This program outputs a voltage sine wave @ 3.0 Vac PP and 1000 Hz from channel A. The output value is returned to the controller CRT. The HP 3245A is set for power-on/default conditions.

```
10   !file SINPO4
20   !
30   CLEAR 709                     !Clear HP 3245A
40   OUTPUT 709;"RST"              !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"          !Clear HP 3245A memory
60   OUTPUT 709;"APPLY ACV 3.0"    !Output sine wave @3.0 Vac PP
70   OUTPUT 709;"OUTPUT?"          !Query sine wave PP value
80   ENTER 709;A                   !Enter PP value
90   PRINT "Output =";A;"Vac PP"   !Display PP value
100  END
```

A typical controller CRT display (in Vac PP) follows. Figure 4-2 shows a typical oscilloscope display for the output. If you measure the output on an HP 3458A DMM, the result will be 1.0608 Vac RMS, since $V_{RMS} = V_{PP}/2.828$ for a sine wave.

```
Output = 3 Vac PP
```

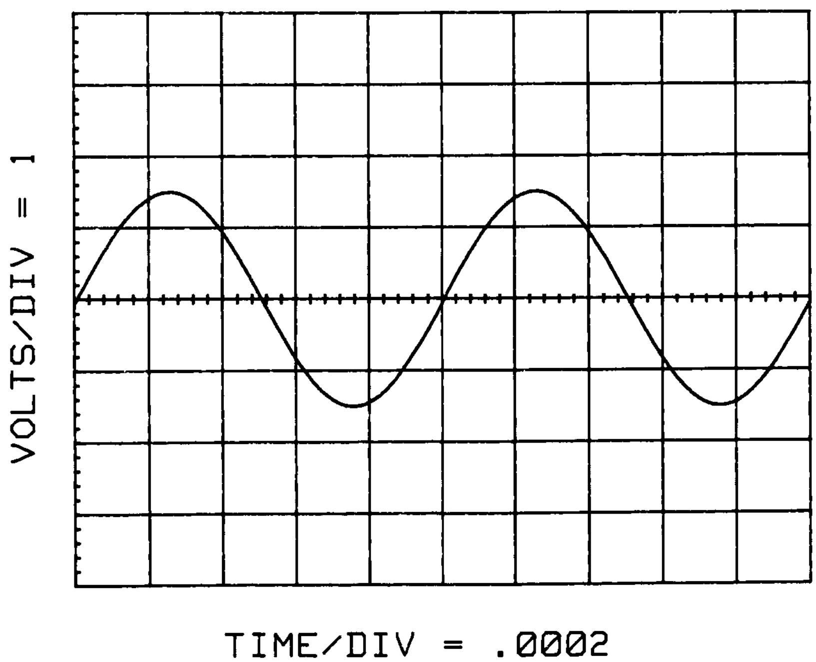


Figure 4-2. Example: Power-On Sine Waveform

Example 4-2: Power-On Square Waveform (SQRPO4)

This program outputs a 50% duty cycle voltage square waveform @ 3.0 Vac PP and 1000 Hz from channel A. The output value is returned to the controller CRT. The HP 3245A is set for power-on/default conditions (refer to Table 4-1). (To output a 3.0 Vac PP ramp waveform @ 1000 Hz and 50% duty cycle, use **APPLY RPV 3.0** in line 60.)

```
10   !file SQRPO4
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   OUTPUT 709;"APPLY SQV 3.0"      !Output square wave @3.0 Vac PP
70   OUTPUT 709;"OUTPUT?"            !Query square wave PP value
80   ENTER 709;A                     !Enter PP value
90   PRINT "Output =";A;"Vac PP"     !Display PP value
100  END
```

A typical controller CRT display (in Vac PP) follows. Figure 4-3 shows a typical waveform output. If you measure the output with an HP 3458A DMM, the result will be 1.5000 Vac RMS, since $V_{RMS} = V_{PP}/2.00$ for a 50% duty cycle square wave. (For a 50% duty cycle ramp waveform, the result is 3.000/3.464 = 0.8667 Vac RMS.)

```
Output = 3 Vac PP
```

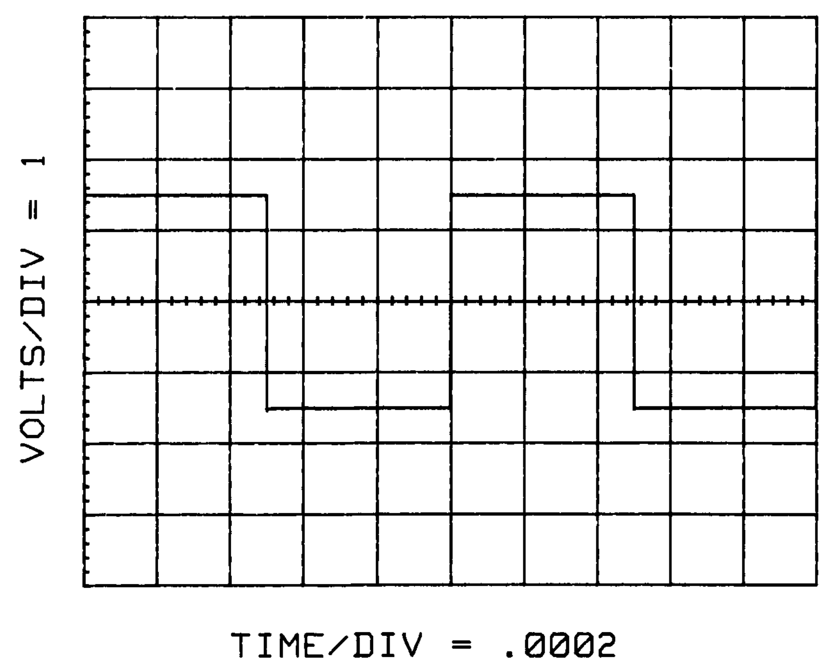


Figure 4-3. Example: Power-On Square Waveform

Modified Waveform Programs

As required, you can modify the power-on/default settings for channel and/or waveform parameters. Table 4-1 lists the channel and waveform parameters which can be changed and shows power-on settings vs. alternate settings. Example programs for modified waveform outputs follow.

Table 4-1. Defined Waveforms - Waveform Parameters

Channel/Waveform Function	Command	Power-On/ Default	Can Change Parameter to:
Use channel	USE	Channel A	Channel B
Output terminal	TERM	Front	Rear
Output impedance	IMP	0 Ohms	50 Ohms
Autorange mode	ARANGE	Enabled	Disabled
Output range	RANGE	Autorange	Fixed range
Output delay	DELAY	0.04 sec	0 to 9.99 sec
Output frequency	FREQ	1000 Hz	0 to 1 MHz*
DC offset	DCOFF	0 volts	50% of range**
Duty cycle	DUTY	50%***	5% to 95%****

*   = Ramp performance not specified above 100 kHz.
**   = Waveform plus offset magnitude ≤ 50% of PP range.
***   = Duty cycle applies to ramp and square waves only.
**** = For square waves, 50% duty cycle only above 100 kHz.

Example 4-3: Frequency-Sweep Sine Waveform (SWEEP4)

This program performs a 100 Hz to 100 kHz frequency-stepped sweep. The program outputs a 5 Vac PP sine waveform from channel A and runs through the frequency range in 100 Hz steps. The step rate is about 600 Hz. (Chapter 14 - Advanced Programming shows one way to increase the sweep speed to about 10 times the rate in this program.)

```
10   !file SWEEP4
20   !
30   CLEAR 709                            !Clear HP 3245A
40   OUTPUT 709;"RST"                     !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                 !Clear HP 3245A memory
60   OUTPUT 709;"SUB SWEEP"               !Define subroutine
70   OUTPUT 709;"  APPLY ACV 5"           !5 Vac PP sine wave
80   OUTPUT 709;"  FOR I = 100 to 100000 STEP 100"  !Sweep through range
90   OUTPUT 709;"    FREQ I"              !Change frequency
100  OUTPUT 709;"  NEXT I"                !Increment count
110  OUTPUT 709;"SUBEND"                  !End subroutine
120  OUTPUT 709;"CALL SWEEP"              !Execute subroutine
130  END
```

Example 4-4: Modified Ramp Waveform (RAMPM4)

This program outputs a voltage ramp waveform @ 3.0 Vac PP and 2000 Hz from channel A. The duty cycle is set for 30% and the DC offset for 1.0 Volts. The output value is returned to the controller CRT.

```
10    !file RAMPM4
20    !
30    CLEAR 709                         !Clear HP 3245A
40    OUTPUT 709;"RST"                  !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"              !Clear HP 3245A memory
60    OUTPUT 709;"FREQ 2000"            !2 kHz waveform frequency
70    OUTPUT 709;"DUTY 30"              !Set 30% duty cycle
80    OUTPUT 709;"DCOFF 1.0"            !Set DC offset to 1.0 V
90    OUTPUT 709;"APPLY RPV 3.0"        !Output ramp wave @3.0 Vac PP
100   OUTPUT 709;"OUTPUT?"              !Query ramp wave PP value
110   ENTER 709;A                       !Enter PP value
120   PRINT "Output =";A;"Vac PP"       !Display PP value
130   END
```

A typical controller CRT display (in Vac PP) follows. Figure 4-4 shows a typical waveform output (the scope is set for DC position). If you measure the output on an HP 3458A DMM, the result will be 0.866 Vac RMS, since $V_{RMS} = V_{PP}/3.464$ for a ramp waveform.

```
Output = 3 Vac PP
```

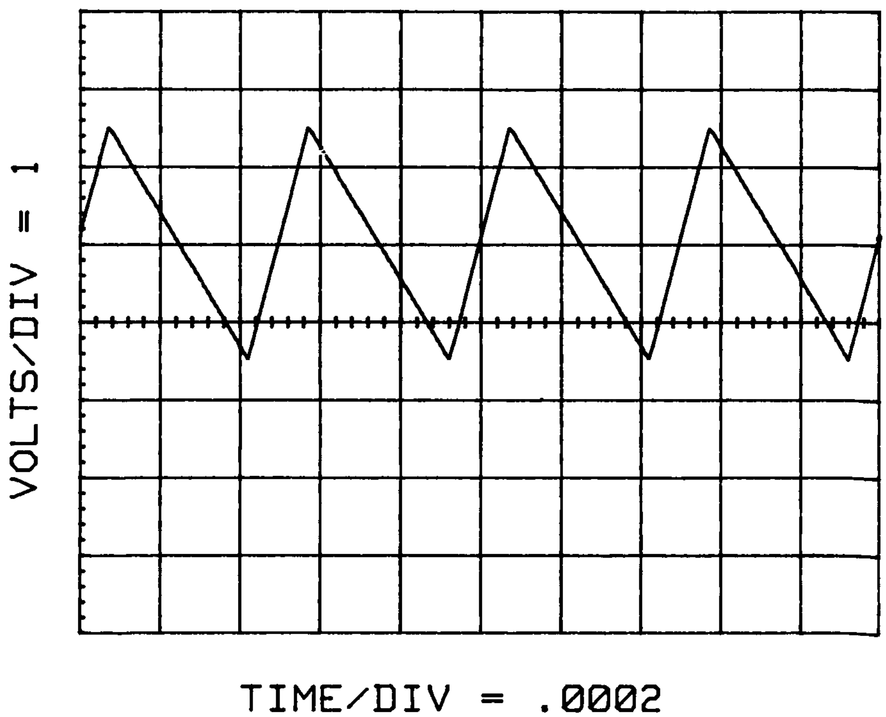


Figure 4-4. Example: Modified Ramp Waveform

Example 4-5: Modified Square Waveform (SQUAR4)

This program outputs a 0.005 A (5 mA) current square waveform @ 1000 Hz and 50% duty cycle from channel A. For this program, the output impedance is changed from 0Ω to 50 Ω. The output value is returned to the controller CRT.

```
10   !file SQUAR4
20   !
30   CLEAR 709                     !Clear HP 3245A
40   OUTPUT 709;"RST"              !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"          !Clear HP 3245A memory
60   OUTPUT 709;"IMP 50"           !Set 50 ohm output imped
70   OUTPUT 709;"APPLY SQI 5E-3"   !Set output to 5 mA PP
80   OUTPUT 709;"OUTPUT?"          !Query AC output value
90   ENTER 709;A                   !Enter output value
100  PRINT "Output =";A;"Amps"     !Display output value
110  END
```

A typical controller CRT display (in Amps AC) follows. If the output is measured with an HP 3458A DMM, the result (since this is a 50% duty cycle square wave) is .005 A/2.000 = 0.0025 A (2.5 mA).

```
Output = .005 Amps
```

Triggered Waveform Programs

You can use triggering to control AC waveform output generation. The trigger mode is set with the **TRIGMODE** *mode* command. At power-on, **TRIGMODE OFF** is set which allows waveforms to be output immediately when the **APPLY** command is executed. (**TRIGMODE** has no effect on DCV or DCI outputs.) However, when **TRIGMODE OFF** is not set, the output is not generated until a trigger from the source set by **TRIGIN** *event* occurs.

The **TRIGMODE** command has three parameters to set trigger modes. **TRIGMODE ARMWF** sets armed (synchronized) mode, **TRIGMODE GATED** sets gated mode, and **TRIGMODE DUALFR** sets dual-frequency mode. For any triggered mode, **TRIGIN** *event* sets the event which will generate the trigger. As with modified waveforms, you can change channel and waveform parameters for your specific application (refer to Table 4-2). Example programs to generate triggered waveforms follow.

Example 4-6: Triggered Sine Waveform (TRIGS4)

This program outputs a 2 kHz voltage sine waveform @ 3.0 Vac PP from channel A with 1.0 Volts DC offset. Since **TRIGMODE ARMWF** is set and **WAIT 2** (line 110) sets a 2-second delay before a (software) trigger is generated by **TRIGIN SGL**, the output is delayed 2 seconds after **APPLY ACV** is executed.

```
10   !file TRIGS4
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE ARMWF"    !Set armed trigger mode
70   OUTPUT 709;"FREQ 2000"         !2 kHz waveform frequency
80   OUTPUT 709;"DCOFF 1.0"         !Set DC offset of 1.0V
90   OUTPUT 709;"APPLY ACV 3.0"     !Output sine wave @3.0 Vac PP
100  OUTPUT 709;"WAIT 2"            !Wait 2 seconds
110  OUTPUT 709;"TRIGIN SGL"        !Single-trigger the HP 3245A
120  OUTPUT 709;"OUTPUT?"           !Query sine wave PP value
130  ENTER 709;A                    !Enter PP value
140  PRINT "Output =";A;"Vac PP"    !Display PP value
150  END
```

A typical controller CRT display (in Vac PP) follows. Figure 4-5 shows a typical waveform output. If you measure the output on an HP 3458A DMM, the result will be 1.0608 Vac RMS, since $V_{RMS} = V_{PP}/2.828$ for a sine waveform.

```
Output = 3 Vac PP
```

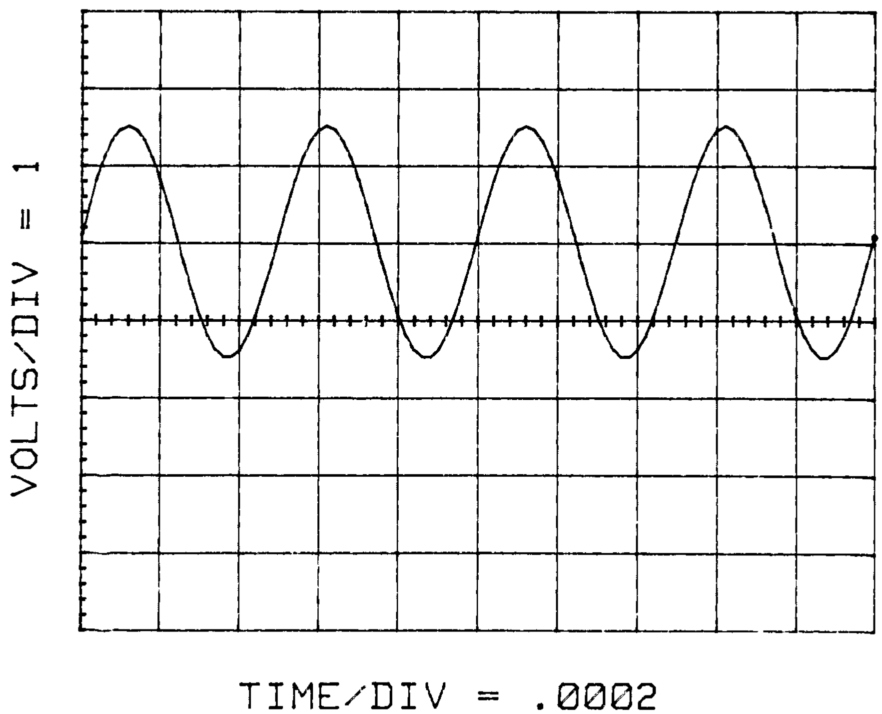


Figure 4-5. Example: Triggered Sine Waveform

Example 4-7: Triggered Ramp Waveform (TRIGR4)

This program outputs a voltage ramp waveform @ 3.0 Vac PP and 2000 Hz from channel A. The duty cycle is set for 30% and the DC offset for 1.0 Volts. The waveform output does not begin until an external (high-to-low) trigger is input to the channel A Trigger port. The program requires a BNC connector to the channel A Trigger connector and a TTL-compatible (user-supplied) input trigger. Note that **TRIGMODE ARMWF** sets channel A to the armed mode, so the waveform is not output until the external trigger is received.

```
10   !file TRIGR4
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE ARMWF"    !Set trigger armed mode
70   OUTPUT 709;"FREQ 2000"         !Set output freq to 2000 Hz
80   OUTPUT 709;"DUTY 30"           !Set 30% duty cycle
90   OUTPUT 709;"DCOFF 1.0"         !Set DC offset of 1.0V
100  OUTPUT 709;"APPLY RPV 3.0"     !Output ramp wave @3.0 Vac PP
110  OUTPUT 709;"TRIGIN EXT"        !External trigger channel A
120  END
```

Figure 4-6 shows a typical waveform output. If you measure the output on an HP 3458A DMM, the result will be 0.866 Vac RMS, since $V_{RMS} = V_{PP}/3.464$ for a ramp waveform.

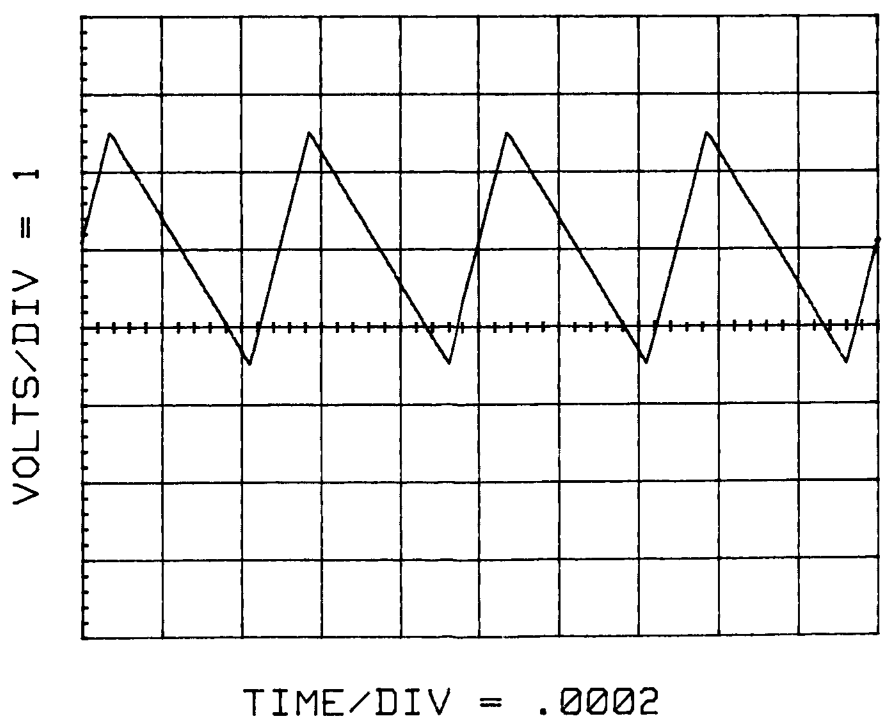


Figure 4-6. Example: Triggered Ramp Waveform

Example 4-8: Gated Ramp Waveform (GATER4)

This program enables gated waveform mode on the channel A. When the trigger signal to the channel A Trigger (I/O) connector is low (0 V), the channel outputs a 2 kHz ramp waveform with a 20% duty cycle. When the trigger input is high (+5 V), the ramp waveform continues until it reaches its zero phase point and then stops. For this program, connect an external trigger source to the channel A front panel Trigger (I/O) connector.

```
10   !file GATER4
20   !
30   CLEAR 709                         !Clear HP 3245A
40   OUTPUT 709;"RST"                  !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"              !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE GATE"        !Set gated mode
70   OUTPUT 709;"TRIGIN EXT"           !External trigger source
80   OUTPUT 709;"FREQ 2000"            !2 kHz waveform frequency
90   OUTPUT 709;"DUTY 20"              !20% duty cycle
100  OUTPUT 709;"APPLY RPV 5"          !Output ramp waveform @ 5 Vac PP
110  END
```

Figure 4-7 shows a typical waveform output with an approximate 125 Hz trigger rate. If you measure the output on an HP 3458A DMM, the result will be 1.443 Vac RMS, since $V_{RMS} = V_{PP}/3.464$ for a ramp waveform.

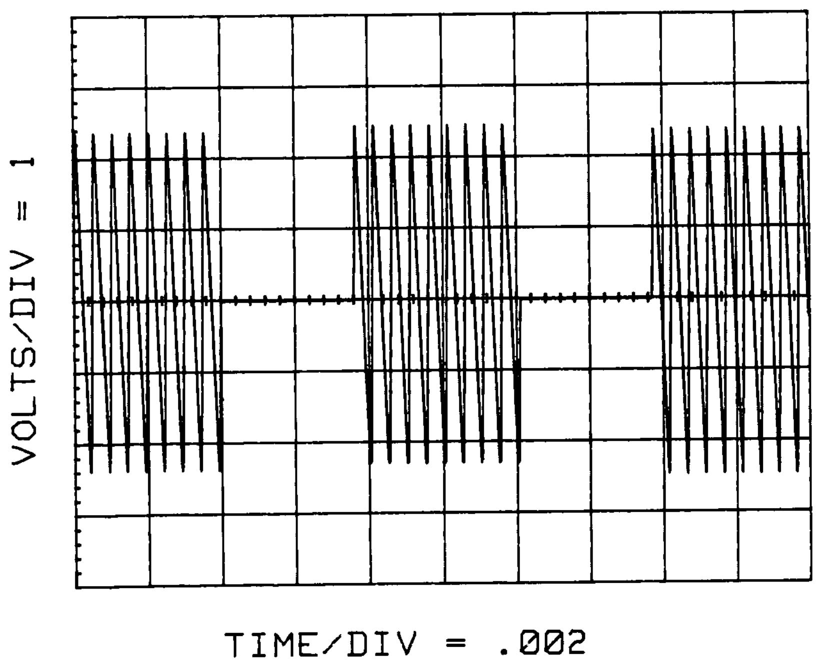


Figure 4-7. Example: Gated Ramp Waveform

Example 4-9: Multi-Period Gated Sine Waveform (GATES4)

You can also use **TRIGMODE GATE** to generate a multi-period burst of a waveform by externally programming the gate pulse width to be (N + 1/2) periods long, where N is an integer number of periods. This program generates a 3.5 msec trigger input. Since this is 3.5 periods of a 1 kHz waveform (1 period = 1 msec), the program produces a four-period burst of the sine waveform.

```
10   !file GATES4
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE GATE"     !Set gated mode
70   OUTPUT 709;"FREQ 1000"         !1 kHz waveform frequency
80   OUTPUT 709;"APPLY ACV 5"       !Output sine wave @ 5 Vac PP
90   OUTPUT 709;"SUB BURST"         !Begin subroutine
100  OUTPUT 709;"  TRIGIN LOW"      !Set trigger level low (0 V)
110  OUTPUT 709;"  WAIT 3.5E-3"     !Wait 3.5 msec
120  OUTPUT 709;"  TRIGIN HIGH"     !Set trigger level high (+5 V)
130  OUTPUT 709;"SUBEND"            !End subroutine
140  OUTPUT 709;"CALL BURST"        !Run subroutine
150  END
```

Example 4-10: Dual-Frequency Sine Waveform (DUALS4)

This program enables channel A to output a dual-frequency sine waveform with an amplitude of 5 Vac PP. The output frequency is 50 Hz when the input to the channel A Trigger (I/O) connector is high (+5 V) and 500 Hz when the signal is low (0 V). For this program, connect an external trigger source to the channel A Trigger (I/O) connector.

```
10   !file DUALS4
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE DUALFR"   !Set dual-frequency mode
70   OUTPUT 709;"FREQ 50,500"       !Output 50 Hz or 500 Hz
80   OUTPUT 709;"TRIGIN EXT"        !External trigger source
90   OUTPUT 709;"APPLY ACV 5"       !Output sine wave @ 5 Vac PP
100  END
```

Example 4-11: Sync Waveform Using PHSYNC (SYNCP4)

This program outputs a 5 kHz sine waveform (phase = 0°) from the channel A (the master channel) and a 5 kHz ramp waveform (phase = 180°) from channel B (the slave channel). **PHSYNC** sets synchronized waveform mode, sets TB1 as the channel input trigger sources, sets the FREQ REF connector as the output/input path for the reference frequency, and simultaneously triggers both channels (via TB1) to output their waveforms. See Example 4-12 for an alternate approach.

```
10   !file SYNCP4
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   !
70   !Configure channel A
80   !
90   OUTPUT 709;"USE 0"             !Use channel A
100  OUTPUT 709;"FREQ 5000"         !Output freq is 5 kHz
110  OUTPUT 709;"APPLY ACV 5"       !Output sine wave @ 5 Vac PP
120  !
130  !Configure channel B
140  !
150  OUTPUT 709;"USE 100"           !Use channel B
160  OUTPUT 709;"FREQ 5E3"          !Output freq is 5 kHz
170  OUTPUT 709;"PANG 180"          !Phase angle = 180 degrees
180  OUTPUT 709;"APPLY RPV 5"       !Output ramp wave @ 5 Vac PP
190  !
200  !Generate synchronized outputs
210  !
220  OUTPUT 709;"PHSYNC"            !Generate sync outputs
230  END
```

Example 4-12: Sync Waveform Using REFIN/REFOUT (SYNCR4)

This program outputs a 5 kHz sine waveform (phase = 0°) from channel A (the master channel) and a 5 kHz ramp waveform (phase = 180°) from channel B (the slave channel). In this program, the master channel is triggered using the **TRIGIN** command HIGH and LOW parameters (software triggering).

With **TRIGOUT EXT** set, Channel A routes its trigger signal to the channel A Trigger (I/O) port. To trigger channel B, a BNC connector must be connected between the channel A and channel B Trigger (I/O) ports. Channel A and channel B are simultaneously triggered when **TRIGIN LOW** is executed. With **REFOUT EXT** and **REFIN EXT**, the channel A reference frequency is sensed by channel B and external connections are not required for the reference frequency.

```
10  !file SYNCR4
20  !
30  CLEAR 709                         !Clear HP 3245A
40  OUTPUT 709;"RST"                  !Reset HP 3245A
50  OUTPUT 709;"SCRATCH"              !Clear HP 3245A memory
60  !
70  !Configure channel A (master)
80  !
90  OUTPUT 709;"USE 0"                !Use channel A
100 OUTPUT 709;"TRIGIN HIGH"          !Set trigger level high (+5V)
110 OUTPUT 709;"REFOUT EXT"           !REF FREQ connector is dest
120 OUTPUT 709;"TRIGMODE ARMWF"       !Set sync mode on channel A
130 OUTPUT 709;"TRIGOUT EXT"          !Enable ch A Trigger output
140 OUTPUT 709;"FREQ 5000"            !Output freq is 5 kHz
150 OUTPUT 709;"PANG 0"               !Phase angle = 0 degrees
160 OUTPUT 709;"APPLY ACV 5"          !Output sine wave @ 5 Vac PP
170 !
180 !Configure channel B (slave)
190 !
200 OUTPUT 709;"USE 100"              !Use channel B
210 OUTPUT 709;"REFIN EXT"            !REF FREQ connector is source
220 OUTPUT 709;"TRIGMODE ARMWF"       !Set sync mode on channel B
230 OUTPUT 709;"TRIGIN EXT"           !Trigger conn is trigger source
240 OUTPUT 709;"FREQ 5E3"             !Output freq is 5 kHz
250 OUTPUT 709;"PANG 180"             !Phase angle = 180 degrees
260 OUTPUT 709;"APPLY RPV 5"          !Output ramp wave @ 5 Vac PP
270 !
280 !Generate synchronized outputs
290 !
300 OUTPUT 709;"USE 0"                !Use channel A
310 OUTPUT 709;"TRIGIN LOW"           !High-to-low trigger
320 END
```

Command Summary

This section summarizes commands and shows a suggested programming sequence to generate defined waveforms. Figure 4-8 shows a typical sequence to program the use channel for power-on, modified, and triggered defined waveforms and lists the related commands. Table 4-2 summarizes commands used to generate defined waveforms. Refer to the HP 3245A Command Reference Manual for command details.

Table 4-2. Defined Waveforms - Command Summary

Command	Description	Power-On
Waveform Function		
APPLY ACI amps	Sine wave (current)	0 Amps
APPLY ACV volts	Sine wave (voltage)	0 Volts
APPLY RPI amps	Ramp wave (current)	0 Amps
APPLY RPV volts	Ramp wave (voltage)	0 Volts
APPLY SQI amps	Square wave (current)	0 Amps
APPLY SQV volts	Square wave (voltage)	0 Volts
Channel Parameters		
USE ch	Use channel	Chan A
TERM [mode]	Output terminal	Front
IMP mode	Output impedance	0 Ohms
ARANGE [control]	Autorange mode	ON
RANGE [max_output]	Output range	Auto (1V)
DELAY [time]	Output delay	0.04 sec
Waveform Parameters		
FREQ freq1	Waveform frequency	1000 Hz
DCOFF volts	DC offset	0 Volts
DUTY %_duty	Duty Cycle	50%
Trigger Event/Mode		
TRIGIN event	Trigger event	High
TRIGMODE mode	Trigger mode	Off
REFIN source	Ref freq in source	Int
REFOUT	Ref freq out dest	Ext
PHSYNC	Synchronized mode	---

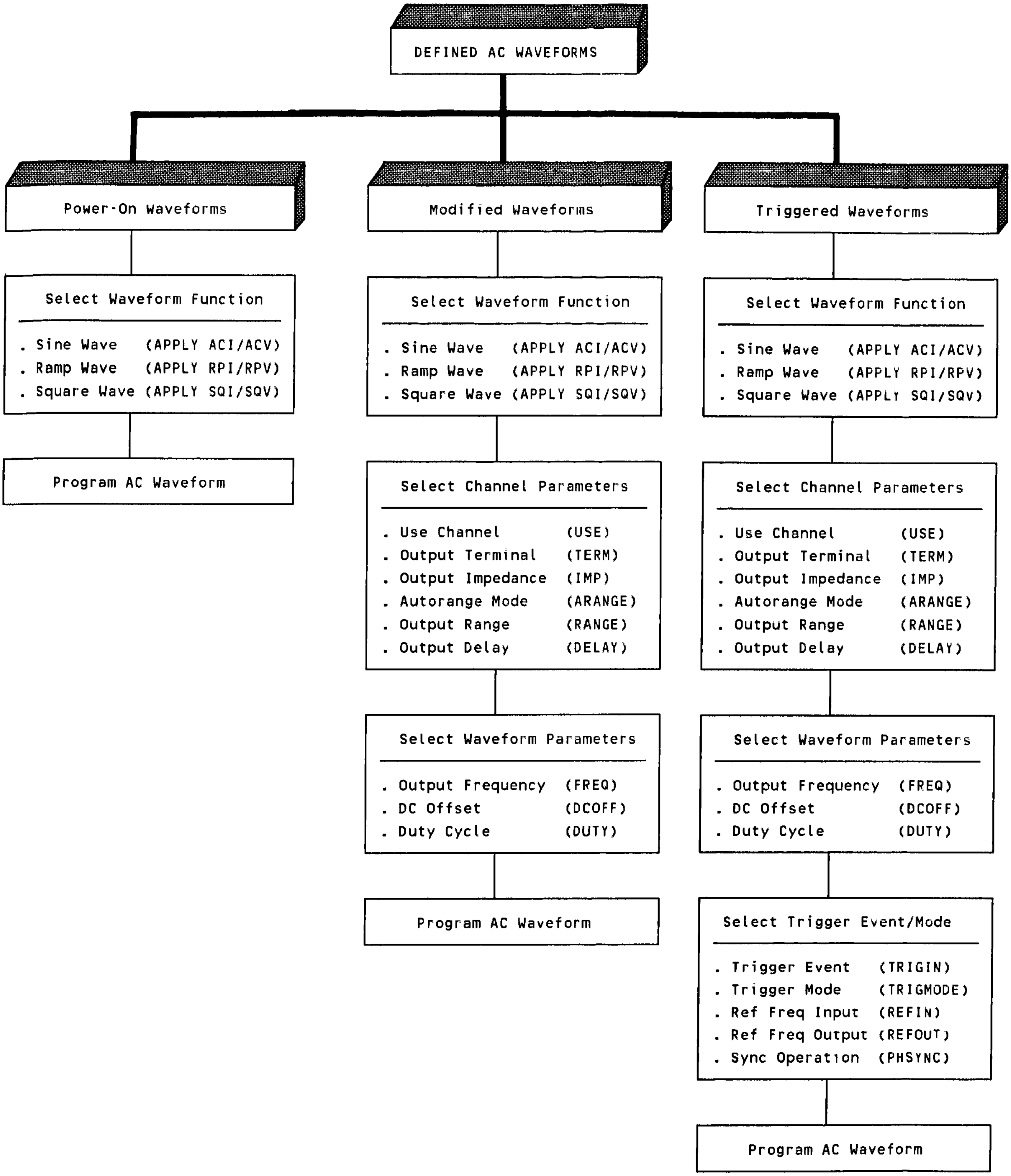


Figure 4-8. Defined Waveforms - Programming Steps

Contents

Chapter 5
Arbitrary Waveforms Programs

Chapter 5
Arbitrary Waveforms Programs

Chapter Contents

This chapter contains several example programs to output arbitrary (user-defined) waveforms. The example programs are divided into three categories: user-defined, precomputed, and triggered. Chapter contents are:

- **Arbitrary Wavforms Overview** summarizes channel operation for arbitrary waveform generation.

- **Arbitrary Waveform Programs** provides several example programs to generate arbitrary waveforms. Programs are divided into three categories: user-defined, precomputed, and triggered.

- **Command Summary** shows a suggested command sequence to program arbitrary waveforms and summarizes commands which apply to arbitrary waveform generation.

NOTE

1. All programs in this chapter are stored on an Example Programs disc. Refer to Chapter 1 - Using This Manual for details on using the disc. Each program clears and resets the HP 3245A and clears the HP 3245A memory so that the instrument is in a known (power-on) state for program execution. Table 5-1 at the end of this chapter lists commands which apply to defined waveforms and shows the power-on/default setting for each command.

2. Refer to Chapter 12 - Programming Arbitrary Waveforms for programming information on arbitrary AC waveforms. Refer to the HP 3245A Command Reference Manual for command details. If you want to generate arbitrary waveforms using the HP 3245A front panel, refer to Chapter 6 - Front Panel Operation.

Arbitrary Waveforms Overview

An arbitrary waveform is defined as a waveform in which each waveform point is defined by the user. Arbitrary waveforms must contain 2048 points and are stored in an HP 3245A REAL or INTEGER array. Each point within a REAL array must have a value between -1 and +1, inclusive.

Arbitrary waveform voltage range is from 0.03125 Vac PP to 20 Vac PP with 0Ω output impedance, or from 0.015625 Vac PP to 10 Vac PP with 50Ω output impedance (when terminated in a 50 Ω load). Arbitrary waveform current range is 0.0001 Amps AC PP to 0.2 Amps AC PP for either output impedance. Each waveform point (array element) is multiplied by the peak value of the peak-to-peak amplitude specified.

Figure 5-1 summarizes channel operation to generate arbitrary waveforms. The user specifies the shape of an arbitrary waveform mathematically, such as f(t) = sin(x)/x or by identifying 2048 points on the waveform (p1 = 0.01, p2 = -0.4, etc.). When the waveform is identified, the 2048 points are entered into a defined array.

As required, the user can specify one or more channel and/or waveform parameters and can select various trigger modes. If triggering is not used, when an **APPLY WFV** or **APPLY WFI** command is executed, the specified waveform is immediately output. When triggering is used, the waveform is not generated until the specified trigger event occurs.

Arbitrary Waveform Programs

Example programs to generate arbitrary waveforms follow. All programs are stored on an Example Programs disc and the file name appears in parentheses in the program title.

User-Defined Waveform Programs

User-defined arbitrary waveforms are arbitrary waveforms in which precomputing (to reduce the time required for output) and triggering are not used Example programs to generate user-defined arbitrary waveforms follow

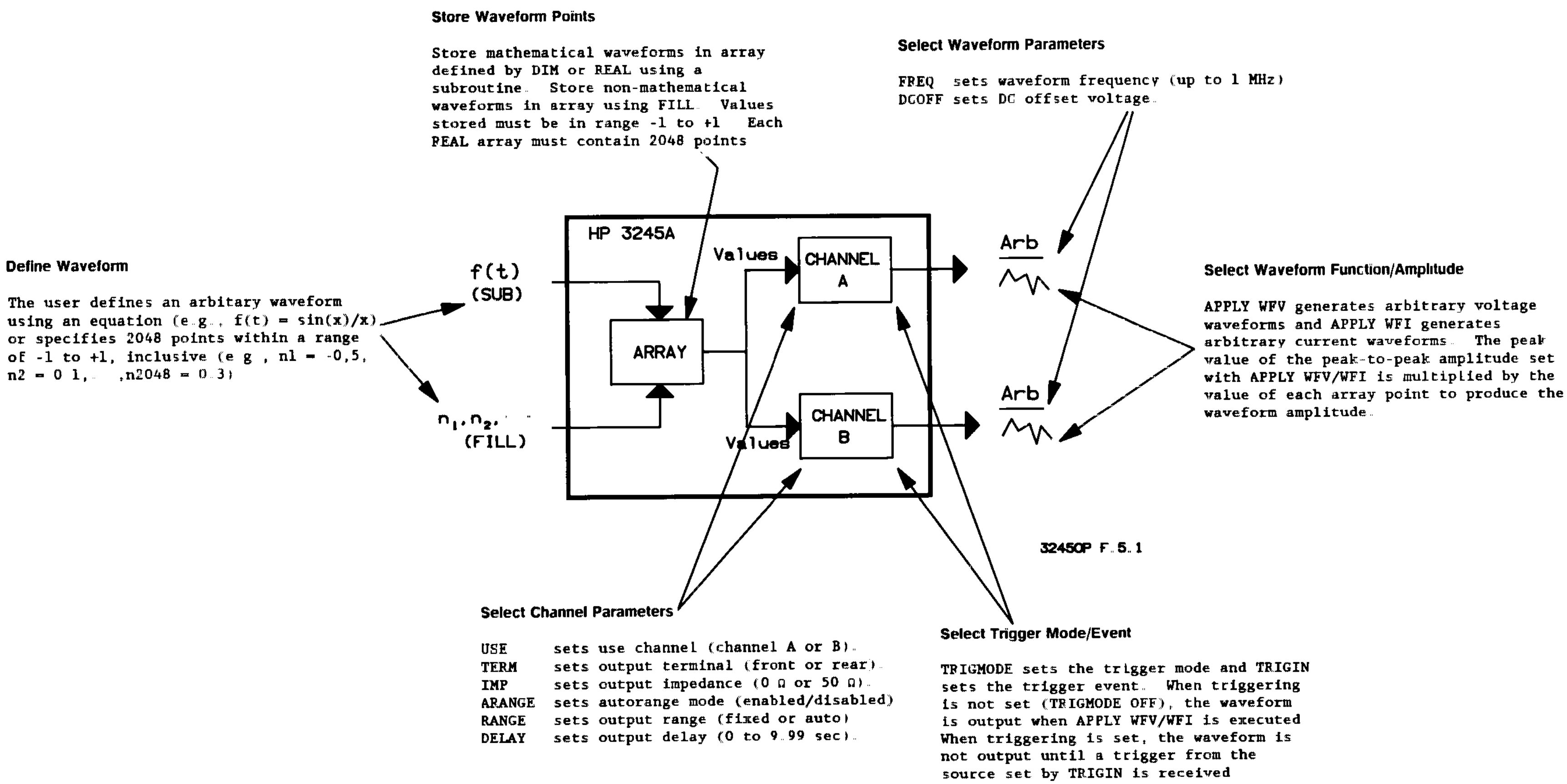


Figure 5-1. Arbitrary Waveforms - Channel Operation

Example 5-1: Sin(x)/x (SINXX5)

This program generates a 1 kHz sin(x)/x waveform with 2048 points The waveform points are stored in REAL array WV_AMP and are then output with **APPLY WFV**.

```
10   !file SINXX5
20   !
30   CLEAR 709                          !Clear HP 3245A
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60   OUTPUT 709;"SUB SINX"              !Define subroutine
70   OUTPUT 709;"  REAL WV_AMP(2047)"   !Declare REAL array
80   OUTPUT 709;"  PI=3.1415"           !Assign/initialize variables
90   OUTPUT 709;"  X=(-5*PI+10*PI/2048)"
100  OUTPUT 709;"  FOR I = 0 TO 2047"   !Define points (2048)
110  OUTPUT 709;"    IF X = 0.0 THEN"
120  OUTPUT 709;"      WV_AMP(I) = 1.0"
130  OUTPUT 709;"    ELSE"
140  OUTPUT 709;"      WV_AMP(I) = SIN(X)/X"
150  OUTPUT 709;"    END IF"
160  OUTPUT 709;"    X=X+(10*PI/2048)"
170  OUTPUT 709;"  NEXT I"
180  OUTPUT 709;"SUBEND"
190  OUTPUT 709;"CALL SINX"             !Execute subroutine
200  OUTPUT 709;"USE 0"                 !Use channel A
210  OUTPUT 709;"  APPLY WFV 2,WV_AMP"  !Generate sin(x)/x waveform
220  END
```

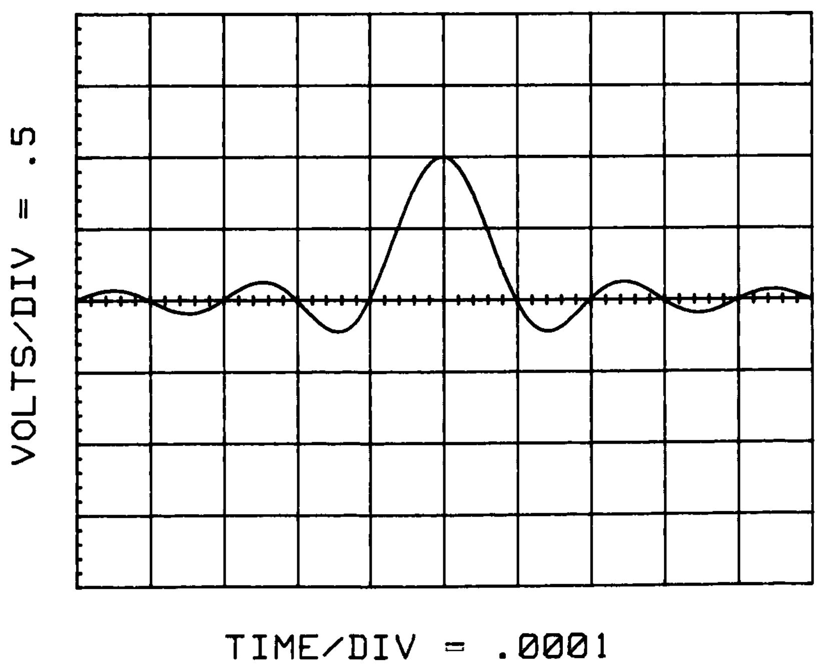


Figure 5-2. Example: Sin(x)/x

Example 5-2: Damped Sine Wave (SINED5)

This program generates a damped sine wave of 2048 points defined by the following equation. The waveform points are stored in REAL array WV_AMP and are then output with **APPLY WFV**.

$$f(t) = \epsilon^{-at} \mathrm{SIN}(\omega t+\Theta)$$

```
10    !file SINED5
20    !
30    CLEAR 709                              !Clear HP 3245A
40    OUTPUT 709;"RST"                       !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
60    OUTPUT 709;"SUB LOAD_ARY"              !Define subroutine
70    OUTPUT 709;"  REAL WV_AMP(2047)"       !Define REAL array
80    OUTPUT 709;"  e=2.71828"               !Initialize variables
90    OUTPUT 709;"  a=4/2048"
100   OUTPUT 709;"  PI=3.1415"
110   OUTPUT 709;"  w=(2*PI)/50"
120   OUTPUT 709;"  FOR T = 0 TO 2047"       !Define 2048 points
130   OUTPUT 709;"    WV_AMP(T) = e^(-1*a*T)*SIN(w*T)"
140   OUTPUT 709;"  NEXT T"
150   OUTPUT 709;"SUBEND"
160   OUTPUT 709;"CALL LOAD_ARY"             !Execute subroutine
170   OUTPUT 709;"USE 0"                     !Use channel A
180   OUTPUT 709;"  APPLY WFV 1, WV_AMP"     !Apply damped sine wave
190   END
```

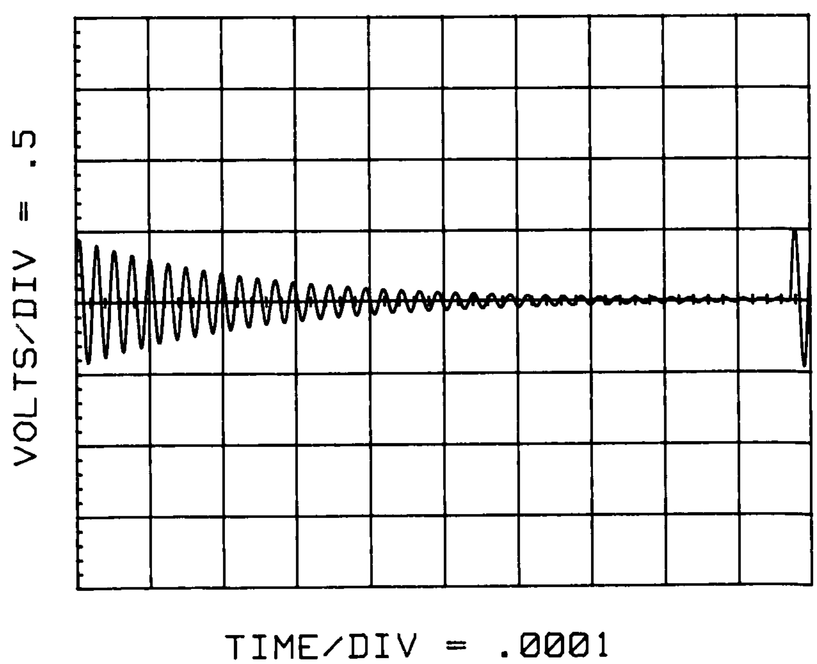


Figure 5-3. Example: Damped Sine Wave

Example 5-3: Exponential Sine Wave (SINEX5)

This program defines a sine wave of 2048 points whose amplitude increases exponentially The point values are stored in REAL array WV_AMP and are then output by **APPLY WFV.**

```
10    !file SINEX5
20    !
30    CLEAR 709                             !Clear HP 3245A
40    OUTPUT 709;"RST"                      !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"                  !Clear HP 3245A memory
60    OUTPUT 709;"SUB LOAD_ARY"             !Define subroutine
70    OUTPUT 709;"  REAL WV_AMP(2047)"      !Define array
80    OUTPUT 709;"  e=2.71828"              !Initialize variables
90    OUTPUT 709;"  a=4/2048"
100   OUTPUT 709;"  PI=3.1415"
110   OUTPUT 709;"  w=(2*PI)/50"
120   OUTPUT 709;"  FOR T = 2047 TO 0 STEP -1"  !Define points (2048)
130   OUTPUT 709;"    WV_AMP(2047-T) = e^(-1*a*T)*SIN(w*T)"
140   OUTPUT 709;"  NEXT T"
150   OUTPUT 709;"SUBEND"
160   OUTPUT 709;"CALL LOAD_ARY"            !Execute subroutine
170   OUTPUT 709;"USE 0"                    !Use channel A
180   OUTPUT 709;"  APPLY WFV 1, WV_AMP"    !Apply sine waveform
190   END
```

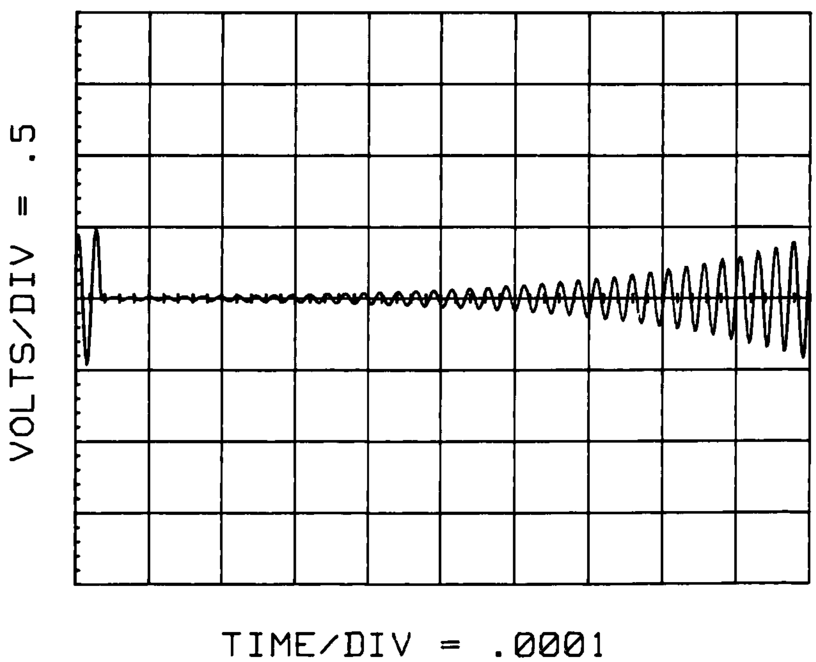


Figure 5-4. Example: Exponentially Increasing Sine Wave

Example 5-4: Linear Step (LSTEP5)

This program defines an arbitrary waveform of 2048 points whose amplitude steps from 0V to 1.0V in 0.2V increments Groups of 341 points (last group 343) are assigned values from 0.0 to 1.0. Point values are stored in REAL array STEP_ARRAY and then output by **APPLY WFV**.

```
10    !file LSTEP5
20    !
30    CLEAR 709                                  !Clear HP 3245A
40    OUTPUT 709;"RST"                           !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"                       !Clear HP 3245A memory
60    OUTPUT 709;"SUB STEP_WV"                   !Define subroutine
70    OUTPUT 709;"  REAL STEP_ARRAY(2047)"       !Define array
80    OUTPUT 709;"  FOR I = 0 TO 340"            !Points 0-340 = 0.0
90    OUTPUT 709;"    STEP_ARRAY(I) = 0.0"
100   OUTPUT 709;"  NEXT I"
110   OUTPUT 709;"  FOR I = 341 to 681"          !Points 341-681 = 0.2
120   OUTPUT 709;"    STEP_ARRAY(I) = 0.2"
130   OUTPUT 709;"  NEXT I"
140   OUTPUT 709;"  FOR I = 682 TO 1022"         !Points 682-1022 = 0.4
150   OUTPUT 709;"    STEP_ARRAY(I) = 0.4"
160   OUTPUT 709;"  NEXT I"
170   OUTPUT 709;"  FOR I = 1023 TO 1363"        !Points 1023-1363 = 0.6
180   OUTPUT 709;"    STEP_ARRAY(I) = 0.6"
190   OUTPUT 709;"  NEXT I"
200   OUTPUT 709;"  FOR I = 1364 TO 1704"        !Points 1364-1704 = 0.8
210   OUTPUT 709;"    STEP_ARRAY(I) = 0.8"
220   OUTPUT 709;"  NEXT I"
230   OUTPUT 709;"  FOR I = 1705 TO 2047"        !Points 1705-2047 = 1.0
240   OUTPUT 709;"    STEP_ARRAY(I) = 1.0"
250   OUTPUT 709;"  NEXT I"
260   OUTPUT 709;"SUBEND"
270   OUTPUT 709;"CALL STEP_WV"                  !Execute subroutine
280   OUTPUT 709;"USE 0"                         !Use channel A
290   OUTPUT 709;"  APPLY WFV 2, STEP_ARRAY"     !Apply waveform
300   END
```

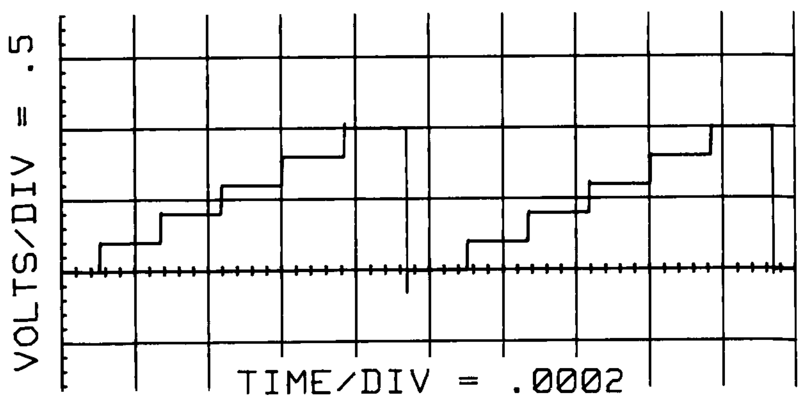


Figure 5-5. Example: Linear Steps

Example 5-5: Exponential Charge/Discharge (CHDSG5)

This program generates an exponential waveform of 2048 points. Each point value is stored in REAL array CURVE and is then output by **APPLY WFV**.

```
10    !file CHDSG5
20    !
30    CLEAR 709                          !Clear HP 3245A
40    OUTPUT 709;"RST"                   !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60    OUTPUT 709;"SUB LOAD_ARY"          !Define subroutine
70    OUTPUT 709;"  REAL CURVE(2047)"    !Define array
80    OUTPUT 709;"  e=2.71828"           !Initialize variables
90    OUTPUT 709;"  RC=200"
100   OUTPUT 709;"  FOR T = 0 TO 2047"   !Define points (2048)
110   OUTPUT 709;"  IF T >= 0 AND T < 1024 THEN"
120   OUTPUT 709;"    CURVE(T) = .1*(1-e^(-1*T/RC))"
130   OUTPUT 709;"  END IF"
140   OUTPUT 709;"  IF T >= 1024 THEN"
150   OUTPUT 709;"    CURVE(T) = .1*(1-e^(-1023/RC))-.1*(1-e^(-1*(T-1023)
/RC))"
160   OUTPUT 709;"  END IF"
170   OUTPUT 709;"  NEXT T"
180   OUTPUT 709;"SUBEND"
190   OUTPUT 709;"CALL LOAD_ARY"         !Execute subroutine
200   OUTPUT 709;"USE 0"                 !Use channel A
210   OUTPUT 709;"APPLY WFV 10,CURVE"    !Apply waveform
220   END
```

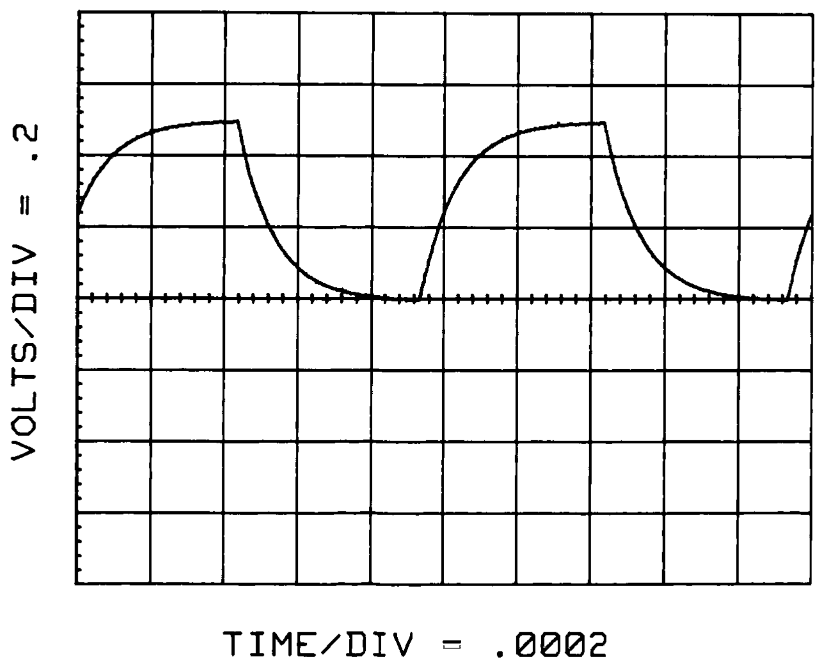


Figure 5-6. Example: Exponential Charge/Discharge

Example 5-6: 1/2 Wave Rectified Sine Wave (SINER5)

This program generates a 1/2 wave rectified sine wave of 2048 points. Each point value is stored in REAL array REC_SINE and is then output by **APPLY WFV**. The waveform frequency is set to 60 Hz.

```
10    !file SINER5
20    !
30    CLEAR 709                                 !Clear HP 3245A
40    OUTPUT 709;"RST"                          !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"                      !Clear HP 3245A memory
60    OUTPUT 709;"SUB RECTIFY"                  !Define subroutine
70    OUTPUT 709;"  REAL REC_SINE (2047)"       !Define array
80    OUTPUT 709;"  PI = 3.1415"                !Initialize variable
90    OUTPUT 709;"  FOR I = 0 TO 2047"          !Define  2048 points
100   OUTPUT 709;"    REC_SINE(I) = SIN(2*PI*(I/2047))"
110   OUTPUT 709;"  NEXT I"
120   OUTPUT 709;"  FOR I = 1023 TO 2047"       !Set points 1023-2047 to 0
130   OUTPUT 709;"    REC_SINE(I) = 0"
140   OUTPUT 709;"  NEXT I"
150   OUTPUT 709;"SUBEND"
160   OUTPUT 709;"CALL RECTIFY"                 !Execute subroutine
170   OUTPUT 709;"USE 0"                        !Use channe A
180   OUTPUT 709;"FREQ 60"                      !60 Hz waveform frequency
190   OUTPUT 709;"APPLY WFV 10, REC_SINE"       !Apply waveform
200   END
```

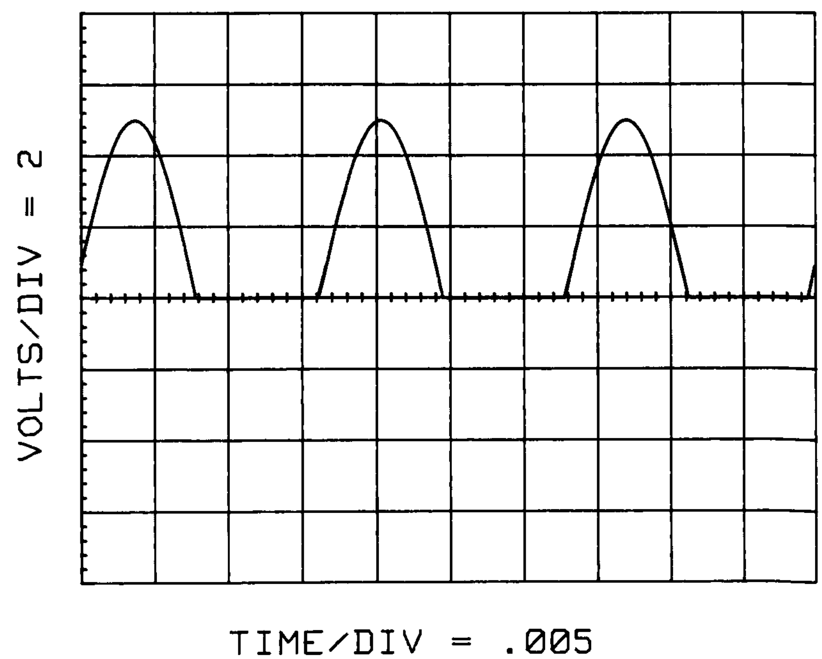


Figure 5-7. Example: 1/2 Wave Rectified Sine Wave

Example 5-7: Sine Wave With Voltage Spikes (SPIKE5)

This program generates a 5 kHz sine waveform of 2048 points which contains voltage spikes on the positive peaks. The program first defines the sine waveform using all 2048 points and then redefines the appropriate points to add the voltage spikes. (It takes about 15 seconds for the HP 3245A to generate this waveform.)

```
10   !file SPIKE5
20   !
30   CLEAR 709                              !Clear HP 3245A
40   OUTPUT 709;"RST"                       !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
60   OUTPUT 709;"REAL PI"                   !Define REAL variable
70   OUTPUT 709;"INTEGER I,J,WIDTH"         !Define INTEGER vars
80   OUTPUT 709;"DIM CARRAY(2047)"          !Define REAL array
90   OUTPUT 709;"PI = 3.1415925"            !Assign value to PI
100  OUTPUT 709;"SUB DEF_ARRAY"             !Define subroutine
110  OUTPUT 709;"  FOR I = 0 TO 2047"       !Define sine waveform
120  OUTPUT 709;"    CARRAY(I) = 0.1*SIN(2*PI*I/2048)"
130  OUTPUT 709;"  NEXT I"
140  OUTPUT 709;"  WIDTH = 50"              !Set spike width
150  OUTPUT 709;"  FOR J = 1 TO WIDTH"      !Define spike shape
160  OUTPUT 709;"    I = 512-WIDTH+J"
170  OUTPUT 709;"    CARRAY(I) = CARRAY(I) + 0.9*J/WIDTH"
180  OUTPUT 709;"  NEXT J"
190  OUTPUT 709;"  FOR J = 1 TO (WIDTH-1)"
200  OUTPUT 709;"    I = 512+WIDTH-J"
210  OUTPUT 709;"    CARRAY(I) = CARRAY(I) + 0.9*J/WIDTH"
220  OUTPUT 709;"  NEXT J"
230  OUTPUT 709;"SUBEND"
240  OUTPUT 709;"CALL DEF_ARRAY"            !Execute subroutine
250  OUTPUT 709;"FREQ 5000"                 !5 kHz frequency
260  OUTPUT 709;"USE 0"                     !Use channel A
270  OUTPUT 709;"APPLY WFV 1.2,CARRAY"      !Apply waveform
280  END
```

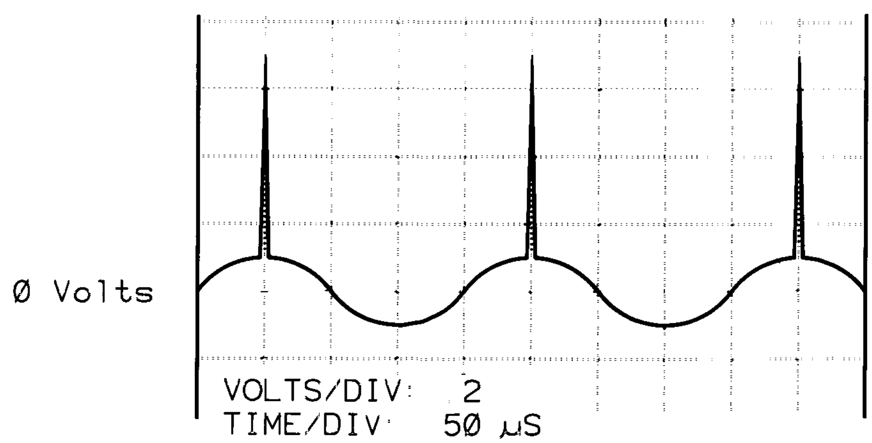


Figure 5-8. Example: Sine Waveform With Voltage Spikes

Example 5-8: Generating Pseudo-Random Noise (NOISE5)

Pseudo-random noise can be generated as an arbitrary waveform or can be added to a previously defined arbitrary waveform by using the HP 9000 Series 200/300 controller **RND** command. This program generates a Gaussian (normal distribution) noise pattern. (It takes about 30 seconds for the controller to generate and download this waveform.)

```
10   !file NOISE5
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"DIM N(2047)"       !Define REAL array
70   FOR I = 0 TO 2047              !Generate pseudo-random pts
80     OUTPUT 709;"N(";I;") = "; 0.4*(RND+RND+RND+RND+RND)-1
90   NEXT I
100  OUTPUT 709;"FREQ 5"            !5 kHz frequency
110  OUTPUT 709;"USE 0"             !Use channel A
120  OUTPUT 709;"APPLY WFV 2.5,N"   !Generate Gaussian waveform
130  END
```

Precomputed Waveform Programs

When a waveform is output with an **APPLY** command, the HP 3245A converts the waveform data to the format required by the instrument digital-to-analog converter (DAC) as the command executes. By precomputing the waveform, data is converted to the DAC format before the **APPLY** command is executed.

As a result, the time required to output the waveform (i.e., execution time of the **APPLY** command) is reduced. HP 3245A waveforms which can be precomputed are sine waves, ramp waves, and arbitrary waveforms. Several example programs for precomputed waveforms follow.

Example 5-9: Precomputed Sine Waveform (SINPC5)

This program precomputes and then outputs a 10 kHz sine waveform with an amplitude of 5 Vac PP from channel A. Since scaling is used, the waveform is output about 50 msec after **APPLY WFV** is executed. Without scaling (precomputing), the waveform would be output about 75 msec after **APPLY WFV** was executed.

```
10   !file SINPC5
20   !
30   CLEAR 709                              !Clear HP 3245A
40   OUTPUT 709;"RST"                       !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER SCALE(2047)"       !Dim INTEGER array
70   OUTPUT 709;"FILLAC SCALE"              !Fill INTEGER array
80   OUTPUT 709;"FREQ 10E+3"                !10 kHz frequency
90   OUTPUT 709;"APPLY WFV 5,SCALE"         !Generate sine waveform
100  END
```

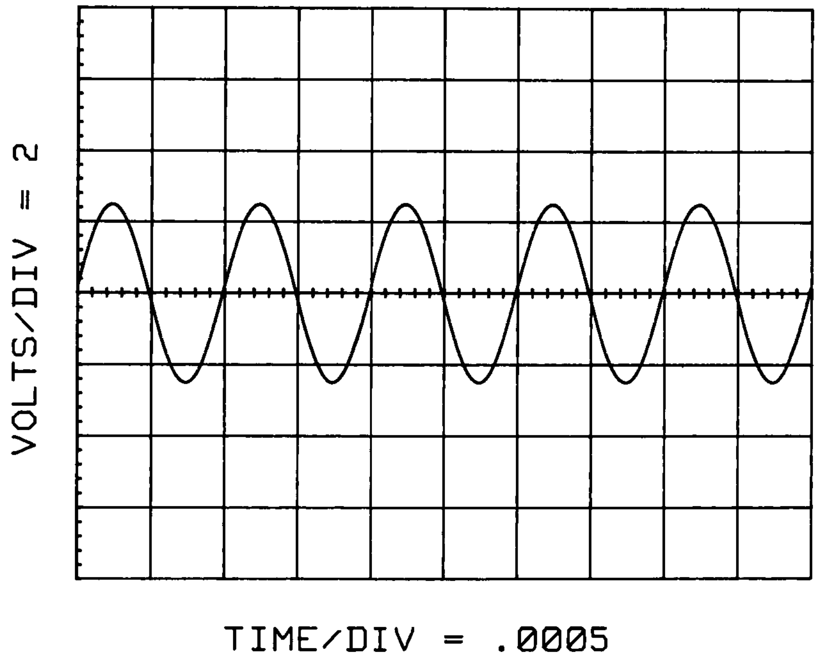


Figure 5-9. Example: Precomputed Sine Waveform

Example 5-10: Precomputed Ramp Waveform (RMPPC5)

This program precomputes and then outputs a 3 kHz ramp waveform with an amplitude of 3.25 Vac PP from channel A. The duty cycle is set to 33%. Since scaling is used, the waveform is output about 50 msec after **APPLY WFV** is executed. Without precomputing, the waveform would be output about 145 msec after **APPLY WFV** was executed.

```
10   !file RMPPC5
20   !
30   CLEAR 709                          !Clear HP 3245A
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER J(2047)"       !Dim INTEGER array
70   OUTPUT 709;"FILLRP J,33"           !Fill array, set 33% duty
80   OUTPUT 709;"FREQ 3000"             !3 kHz frequency
90   OUTPUT 709;"APPLY WFV 3.25,J"      !Generate ramp waveform
100  END
```

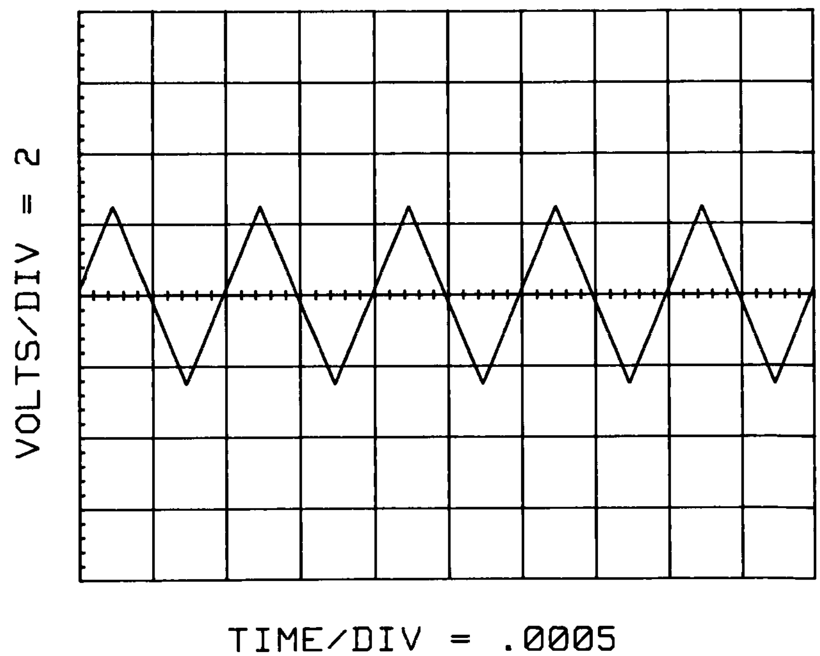


Figure 5-10. Example: Precomputed Ramp Waveform

Example 5-11: Precomputed Arbitrary Waveform (ARBPC5)

This program defines, precomputes, and then outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A. Since precomputing is used, the waveform is output about 50 msec after **APPLY WFV** is executed. Without precomputing, the waveform would be output about 675 msec after **APPLY WFV** was executed.

```
10   !file ARBPC5
20   !
30   CLEAR 709                               !Clear HP 3245A
40   OUTPUT 709;"RST"                        !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                    !Clear HP 3245A memory
60   OUTPUT 709;"DIM CARRAY(2047)"           !Dim REAL array
70   OUTPUT 709;"REAL PI"                    !Define REAL variable
80   OUTPUT 709;"PI = 3.1415925"             !Assign value to PI
90   OUTPUT 709;"SUB DEF_ARRAY"              !Begin subroutine
100  OUTPUT 709;"  FOR I = 0 TO 2047"        !Compute array values
110  OUTPUT 709;"    CARRAY(I) = COS(2*PI*I/2048)"
120  OUTPUT 709;"  NEXT I"
130  OUTPUT 709;"SUBEND"                     !End subroutine
140  OUTPUT 709;"CALL DEF_ARRAY"             !Call subroutine
150  OUTPUT 709;"INTEGER J(2047)"            !Define INTEGER array
160  OUTPUT 709;"FILLWF CARRAY,J"            !Scale, store result in J
170  OUTPUT 709;"FREQ 2000"                  !2 kHz waveform frequency
180  OUTPUT 709;"APPLY WFV 2.3,J"            !Output precomputed wave
190  END
```

Triggered Waveform Programs

The HP 3245A has four triggering modes: off, armed (synchronized), gated, and dual-frequency. When the trigger mode is off, no triggering is required, since the waveform is immediately generated from the use channel when the appropriate **APPLY** command is executed. In armed mode, the waveform is not output until the use channel is triggered from a source specified by the user.

In gated mode, the waveform is generated only when the specifed trigger signal level is low (0 V). When the trigger signal goes high (+5 V), the waveform stops upon completion of the present cycle. In dual-frequency mode, the waveform is generated at two different frequencies, based on the level of the specified trigger signal.

Example 5-12: Triggered Cosine Waveform (TRIGC5)

This program defines and outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A. The output is generated when a (user-supplied) external trigger is input to the channel A Trigger (I/O) port.

```
10   !file TRIGC5
20   !
30   CLEAR 709                               !Clear HP 3245A
40   OUTPUT 709;"RST"                        !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                    !Clear HP 3245A memory
60   OUTPUT 709;"DIM CARRAY(2047)"           !Dim REAL array
70   OUTPUT 709;"REAL PI"                    !Define REAL variable
80   OUTPUT 709;"PI = 3.1415925"             !Assign value to PI
90   OUTPUT 709;"TRIGMODE ARMWF"             !Set ARMED mode
100  OUTPUT 709;"SUB DEF_ARRAY"              !Begin subroutine
110  OUTPUT 709;"  FOR I = 0 TO 2047"        !Compute array values
120  OUTPUT 709;"    CARRAY(I) = COS(2*PI*I/2048)"
130  OUTPUT 709;"  NEXT I"
140  OUTPUT 709;"SUBEND"                     !End subroutine
150  OUTPUT 709;"CALL DEF_ARRAY"             !Call subroutine
160  OUTPUT 709;"FREQ 2000"                  !2 kHz frequency
170  OUTPUT 709;"APPLY WFV 2.3,CARRAY"       !Apply cosine waveform
180  OUTPUT 709;"TRIGIN EXT"                 !External trigger
190  END
```

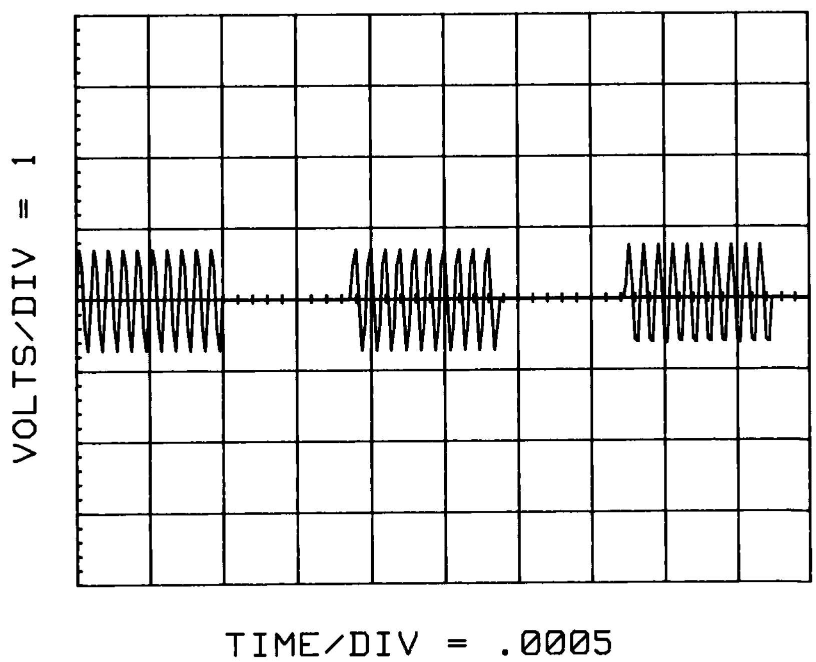


Figure 5-11. Example: Triggered Cosine Waveform

Command Summary

This section summarizes commands used to generate arbitrary waveforms and shows a suggested programming sequence to generate arbitrary waveforms. Figure 5-12 shows a typical sequence to program the use channel for user-defined, precomputed, and triggered arbitrary waveforms and lists related commands. Table 5-1 summarizes commands used to generate arbitrary waveforms. Refer to the HP 3245A Command Reference Manual for command details.

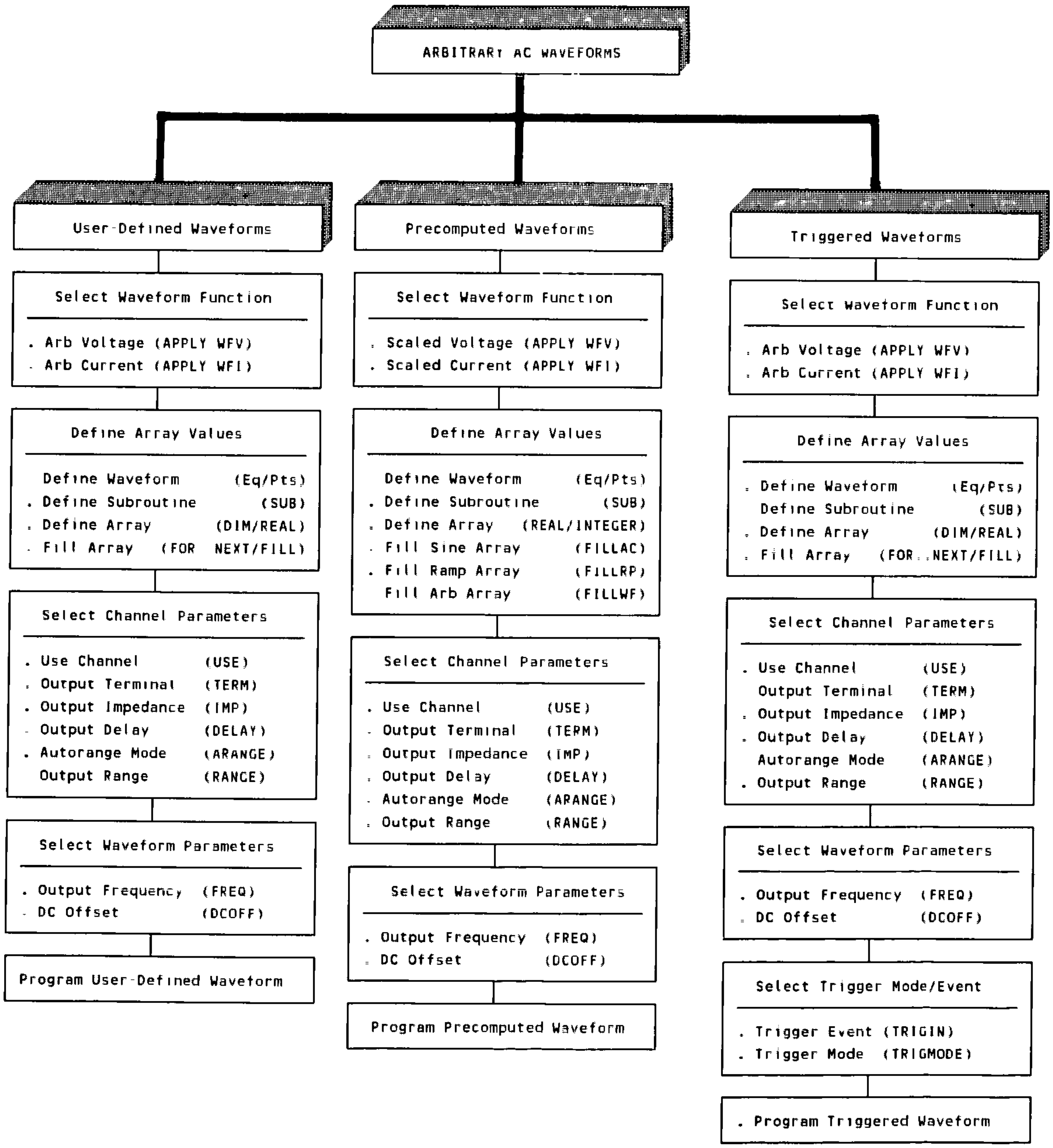


Figure 5-12. Arbitrary Waveforms - Programming Steps

Table 5-1. Arbitrary Waveforms - Command Summary

Command	Description	Power-On
Waveform Function		
APPLY WFV	Arb wave (voltage)	0 Volts
APPLY WFI	Arb wave (current)	0 Amps
Subroutine Operations		
SUB sub_name	Define subroutine	---
FOR...NEXT	FOR...NEXT loop	---
SUBEND	End subroutine	---
CALL sub_name	Call subroutine	---
RUN sub_name	Run subroutine	---
Array Operations		
DIM array_name	Define REAL array	---
REAL name	Define REAL array	---
INTEGER name	Define INTEGER array	---
FILL name,list	Fill array	---
FILLAC	Fill array/sine wave	---
FILLRP	Fill array/ramp wave	---
FILLWF	Fill array/arb wave	---
Channel Parameters		
USE ch	Use channel	Chan A
TERM [mode]	Output terminal	Front
IMP mode	Output impedance	0 Ohms
ARANGE [control]	Autorange mode	ON
RANGE [max_output]	Output range	Auto (1V)
DELAY [time]	Output delay	0.04 sec
Waveform Parameters		
FREQ freq1	Waveform frequency	1000 Hz
DCOFF volts	DC offset	0 Volts
Trigger Event/Mode		
TRIGIN event	Trigger event	High
TRIGMODE mode	Trigger mode	Off
REFOUT dest	Ref freq out dest	Ext
REFIN source	Ref freq in source	Int
PANG degrees	Phase angle	0 degrees
PHSYNC	Synchronized mode	---

PART III
INSTRUMENT OPERATIONS

Contents

Chapter 6
Front Panel Operation

Chapter 6
Front Panel Operation

Chapter Contents

This chapter shows how to enter commands and generate outputs using the front panel keyboard. Commands can be entered using the front panel keys to enter the equivalent command or using menu operation to directly enter the command This chapter describes both methods and illustrates some example keystroke sequences to enter and edit commands.

NOTE

Figure 6-1 at the end of the chapter is a diagram of the HP 3245A front panel which shows the functions of all front panel keys, the front panel display, and the input/output connectors.

The chapter contents are:

- **Front Panel Overview** summarizes the functions of the display, the keyboard groups, and the input/output connectors.

- **Front Panel Operation** shows how to get started using the front panel, how to use the front panel keys, and how to use menu operation.

- **Front Panel Functions** describes basic front panel functions, summarizes DC outputs and AC waveforms, and lists several advanced operations which can be done from the front panel.

Front Panel Overview

This section provides an overview of HP 3245A front panel features, including:

- Display functions/annunciators.
- Keyboard group functions.
- Input/output connector functions.

Display Functions

The front panel display has two major functional areas: display and HP-IB/status annunciators.

Display

The display displays the value of the output voltage/current or shows the command(s) to be entered. In addition, when monitor mode is set, the display shows the current state of the specified channel. The display includes a 15-character display window plus a Channel A/Channel B indicator arrow. The indicator arrow shows which channel's output is being displayed and is receiving front panel commands.

Display When Commands Are Entered

When a command is entered into the HP 3245A, the command is displayed on the display. At power-on or following a reset, the display shows 0.000000E+0DCV and the display arrow points to Channel A showing that channel A is the use channel (the channel to receive output commands). When a command is entered, the command and associated parameter(s) are displayed. For example, when the **DCV** key is pressed, APPLY DCV ■ is displayed, where ■ represents a blinking cursor.

Display When Outputs Are Generated

When a voltage or current is output from the use channel, the display shows the value of the output and shows the type of output. The form is -1.123456E+0type. For example, an output of -5.125 VDC is displayed as -5.125000E+0DCV on the display.

Display When Monitor HP-IB Mode is Set

When the monitor HP-IB mode is set (with the **MON HP-IB** command), the display shows the latest command the be sent over HP-IB (**MON HP-IB**). Refer to the **MON** command for details.

HP-IB/Status Annunciators

There are four HP-IB annunciators on the display: REM, SRQ, TALK, and LISTEN. When the HP 3245A is in remote state, the REM annunciator is ON. There are four status annunciators on the display: MRNG, ERR, SHIFT, and MORE INFO. Figure 6-1 summarizes HP-IB/status annunciator functions.

Keyboard Groups

The HP 3245A front panel has three groups of keyboards: FUNCTION/RANGE, NUMERIC/USER, and MENU. See Figure 6-1 at the end of the chapter for the function of each key.

FUNCTION/RANGE Keys

The FUNCTION/RANGE keys are primarily used to output DC and AC voltages/currents. However, this key group can be used to set shifted mode, set local (front panel) operation, check the HP-IB address, and test or reset the instrument. In addition, the "arrow" keys can be used along with some MENU keys to enter menu commands.

NUMERIC/USER Keys

In unshifted mode, the NUMERIC/USER keys can be used to enter numbers and/or punctuation for command parameters or can be used to enter commands or backspace the cursor. In shifted mode, this key group can be used to enter output parameters and select triggering modes. In addition, the keys can be used to clear the HP 3245A, recall commands, redefine numeric keys, read error messages, and use the Help function.

MENU Keys

In unshifted mode, the MENU key group can be used to enter output parameters or to store and recall channel states. In shifted mode, the keys provide an entry point into the alphabetical list of menu commands or can be used to create a query command (with the **?** key). For example, when the **Shift** key and then the **C** menu keys are pressed, the display shows **CAL** since **CAL** is the C entry point for the menu command list.

Input/ Output Connectors

Each channel has three BNC connectors on the front panel: Output, Sync Out, and Trigger (I/O). Use the Output connectors to output a signal from the channel. When outputting AC waveforms, you can use the Sync Out port to generate a sync signal with logic level "1" when the output level is positive with respect to its DC offset and logic level "0" when the output level is negative with respect to its DC offset.

The Trigger (I/O) ports act as a trigger entry point and as an output trigger source. When trigger output mode is enabled (with **TRIGOUT EXT**), a TTL-compatible trigger can be output from the Trigger (I/O) port. When the Trigger (I/O) port is disabled (with **TRIGMODE OFF**), the port will accept a TTL-compatible trigger from an external source.

Front Panel Operation

This section shows how to use the front panel keys and menu operation to generate outputs or perform several instrument functions. Refer to the next section "Front Panel Functions" for a description of the functions you can do from the front panel. This section includes:

- Initial operation of the HP 3245A.
- Using the front panel keyboard.
- Using menu operation

Getting Started

The following paragraphs show how to set the HP 3245A for initial operation using front panel keys, including applying power to the HP 3245A and resetting, testing, and clearing the instrument.

Applying Power

To begin front panel operation, connect a BNC cable from the channel A Output terminal to an oscilloscope or to a DMM as shown in Chapter 2 - Initial Operation (see Figure 2-5). Then, press the HP 3245A ON/OFF switch ON and watch the display as the HP 3245A cycles through its power-on sequence. When the sequence successfully completes, the display shows 0.000000E+0DCV and the MORE INFO annunciator is ON.

Also, the display arrow points to Channel A which means that channel A is the **USE** channel (the channel to receive output commands). For a two-channel instrument, you can change the **USE** channel to channel B by pressing the **Chan A/Chan B** key. Press this key again to return to channel A as the **USE** channel When the power-on sequence completes, the HP 3245A is set to the power-on state shown in Table 6-1 and is ready for operation

Table 6-1. HP 3245A Power-On State

Item	Command	Power-On/Reset State
Output Function	APPLY	DC Volts (0 V)
Auto Range	ARANGE	On
DC Offset	DCOFF	0 Volts
DC Resolution	DCRES	High-Res Mode
Delay Time	DELAY	0.04 Seconds
Drive TB0	DRIVETB0	Off
Drive TB1	DRIVETB1	Off
Duty Cycle	DUTY	50%
Frequency	FREQ	1000 Hz
Output Impedance	IMP	0 Ohms
Phase Angle	PANG	0 Degrees
Range	RANGE	1V (Autogange)
Ref Freq Input	REFIN	External (ch B)
Ref Freq Output	REFOUT	External (ch A)
Sync Destination	SYNCOUT	Off
Output Terminal	TERM	Front
Input Trigger Source	TRIGIN	High
Trigger Mode	TRIGMODE	Off
Trigger Output Mode	TRIGOUT	Off

Setting Shifted Mode

The blue **Shift** key at the lower right-hand corner of the FUNCTION/RANGE key group controls which key function is to be entered when the key is pressed. At power-on, the SHIFT annunciator is OFF and the function printed on the key is entered when the key is pressed. This is the unshifted mode.

When the **Shift** key is pressed, the SHIFT annunciator is turned ON and the function printed above the key is entered when that key is pressed. This is the shifted mode. If you press the **Shift** key again, the HP 3245A reverts back to the unshifted mode.

For example, at power-on, if the **Local/Address** key (next to the **Shift** key) is pressed, the display does not change. However, if the **Shift** key is pressed first and then the **Local/Address** key is pressed, the HP-IB address of the HP 3245A (9 with factory setting) is displayed.

NOTE

*For the NUMERIC (***0** *through* **9***) keys only, pressing the NUMERIC key before any other entry will display the shifted function, even if the instrument is in unshifted mode. For example, pressing the* **3** *key after power-on displays the shifted function (***DRIVETB1***).*

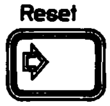
Resetting the HP 3245A

If the HP 3245A is set to an undesired state or if you want to "start over" programming the instrument, you can cycle power with the ON/OFF switch or you can reset the instrument with the **Shift** key followed by the **Reset** key. Resetting the HP 3245A places it in its power-on state, except that reset does not perform a self-test; delete variables, arrays, and subroutines; or change the present use channel.

NOTE

If you make a mistake in entering keystroke sequences, or you want to return the instrument to a known state, simply reset the HP 3245A by pressing the **Shift** *key followed by the* **Reset** *key as shown in the following keystroke sequence. All example keystroke sequences in this chapter assume a* **Reset** *starting point.*

Example 6-1: Resetting the HP 3245A

Key	Action	Display
	Shifts keyboard to the shifted mode.	0.000000E+0DCV*
Reset 	Resets HP 3245A.	0.000000E+0DCV**

* = SHIFT annunciator ON.
** = All annunciators except MORE INFO are OFF.

Testing the HP 3245A

Although the HP 3245A performs a self-test at power-on, you can also do a self-test from the front panel by using the **Shift** and **Test** keys, as shown in the following keystroke sequence. (When the keystroke sequence completes, reset the instrument with the **Shift** and **Reset** keys.)

NOTE

You can also perform more extensive tests of the HP 3245A using the **FTEST** *or* **DTEST** *command entered via menu operation. Refer to Chapter 9 - Other Instrument Functions for details.*

Example 6-2: Testing the HP 3245A

Key	Action	Display
	Shifts keyboard to the shifted mode.	0.000000E+0DCV*
Test 	Self-tests HP 3245A.	PASS**

* = SHIFT annunciator is ON.
** = If the test fails, FAIL is displayed and qualified service-trained personnel should verify proper operation before the instrument is used.

Clearing the HP 3245A

When the **Clear** key is pressed, the HP 3245A is cleared. Clear differs from reset in that with clear any outputs from the channel(s) are not changed (i.e., it does not reset the hardware, but DOES abort any subroutines).

If you want to change a keystroke entry before the entry is executed (with the **Enter** key), you can clear the entry with the **Shift** key followed by the **Clear** key. However, if the entry has already been executed, use **Shift** followed by **Reset** to reset the HP 3245A.

Example 6-3: Clearing the HP 3245A

Key	Action	Display
	Shifts keyboard to shifted mode.	0.000000E+0DCV*
Clear 	Clears HP 3245A.	0.000000E+0DCV**

* = SHIFT annunciator is ON.
** = All annunciators except MORE INFO are OFF unless ERR annunciator was ON when CLEAR was executed.

Using the Keyboard

The following paragraphs show how to use front panel keys to control the display and to enter or recall commands.

Controlling the Display

The front panel display is controlled using the four "arrow" keys (**Up Arrow**, **Down Arrow**, **Left Arrow**, and **Right Arrow**) and the **Back Space** key.

Using the Left Arrow/Right Arrow Keys

The **Left Arrow** key moves the display one character to the right, while the **Right Arrow** key moves the display one character to the left. The **Back Space** key moves the display to its "home" position.

NOTE

If the **Left Arrow** *and* **Right Arrow** *keys seem to move the information in the wrong direction, think of the display as a window you can move to the left or right to view the information.*

For example, at power-on the display is 0.000000E+0DCV. If the **Right Arrow** key is pressed once, the display becomes .000000E+0DCV (moves one character to the left). Each time the **Right Arrow** key is pressed, the display moves another character to the left.

Viewing Long Displays

For some commands or messages, the information exceeds the length of the 15-character display window. In this case, the MORE INFO annunciator is ON and the additional information is stored in HP 3245A memory. You can read the additional information using the **Right Arrow** key. For example, holding the **Right Arrow** key down until the display movement stops allows the entire power-on display message to be scrolled, ending with EXT, DELAY 0.04.

Using the Back Space Key

The **Back Space** key returns the display to its "home" position. For example, in the power-on display which was just cycled to its end position (EXT, DELAY 0.04), you can return the display to its start (0.000000E+0DCV) position by repeatedly using the **Left Arrow** key. However, it is much easier to return to this position by pressing the **Back Space** key once.

Using the Up/Down Arrow Keys

The **Up Arrow** and **Down Arrow** keys can also be used to scroll the display. However, these keys scroll the information one field at a time, rather than one character at a time. (These keys are also used in menu mode operation - refer to "Using Menu Operation" for details.)

For example, in the power-on display, pressing the **Down Arrow** key advances the display to the next field (FREQ 1000.00). Then, pressing the **Up Arrow** key returns the display back to its original position (0.000000E+0DCV). Thus, the **Up Arrow** and **Down Arrow** offer a faster way to scroll the display than do the **Left Arrow** and **Right Arrow** keys.

Entering Commands

Commands can be entered into the HP 3245A from the front panel using the specified key(s) or using the MENU keys. When a front panel key is pressed, the command keyword or parameter associated with the key is displayed on the display. For example, assuming power-on conditions, when the **DCV** key is pressed, the associated command keyword **APPLY DCV** appears on the display

When menu operation is used, pressing the **Shift** key followed by one of the seven menu entry keys (**C**, **D**, etc) displays the command associated with the menu key For example, assuming power-on operation, pressing the **Shift** key followed by the **C** key displays **CAL** on the display. Refer to "Using Menu Operation" for details on using the menu keys. The following paragraphs show how to enter command keywords and parameters using keyboard operation.

HP 3245A Command Structure

Commands for the HP 3245A have the form **KEYWORD** [*parameter* [,*parameter*]] where the keyword identifies the command function and the parameter(s) identify the function values or settings. A parameter is enclosed in brackets ([]) is optional. A query command has the form **KEYWORD?** Also, multiple commands can be entered by separating the commands with a semicolon (;).

For example, the **ADDRESS** command has a keyword with no parameters, while **ADDRESS?** is a query command. **APPLY DCV** *volts* has a single (required) parameter *volts*, while **DELAY** [*time*] has a single (optional) parameter. **APPLY WFV** *pp_amplitude* [,*array_name*] is an example of a command with both a required and an optional parameter. **APPLY DCV 1.5;OUTPUT?** is an example of two commands in a string since the commands are separated by a semicolon.

NOTE

Not all HP 3245A commands can be entered from the front panel. Refer to Table 6-1 for a list of commands which can be entered using front panel keys and/or menu operation.

Using the Enter Key

As shown previously, the reset function was executed by pressing the **Shift** key and then the **Reset** key. Thus, **RESET** is an example of an immediate execution command since the command was executed without having to press the **Enter** key.

However, many commands are not immediately executable and you must press the **Enter** key (at the lower right-hand corner of the NUMERIC/USER key group) to actually execute the command. For example, pressing the **Shift** key followed by the **Term** key displays **TERM** on the display. However, the command is not executed until the **Enter** key is pressed.

Using the Chan A/Chan B Key

For a two-channel instrument, you can select the **USE** channel (A or B) to receive commands by using the **Chan A/Chan B** key. At power-on/reset, channel A is the **USE** channel and the front panel arrow points to Channel A. By pressing the **Chan A/Chan B** key, the **USE** channel is changed to channel B. Pressing this key again returns channel A as the **USE** channel.

Entering Command Keywords

To enter a command keyword, first determine whether the keyword is represented by the key function in shifted or unshifted mode. Then, press the **Shift** key and the desired key (shifted mode) or press the function key to enter the keyword. When the keyword is entered, it is displayed on the display. An example of entering a command keyword follows which uses the **ADDRESS?** command (a query command) to read the HP 3245A HP-IB address.

Example 6-4: Entering Command Keyword

Key	Action	Display
	Shifts keyboard to shifted mode.	0.000000E+0DCV*
Address 	Displays HP-IB address.	9

* = SHIFT annunciator is ON.

Entering Command Parameters

After the keyword is entered, you may need to enter the command parameter(s). Several commands, such as the **APPLY** commands, require a number and (sometimes) punctuation for the command parameter Use the numeric keys (**0** through **9** and the **E** key) to enter required numbers (0, 1, ,9 and E).

Use the punctuation keys (in the NUMERIC/USER key group) to enter a minus sign (**-** key), a comma (**,** key), a semicolon (**;** key), or a period (**.** key). Example 6-5 uses numeric and punctuation keys to output a DC voltage @ 1.2 VDC. In this example, as in all displays, the symbol ■ denotes a blinking cursor (the point at which the next item to be entered will appear).

Example 6-5: Entering Command Parameters

Key	Action	Display
⎓V	Set HP 3245A to output DCV on channel A.	APPLY DCV ■
1	Enter "1" value.	APPLY DCV 1■
.	Enter . point.	APPLY DCV 1.■
2	Enter "2" value.	APPLY DCV 1.2■
Enter	Output 1.2 VDC on channel A.	1.200000E+0DCV

Recalling Commands

You can recall the last commands executed without repeating the command entry process by using the **Recall** key (in the NUMERIC/USER key group) the required number of times. When you press the **Shift** key followed by the **Recall** key, the display shows the last command executed. (You cannot recall commands which are immediately executed) After recalling the command, you can modify it (as required) and execute the modified command as shown in "Using Menu Operation".

Using Menu Operation

The following paragraphs show how to use menu operation to enter and edit commands. It lists menu commands alphabetically and shows how to use the MENU and arrow keys to enter and edit commands. Table 6-2 is an alphabetical listing of HP 3245A menu commands. If you need more information on a menu command, refer to HP 3245A Command Reference Manual for details. Note that not all HP 3245A commands are included in the menu command set.

Using the MENU Keys

As noted, some commands can be executed using the front panel keys. However, some commands do not have an associated front panel key. For some HP 3245A commands, you can enter the command using the MENU keys. Even if a command has an associated key (such as **APPLY DCV**, for example), you can also use the MENU keys to enter the command

There are seven MENU keys: the **C**, **D**, **H**, **M**, **R**, **S**, and **T** keys in the MENU key group. To use menu operation, press the **Shift** key followed by one of the seven MENU keys. Each MENU key provides an entry point into the alphabetical list of menu commands (refer to Table 6-2 for a list of available commands).

Table 6-2. HP 3245A Menu Commands

A

Command	Description
ABORT	Abort Subroutine
ADDRESS	Set HP IB Address
ADDRESS?	Read HP-IB Address
APPLY?	Query Output Function
APPLY ACI	Apply Current Sine Wave
APPLY ACV	Apply Voltage Sine Wave
APPLY DCI	Apply DC Current
APPLY DCMEMI	Apply Triggered DCI
APPLY DCMEMV	Apply Triggered DCV
APPLY DCV	Apply DC Voltage
APPLY RPI	Apply Current Ramp Wave
APPLY RPV	Apply Voltage Ramp Wave
APPLY SQI	Apply Current Square Wave
APPLY SQV	Apply Voltage Square Wave
APPLY WFI	Apply Current Arbitrary Wave
APPLY WFV	Apply Voltage Arbitrary Wave
ARANGE	Set Autorange
ARANGE?	Read Autorange

B - C

Command	Description
BEEP	Beep
BLOCKOUT	Block Output Mode
CAL	Channel Adjustment
CALEN?	Read CAL Jumper Setting
CALL	Call Subroutine
CALNUM?	Calibration Number Query
CALSTR?	Read Calibration Data
CAT	Catalog
CLR	Clear HP 3245A
CLROUT	Clear Output Buffer
COMPRESS	Compress Subroutine
CONT	Continue
CRESET	Channel Reset
CTYPE?	Channel Number Query

D

Command	Description
DCOFF	Set DC Offset Voltage
DCOFF?	Read DC Offset Voltage
DCRES	Set DC Resolution
DCRES?	Read DC Resolution
DELAY	Set Output Delay
DELAY?	Read Output Display
DELSUB	Delete Subroutine
DEMO	HP 3245A Demonstration
DRIVETBn	Set Trigger Bus n Source
DRIVETBn?	Read Trigger Bus n Source
DTEST	Device Self-Test
DUTY	Set Duty Cycle
DUTY?	Read Duty Cycle

E - I

Command	Description
ERRSTR?	Error String Query
FASTAMP	Fast Amplitude Sweep
FASTFREQ	Fast Frequency Sweep
FREQ	Set Output Frequency
FREQ?	Read Output Frequency
FTEST	Fixtured Self-Test
HELP	HELP Function
ID?	Model Number Query
IDN?	Identity Query
IMP	Set Output Impedance
IMP?	Read Output Impedance
INBUF	Enable Input Buffer

L - P

Command	Description
LIST	List Subroutine
LOCAL	Go to Local
MEMAVAIL?	Memory Available Query
MON	Monitor Conditions
OFORMAT	Output Format
OUTBUF	Output Buffer
OUTPUT?	Output Value Query
PANG	Set Phase Angle
PANG?	Read Phase Angle
PAUSE	Pause Subroutine
PAUSED?	Pause Query
PHSYNC	Synchronized Output Mode
PONSRQ	Power-On SRQ

R

Command	Description
RANGE	Set Output Range
RANGE?	Read Output Range
READY?	Ready Query
REFIN	Set Ref Frequency Input
REFIN?	Read Ref Frequency Input
REFOUT	Set Ref Frequency Output
REFOUT?	Read Ref Frequency Output
RESET	Reset HP 3245A
REV?	Revision Query
RMT	Remote
RQS	Set Service Request Enable
RQS?	Read Service Request Enable
RSTATE	Recall Stored State
RSTATEA	Recall Channel A State
RSTATEB	Recall Channel B State
RUN	Run Subroutine
RUNNING?	Running Query

S

Command	Description
SCRATCH	Delete All
SECURE	Calibration Security
SER?	Serial Number Query
SET	Send HP 3245A State
SET?	Read HP 3245A State
SET TIME	Set Time
SIZE?	Size Query
SRQ	Programmed Service Request
SSTATE	Store HP 3245A State
SSTATEA	Store Channel A State
SSTATEB	Store Channel B State
STA?	Status Word Query
STB?	Status Byte Query
STEP	Single Step Subroutine
STOREATOB	Copy Channel A State
STOREBTOA	Copy Channel B State
SYNCOUT	Set Sync Out
SYNCOUT?	Read Sync Out

T - W

Command	Description
TBn?	Trigger Bus n Query
TERM	Set Output Terminal
TERM?	Read Output Terminal
TEST	Confidence Test
TIME	Read Time
TRG	Pulse Trigger Bus
TRIGFREQ	Triggered Frequency Mode
TRIGIN	Set Trigger Input Source
TRIGIN?	Read Trigger Input Source
TRIGMODE	Set Trigger Mode
TRIGMODE?	Read Trigger Mode

For example, pressing the **Shift** key followed by the **C** key displays **CAL■** on the display, pressing the **Shift** key and then the **D** key displays **DCOFF■**, etc. That is, each MENU key provides an entry point into the menu command group. Once a menu command is displayed, you can edit the command by scrolling through the command list; or you can change the command parameters; or you can add required numeric parameters; or you can execute the command by using the four arrow keys and the **Back Space** key.

Controlling the Cursor

When a command is displayed which requires a parameter, a blinking cursor (■) is also displayed. The cursor position indicates where the next item entered will be displayed. You can also change the position of the cursor to delete all or part of a displayed command. Use the **Back Space** key to control the cursor position. Each time the **Back Space** key is pressed, the cursor position is backspaced one space <u>or</u> one word on the display. An example follows.

Example 6-6: Backspacing the Cursor

Key	Action	Display
⎓V	Display command and cursor.	APPLY DCV ■
Back Space	Backspace cursor.	APPLY DCV■
Back Space	Backspace cursor.	APPLY ■
Back Space	Backspace cursor.	APPLY■
Back Space	Backspace cursor.	0.000000E+0DCV

Scrolling Menu Commands

As noted, the MENU keys (**C**, **D**, etc.) allow you to enter the menu command set at the specified entry point. For example, pressing the **Shift** key followed by the **C** key displays CAL on the display, since the **CAL** command is the entry point for the **C** key.

You can read the value of any Menu command which returns data (such as **DCOFF?**) by using the **?** key. For example, entering **DCOFF** from the Menu and then pressing the **Shift** and **?** keys returns the DC offset value (0.000000E+00 at power-on/reset).

By using the **Up Arrow** and **Down Arrow** keys, you can scroll through the list of menu commands. Use the **Up Arrow** key to scroll toward the beginning of the alphabet with wraparound to the end of the alphabet. Use the **Down Arrow** key to scroll toward the end of the alphabet with wraparound to the beginning of the alphabet. An example follows.

Example 6-7: Scrolling Menu Commands

Key	Action	Display
	Shift keyboard to shifted mode.	0.000000E+0DCV*
M 	M menu entry point.	MEMAVAIL?■
	Scroll toward start of alphabet.	LOCAL■
	Scroll toward end of alphabet.	MEMAVAIL?■

* = SHIFT annunciator is ON.

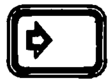
Scrolling Command Parameters

By using the **Up Arrow**, **Down Arrow**, **Left Arrow**, and **Right Arrow** keys, you can also scroll through the available parameters of a menu command or of any command shown on the display. You can also use the **Up Arrow** and **Down Arrow** keys to scroll the display by fields rather than by character.

For example, with the power-on display (0.000000E+0DCV), pressing the **Right Arrow** key changes the display to .000000E+0DCV (moves one character). However, pressing the **Down Arrow** key changes the display to FREQ 1000.00 (moves to the next field in the display). An example using the arrow keys for the **MON** command follows.

Example 6-8: Scrolling Command Parameters

In the example keystroke sequence, the **Right Arrow** key moves the cursor one space to the right which allows the **Up Arrow** or **Down Arrow** key to advance the **HELP** parameter to the next value. By repeatedly pressing the **Up Arrow** or **Down Arrow** keys, you can scroll through the **MON** parameters (**OFF**, **NONE**, **HPIB**, and **STATE**).

Key	Action	Display
	Shift keyboard to shifted mode.	0.000000E+0DCV*
M 	M menu entry point.	MEMAVAIL?■
	Scroll toward end of alphabet.	MON■
	Move cursor one space to right.	MON ■
	Display MON** parameter.	MON OFF■

* = SHIFT annunciator is ON.
** = repeatedly pressing the Up Arrow or Down Arrow key will scroll through all the MON parameters (OFF, NONE, HPIB, STATE).

Front Panel Functions

"Front Panel Operation" showed how to control the operation of the front panel and how to enter/edit commands. This section describes the front panel functions for the HP 3245A, including:

- Basic front panel functions.
- Generating outputs/waveforms.
- Advanced operations functions.

Basic Front Panel Functions

The following paragraphs summarize some basic front panel functions, including setting local mode, setting the HP-IB address, using error HELP/query functions, and redefining front panel keys.

Setting Local Mode

At power-on/reset, the HP 3245A is set for front panel (local) operation. However, if the HP 3245A has been programmed via HP-IB, it may be in remote operation (REM annunciator is ON). In this case, you can return the HP 3245A to front panel operation by pressing the **Local** key. This also turns the REM annunciator OFF, indicating local operation.

Setting HP-IB Address

The HP 3245A HP-IB address is set for 09 at the factory. As required, you can change the HP-IB address (range is 01 to 31) by using the **ADDRESS** command (menu entry) followed by the appropriate numeric keys. This keystroke sequence sets the HP-IB address to 17.

NOTE

You can check the address by pressing the **Shift** *key followed by the* **Address** *key. The address is displayed on the display. (Reset the HP-IB address to 09 to run the example programs in this manual.)*

Example 6-9: Setting HP-IB Address

Key	Action	Display
	Shifts keyboard to shifted mode.	0.000000E+0DCV*
C Freq	Select C menu entry point.	CAL■
⇧	Scroll to ADDRESS command.**	ADDRESS■
1	Enter 1.	ADDRESS 1■
7	Enter 7.	ADDRESS 17■
Enter	Set HP-IB address to 17.	0.000000E+0DCV

* = SHIFT annunciator is ON.
** = Scroll to ADDRESS command with Up Arrow key.

Reading Error Messages

Whenever the HP 3245A detects an error, the ERR annunciator turns ON and the word ERR and the error message are displayed. The error message consists of a code number and a message. You can read the error message by using the **Right Arrow** key or by using the **Error** key. After you use the **Right Arrow** key to read the error, you can clear the display with the **Enter** key. Note, however, that this does NOT turn off the ERR annunciator and the error message is still stored (The HP 3245A stores the last four errors which have occurred.)

The **Error** key scrolls through the first four error messages stored Pressing the **Shift** key followed by the **Error** key displays the first error stored. Then, again pressing these two keys displays the second error, etc. until all error messages (up to four) have been displayed. For example, if two error messages are stored, the third time the keystroke sequence is executed, "0: NO ERROR" is displayed. Note that to clear all error(s), you must use the **Error** key once for <u>each</u> error or use the **Clear** key. An example follows.

Example 6-10: Reading an Error Message

Key	Action	Display
⎓V	Sets HP 3245A to output DC voltage.	APPLY DCV ■
Enter	Gives error message.*	ERR 2:SYNT
	Shift keyboard to** shifted mode.	ERR 2:SYNT
Error	Enter error *** message.	2,"SYNTAX

* = Press Right Arrow key to read message. ERR ann ON.
** = ERR and SHIFT annunciator are ON.
*** = Press Right Arrow key to read message. ERR ann OFF.

Using the HELP Function

The HELP function provides a front panel display of command syntax statements and brief descriptions for some HP 3245A commands. It also provides explanations of some command parameters. The HELP syntax is **HELP** [*topic*], where *topic* is an alphabetical listing of some HP 3245A commands. The *topics* are scrolled using the **Up Arrow** and **Down Arrow** keys. An example using the HELP function follows.

Example 6-11: Using the HELP Function

Key	Action	Display
	Shifts keyboard to shifted mode.	0.000000E+0DCV*
Help E	Display HELP function.	HELP ■
⇩	Display HELP ABORT.	HELP ABORT■
Enter	Executes HELP ABORT.**	ABORT: ABORT ;

* = SHIFT annunciator is ON.
** = Use Right Arrow key to read remainder of message.

Redefining Numeric Keys

You can assign a string of one or more commands to each of the MUMERIC keys (**0**, **1**, **2**, ..., **9**) by using the **Def Key** key. After a string is assigned to a key (maximum string is 40 characters), pressing that key displays the string on the display and pressing the **Enter** key executes the string.

To restore a redefined key to its original function, use **Def Key** followed by the number of the key to be restored, the , key, and the **Enter** key. For example, to restore NUMERIC key **5** to its original (default) function, use the **Shift**, **Def Key**, **5**, **,**, and **Enter** keys (If you use the HP 3245A with a controller, enter OUTPUT 709; "DEFKEY DEFAULT" to restore ALL NUMERIC keys to the default condition) An example to redefine a NUMERIC key follows

NOTE

A keyboard overlay was shipped with your HP 3245A. If desired, you can write on this overlay to identify the function assigned to each user-defined NUMERIC key. To install the overlay, insert the left tab of the overlay into the left side of the **5** *key collar. Then, bend the overlay and press the right tab into the right side of the* **5** *key collar.*

Example 6-12: Redefining a Numeric Key

This keystroke sequence assigns **APPLY DCV 1;IMP 0** to the NUMERIC **3** key Then, when the **3** key and the **Enter** key are pressed, channel A outputs a 1.0 VDC signal and the channel is set for 0 Ω output impedance. Note, however, that the **1** key still retains its numeric "1" function. For example, if the **Sine Wave** key and then the **3** key are pressed, **APPLY ACV 3** appears on the display.

Key	Action	Display
(shift)	Shifts keyboard to shifted mode.	0.000000E+0DCV*
Def Key .	Sets HP 3245A to redefine a key.	DEFKEY ■
3	Sets 3 key to be redefined.	DEFKEY 3■
,	Begin string for the 3 key.	DEFKEY 3, "■
V	Assigns APPLY DCV to the 3 key.	3, "APPLY DCV ■
1	Assigns APPLY DCV 1 to the 3 key.	3, "APPLY DCV 1■
(shift)	Set shifted mode.	3, "APPLY DCV 1■
; ,	Semicolon to separate commands.	, "APPLY DCV 1;■
Imp	Continue string.	PLY DCV 1;IMP ■
0	Continue string.	LY DCV 1;IMP 0■
Enter	Store new def.	0.000000E+0DCV**

* = SHIFT annunciator is ON.
** = End quotation mark automatically added.

Generating Outputs/ Waveforms

The following paragraphs summarize DC output/AC waveform generation using front panel operation. Refer to Chapter 10 - Programming DC Outputs for details on DC output generation; to Chapter 11 - Programming Defined Waveforms for AC waveform generation; or to Chapter 13 - Triggering Outputs/Waveforms for triggering techniques.

Generating DC Outputs

You can generate DC voltage or current outputs using the front panel keys or using menu operation. There are three main steps to generate DCV or DCI outputs:

- Select output function (DCV or DCI).
- Select channel parameters required.
- Select type of triggering (if any).

Refer to Chapter 10 - Programming DC Outputs for information on generating DCV or DCI outputs. An example keystroke sequence to generate a DC voltage output follows.

Example 6-13: Generating DC Voltage Output

This keystroke sequence outputs -1.0 VDC from channel A.

Key	Action	Display
mV	Sets HP 3245A for DC voltage output.	APPLY DCV ■
-	Enters "-".	APPLY DCV -■
1	Enters "1".	APPLY DCV -1■
Enter	Outputs -1.0 DCV	-1.000000E+0DCV

Generating AC Waveforms

You can output sine, ramp, square, or arbitrary waveforms using the front panel keys or by using menu operation. There are three main steps to generate AC waveforms:

- Select waveform function (sine, ramp, square, arb).
- Select channel/waveform parameters required.
- Select triggering technique (as required).

Refer to Chapter 11 - Generating Defined Waveforms for information on sine, ramp, and square waveform generation. Refer to Chapter 12 - Generating Arbitrary Waveforms for information on arbitrary waveform generation. Refer to Chapter 13 - Triggering Outputs/Waveforms for details on triggering AC waveforms. An example keystroke sequence to generate a sine waveform follows.

Example 6-14: Generating AC Sine Waveform

This keystroke sequence outputs a 1.0 Vac PP sine waveform. To output a 1.0 Vac PP ramp wave, use the **Sine Wave** key. Or, to output a 1.0 Vac PP square waveform, use the **Square Wave** key. If you measure the RMS value of the output with an HP 3458A DMM, the results will be 0.3535 Vac for the sine waveform, 0.289 Vac for the ramp waveform, and 0.500 Vac for the square waveform.

Key	Action	Display
∿V	Sets HP 3245A to output AC sine waveform.	APPLY ACV ■
1	Sets output value to 1.0 Vac PP.	APPLY ACV 1■
Enter	Outputs 1.0 Vac PP sine waveform.	1.000000E+0ACV

Triggering Outputs/Waveforms

You can use a variety of triggering techniques to generate triggered DC outputs or AC waveforms. Refer to Chapter 10 - Programming DC Outputs for DC outputs; to Chapter 11 - Programming Defined Waveforms for sine, ramp, and square wave AC waveforms; or to Chapter 12 - Generating Arbitrary Waveforms for triggering methods for Arbitrary Waveforms. Also, refer to Chapter 13 - Triggering Outputs/Waveforms for a general discussion of triggering methods.

Advanced Operations

The following paragraphs summarize some advanced operations topics for front panel operation, including dimensioning/editing arrays; subroutine operations; generating arbitrary waveforms; storing/recalling states; and input/output operations.

Dimension/ Edit Arrays

You can use the **DIM Array** and **Edit Array** keys to define up to 10 arrays (called A0 through A9) from the front panel. The defined arrays can be used for triggered DCV/DCI outputs (refer to Chapter 10 - Programming DC Outputs) or for arbitrary waveform outputs (refer to Chapter 12 - Programming Arbitrary Waveforms).

Dimensioning Arrays

To dimension (define) an array from the front panel, press the **Shift** and **DIM** arrays. (An array must be dimensioned before it can be edited.) With power-on conditions, when **DIM Array** is executed, **DIM A0 (2047)** appears on the display which shows array A0 is a 2048-element array (filled with zeroes at power-on).

To dimension an array, use the arrow keys to position the cursor to the number of elements field and enter the desired number of elements for the array For example, using DIM A0 (0003) defines A0 as a 4-element array. Then, to define array A1, again execute the **DIM Array** key, change the display to **DIM A1 (2047)** and define the number of elements for A1. Repeat for arrays A2, ..., A9 as required.

NOTE

To use an array for triggered DCV/DCI, you can dimension the array for 2 to 2048 elements. However, an array used for arbitrary waveform generation must have 2048 elements. Attempting to dimension more than four 2048-element arrays may result in an "OUT OF MEMORY" error.

Editing Arrays

When an array has been dimensioned, you can define the array values with the **Edit Array** key. For example, suppose A0 was previously defined as a 4-element array. Then, when the **Edit Array** key is pressed, **A0 (0003) = 0.000** appears on the display. Use the arrow keys to move the cursor to the value field (0.000) and enter the desired value. For example, if you want to assign 1.256 to element 0, enter 1.256 in the value field and press the **Enter** key.

When the value has been entered into the first (0) array element, the element number is automatically advanced to the second (1) array element. Repeat the process to assign a value to the second element, etc. When you have assigned values to all elements required, press any key to return the display to its power-on condition.

Example 6-15: Triggered DC Voltage Output

This keystroke sequence outputs triggered DC voltages, using the voltage values stored in array A0. A0 is dimensioned as a 3-element array with element values A(0) = 0.0, A(1) = 1.0, and A(2) = 2.0. The first value in array A0 (0.0 VDC) is immediately output when the **APPLY DCMEMV** command is executed The remaining element values are output (one at a time) each time the **Trig In** key is pressed. (The first value of 0.0 VDC is again output the third time the **Trig In** key is pressed.)

Key	Action	Display
Dimension Array A0		
	Set shifted mode	0.000000E+0DCV
DIM Array Edit Array	Def array A0.	DIM A0(2047)
⇨	Position cursor.	DIM A0(2047)
(3X) 0	Define size (use (key 3 times).	DIM A0(0007)
2	A0 is 3 elements.	DIM A0(0002)
Enter	Power-on display.	0.000000E+0DCV

Key	Action	Display
Edit Array A0		
Edit Array	Edit first element.	A0(0000) = 0.000
Enter	Define element 0 value = 0.	A0(0001) = 0.000
(6X) ⇨	Position cursor (use key 6 times).	A0(0001) = 0.000
1	Define element 1 value = 1.	A0(0001) = 1.000
Enter	Enter element 1 value.	A0(0002) = 0.000
⇦	Position cursor.	A0(0002) = 0.000

Edit Array A0 (cont'd)

2	Define element 2 value = 2.	A0(0002) = 2.000
Enter	Enter element 2 value.	A0(0002) = 2.000
(ANY KEY)	Power-on display (can use any key).	0.000000E+0DCV

Key	Action	Display
Output Triggered DCV		
	Set shifted mode.	0.000000E+0DCV
DC MemV / mV	Set triggered DCV mode.	APPLY DCMEMV
3	Triggered DCV with 3 elements.	APPLY DCMEMV 3
,	Add comma.	PPLY DCMEMV 3,
⇩	Use values in A0.	LY DCMEMV 3,A0
Enter	Output first value (0.0 VDC).	-------- DCMEMV
Trig In	Trigger channel. (sets TRIGIN SGL)	TRIGIN
Enter	Output second value (1.0 VDC).	-------- DCMEMV

Using Subroutines

If a subroutine has been stored in the HP 3245A via HP-IB, you can use any of the subroutine operations (such as **CALL**) or subroutine commands (such as **CAT** or **LIST**) by using the front panel MENU keys and menu operation. Refer to Chapter 7 - Arrays and Subroutines for details on using HP 3245A subroutines. Then, use menu operation to enter and execute the subroutine command(s) needed.

NOTE

You cannot use front panel operation to define a subroutine. The subroutine must be defined in a controller program and downloaded to the HP 3245A.

Generating Arbitrary Waveforms

Arbitrary (user-defined) ACV/ACI waveforms can be generated from the front panel using the **APPLY WFV** or **APPLY WFI** command. Arbitrary ACV waveforms can be generated from the front panel using the **Arbitrary Wave** key. Since the syntax for arbitrary waveforms is **APPLY WFV** *pp_amplitude* [*,array_name*], for front panel operation you must define an array (A0 through A9) to store the values to be output.

The array MUST contain 2048 points, with each value between -1 and +1, inclusive. Then, when the waveform is output, the value in each point is multiplied by the peak value of the peak-to-peak amplitude specified by *pp_amplitude* to arrive at the actual output values. Refer to Chapter 12 - Programming Arbitrary Waveforms for details on using **APPLY WFV**.

The keystroke sequence to generate an arbitrary waveform is similar to that shown in Example 6-17 except that the array must be defined for 2048 points, each point must have a value between -1 and +1, and **APPLY WFV** *pp_amplitude* is used rather than **APPLY DCMEMV** *length*, where *length* is the number of values to be output.

Storing/Recalling States

The hardware state of a channel can be stored in continuous (non-volatile) memory and the stored state can then be recalled (and the state of the channel restored) in the event of a power failure. Since states stored in continuous memory are not destroyed when power is removed, you can restore the channel to the state existing before power-down with a single command, rather than having to reenter all channel parameter settings. Refer to Chapter 9 - Other Instrument Functions for details.

Input/Output Operations

As required, refer to Chapter 8 - Input/Output Operations for a discussion of HP-IB communication, local/remote operation, and interrupts/service requests.

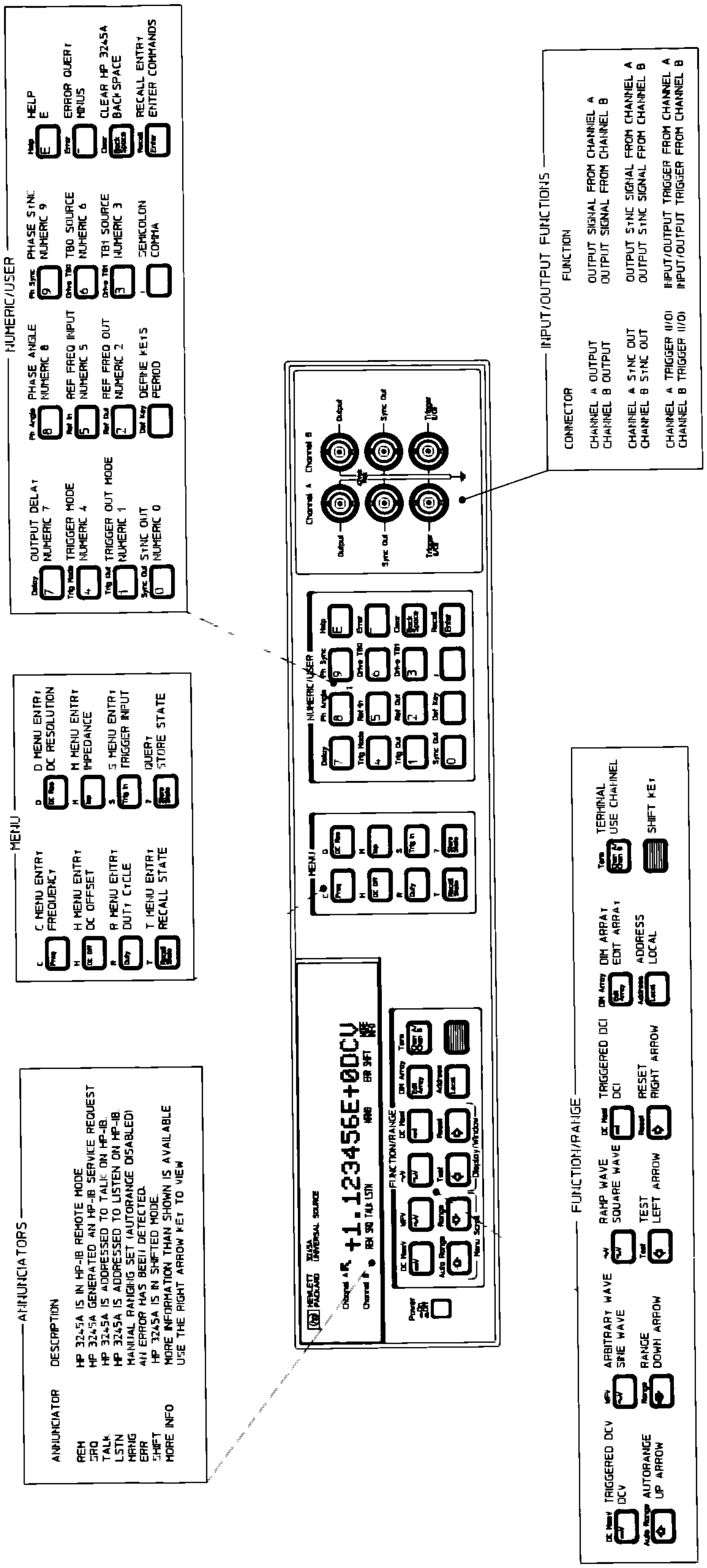


Figure 6-1. HP 3245A Front Panel Features

Contents

Chapter 7
Arrays and Subroutines

Chapter 7
Arrays and Subroutines

Chapter Contents

This chapter shows how to use general purpose math, variables, arrays, and subroutines for HP 3245A operation. The chapter contents are:

- **General Purpose Math** describes math operators and functions which apply to HP 3245A operation. It includes math operators and functions, math heirarchy, number ranges, and math error messages.

- **Variables and Arrays** shows how to define variables and arrays, how to enter and read variable and array values, and how to purge arrays and variables. It also shows how to use memory mode to store information in arrays and variables.

- **Subroutines** shows how to define and delete subroutines, shows how to use conditional and looping commands, and describes subroutine operations.

General Purpose Math

This section describes the math functions for the HP 3245A, including the following items. Table 7-1 summarizes the general purpose math operators and functions for the HP 3245A

- Math Operators
- Math Functions
- Math Hierarchy
- Number Ranges
- Math Errors

Math Operators

Math operators for the HP 3245A include:

- General math operators
- Relational operators
- Logical operators

General Math Operators (+, -, *, /, ^, DIV, MOD)

General math operators include +, -, *, /, and ^ plus **DIV** (integer division) and **MOD** (modulo) (Write unary minus operations as A = 0-B.) Integer division (**DIV**) returns the integer portion of the quotient. That is, normal division takes place, but all digits to the right of the decimal point are truncated (not rounded). For example, 7 **DIV** 3 returns 2, since 7/3 = 2.33 and .33 is truncated.

The modulo (**MOD**) operator returns the remainder resulting from a division. As with **DIV**, normal division takes place, but, with **MOD**, only the remainder is returned. For example, 7 **MOD** 3 returns 1 since the division result of 7/3 is 2 with a remainder of 1.

Relational Operators (<, >, <=, >=, =, <>)

Relational math operators (<, >, <=, >=, =, and <>) are allowed in any expression

Table 7-1. Math Operators/Functions

Math Operators		Math Functions	
General Math Operators		**General Math Functions**	
+	Addition	ABS	Absolute Value
-	Subtraction	SQR	Square Root
*	Multiplication	**Trigonometric Functions**	
/	Division	ATN	Arctangent
^	Exponentiation	COS	Cosine
DIV	Integer Div	SIN	Sine
MOD	Modulo	**Logarithmic Functions**	
Logical Operators		EXP	Natural Antilog
AND	Inclusive-AND	LGT	Common Log
EXOR	Exclusive-OR	LOG	Natural Log
OR	Inclusive-OR	**Binary Functions**	
NOT	Logical Comp	BINAND	Binary AND
		BINCMP	Binary Comp
		BINEOR	Binary Excl-OR
		BINIOR	Binary Incl-OR
		BIT	Read Bit Value
		ROTATE	Rotate Bits
		SHIFT	Shift Bits

Logical Operators (AND, EXOR, OR, NOT)

There are four logical operators: **AND** (inclusive-AND), **OR** (inclusive-OR), **EXOR** (exclusive-OR), and **NOT** (logical inverse). The first three operators compare the two arguments and return a "0" or a "1" based upon the appropriate truth table. Any non-zero value (positive or negative) in an argument is considered a logical "1". Only zero is treated as a logical "0". Logical operators are allowed in any expression.

The logical operator commands have the following forms. Each *argument* parameter can be a number, numeric variable, math function, array element, or numeric expression (enclosed in parentheses). Table 7-2 shows the truth tables for the four functions.

argument **AND** *argument*	(**AND**)
argument **EXOR** *argument*	(**EXOR**)
argument **OR** *argument*	(**OR**)
NOT *argument*	(**NOT**)

Table 7-2. Logical Operators - Truth Tables

AND

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

NOT

A	NOT A
0	1
1	0

OR

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

EXOR

A	B	A EXOR B
0	0	0
0	1	1
1	0	1
1	1	0

Math Functions

Math functions for the HP 3245A include:

- General Math Functions.
- Trigonometric Functions.
- Logarithmic Functions.
- Binary Functions.

General Math Functions (ABS, SQR)

General math functions for the HP 3245A are absolute value (**ABS**) and positive square root (**SQR**). The form for absolute value is **ABS** (*argument*) and the form for square root is **SQR** (*argument*). The argument may be a number, numeric variable, function, array element, or numeric expression (enclosed in parentheses). For example, **ABS (10/-2)** returns 5 and **SQR (2.345)** returns 1.5313393.

Logarithmic Functions (EXP, LGT, LOG)

The HP 3245A can compute both natural and common logarithms. **EXP** (*argument*) raises e to the power of the *argument*. **LGT** (*argument*) computes the common logarithm of the (positive) *argument* to the base 10. **LOG** (*argument*) computes the natural logarithm of the (positive) *argument* to the base e.

For example, **EXP(10)** raises e to the 10th power and returns 22026.466. **LGT(15)** computes the logarithm (base 10) of 15 and returns 1.1760913. **LOG(2.345)** computes the logarithm (base e) of 2.345 and returns 0.8522854.

Trigonometric Functions (ATN, COS, SIN)

Three trigonometric functions are available: arctangent (**ATN**), cosine (**COS**), and sine (**SIN**). The form is **ATN** (*argument*), **COS** (*argument*), or **SIN** (*argument*), where *argument* is the angle in radians within the range $\pm$ 2.98156826 E+8. For example, **SIN(0.2018)** computes the sine of 30 degrees (0.5235988 radians) and returns 0.5.

Binary Functions (BINAND, BINCMP, BINEOR, BINIOR, BIT, ROTATE, SHIFT)

There are seven binary functions which are useful for digital pattern generation. When using binary functions, the *argument* values of REAL variables are rounded to integers within the range -32768 to +32767. Table 7-3 summarizes binary functions for the HP 3245A.

In Table 7-3, the *argument* parameters are specified by A and B. For the **SHIFT** and **ROTATE** commands, if the displacement (*disp*) is positive, rotating or shifting is toward the least significant bit (bit 0). If the displacement is negative, rotating or shifting is toward the most significant bit (bit 15). Within the HP 3245A, the *argument* is represented as a 16-bit 2's complement integer with bit 0 = lsb and bit 15 = msb.

Math Hierarchy

Table 7-4 shows math hierarchy for the HP 3245A. The instrument evaluates parenthetical expressions before evaluating any math functions outside of parentheses. If two or more operations of the same priority are in the expression, the hierarchy is from left to right.

Table 7-4. Math Hierarchy

Highest Priority	Parentheses
	Functions: SIN, COS, etc.
	Exponentiation
	*,/, MOD, DIV, +, -
Lowest Priority	Relational operators (<, >, etc.) and logical operators (AND, OR, etc.)

Table 7-3. Binary Functions

Function/Arg	Example
BINAND(A/B) Bit-by-bit logical AND of arguments.	A = 0000 0000 0000 1100 B = 0000 0000 0000 1001 ------------------- BINAND(A,B) = 0000 0000 0000 1000
BINCMP(A) Bit-by-bit binary complement (NOT) of the argument.	A = 0000 0000 0000 1100 BINCMP(A) = 1111 1111 1111 0011
BINEOR(A,B) Bit-by-bit logical Exclusive-OR (EXOR) of the arguments.	A = 0000 0000 0000 1100 B = 0000 0000 0000 0110 ------------------- BINEOR(A,B) = 0000 0000 0000 1010
BINIOR(A,B) Bit-by-bit logical Inclusive-OR (OR) of the arguments.	A = 0000 0000 0000 1100 B = 0000 0000 0000 0110 ------------------- BINIOR(A,B) = 0000 0000 0000 1110
BIT(A,pos) Reads logic state (0 or 1) of the specified bit in the argument.	A = 0000 0000 0000 1000 BIT(A,4) = 0 BIT(A,3) = 1
ROTATE (A,disp) Rotate argument a specified number of positions (with wraparound).	A = 0000 0000 0000 1100 ROTATE(A,3) = 1000 0000 0000 0001 ROTATE(A,-3) = 0000 0000 0110 0000
SHIFT(A,disp) Shift argument a specified number of positions (no wraparound).	A = 0000 0000 0000 1100 SHIFT(A,3) = 0000 0000 0000 0001 SHIFT(A,-3) = 0000 0000 0110 0000

Number Ranges

Two types of numbers are used in the HP 3245A: INTEGER and REAL. The valid range for REAL numbers is from -1.797 693 134 862 315 E+308 through +1.797 693 134 862 315 E+308. The smallest non-zero REAL number allowed is 2.225 073 858 507 202 E-308. (A REAL number can have a value of zero.) An INTEGER number can have any whole-number value from -32768 through +32767.

The HP 3245A automatically converts between REAL and INTEGER values as required. When REAL numbers are converted to INTEGER representations, information may be lost due to rounding errors and/or range errors. A rounding error may occur when a REAL number is converted to an INTEGER number, since the REAL value is rounded to the closest INTEGER value and all information to the right of the decimal point is lost.

A range error may also occur when converting REAL numbers to INTEGER values. While the range of REAL values is from approximately -1E+308 to +1E+308, the INTEGER range is only from -32768 to +32767 (about -1E+4 to +1E+4). Thus, not all REAL numbers can be rounded to an equivalent INTEGER value, which may generate an "Integer Overflow" error (refer to the following subsection "Math Errors").

Math Errors

Table 7-5 shows the errors which may occur when the HP 3245A is evaluating a math expression. When a math error occurs, **ERROR 94, "MATH ERROR"** followed by the appropriate string shown in Table 7-5 is displayed.

Table 7-5. Math Errors

Error Description	Numeric Values
Division by Zero	Divisor = 0
REAL Overflow	>+1.797 693 134 862 315 E+308
REAL Underflow	<-1.797 693 134 862 315 E+308
INTEGER Overflow	<-32,768 or >32,767
Sq Root of Neg Number	Number <0
Log of Non-Pos Number	Number <0
Illegal REAL Number*	<+-2.225 073 858 507 202
Trig Arg Out of Range**	Outside +-2.98156826 E+8
BCD Exponent Too Big	
HEX/Octal/Dec Arg Error	
* = for non-zero REAL numbers	
** = number in radians	

Variables and Arrays

This section shows how to use variables and arrays for HP 3245A operation, including the items shown. Figure 7-1 summarizes commands used for variable and array operations.

- Defining Variables/Arrays
- Entering Array Values
- Reading Variable/Array Values
- Purging Variables/Arrays
- Using Memory Mode

Defining Variables/ Arrays

Variables and arrays for the HP 3245A can be defined as REAL or INTEGER. **REAL** defines a REAL variable or array, while **INTEGER** defines an INTEGER variable or array. In addition, **DIM** can be used to define a REAL array. The **LET** command defines a variable and assigns a value to the variable. When the **MEM** command is used, the HP 3245A automatically defines a REAL variable or array (refer to "Using Memory Mode" for details on the **MEM** command).

A variable contains one element, while an array contains one or more elements. For example, **DIM A** defines a REAL variable A and **INTEGER B(9)** defines a 10-element INTEGER array B. Variable and array use in the HP 3245A is similar to that in an enhanced BASIC language, except that the HP 3245A does not provide string variables. All variables and arrays are global among front panel, HP-IB, and subroutine operations.

Variable and array names may contain up to 10 characters. The first character must be a letter (A-Z) but the remaining nine characters may be letters, numbers (0-9), the underscore character (_), or the question mark (?). Variable or array names must not be the same as HP 3245A commands or parameters, or stored state names.

Defining Variables

The **REAL, INTEGER, LET,** and **MEM** commands can be used to define variables. For variables, the **REAL** *name* command defines a REAL variable, while the **INTEGER** *name* defines an INTEGER variable. For example, **REAL A** defines a REAL variable A, while **INTEGER IA** defines an INTEGER variable.

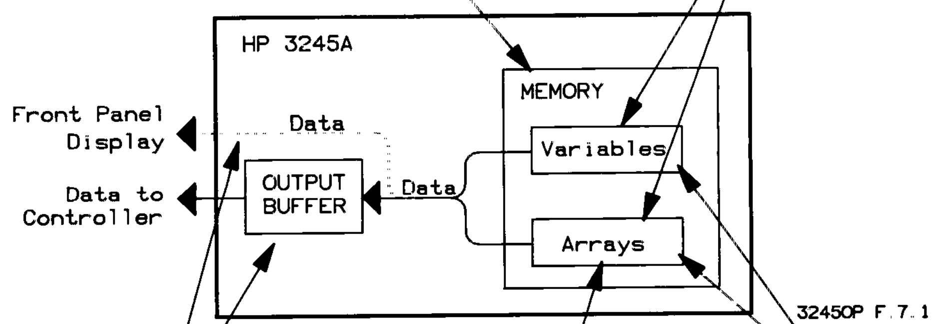
The **[LET]** *variable = expression* command can be used to define a variable and assign a value. Unless the variable was previously defined as INTEGER, a REAL variable is created by **LET**. For example, since the **LET** keyword is optional, **LET A = 2.3** or **A = 2.3** both create a REAL variable A with value of 2.3. The **[LET]** variable parameter can be a variable or an array name with a numeric expression index such as **A(4)** or **A(I+4)**. Also, logical expressions such as **A=(B>C)** are allowed.

Memory Mode Operation

With memory mode ON (MEM), results from the next command to generate data are stored in a previously defined variable or array, rather than being sent to the output buffer or to the front panel display. Use MEMAVAIL? to read available memory storage size

Define Variables/Arrays

Define a REAL variable or array with REAL or DIM. Define an INTEGER variable or array with INTEGER. Variables are automatically created by an assignment (LET) or memory (MEM) command. Arrays use base 0 (first element number is 0).



Read Variable/Array Values

Use FETCH or VREAD to read the value of a variable. Use VREAD to read the array or array element values. Use SIZE? to read the number of elements in an array. When memory mode is off (MEM OFF), the variable or array values are returned to the output buffer or to the front panel display.

Fill Arrays

Enter values into a previously defined array with a FILL, FILLAC, FILLBIN, FILLRP, or FILLWF command. FILL fills the array with user-specified decimal values while FILLBIN fills array with binary values. For FILLAC and FILLRP, the array must be an INTEGER array of 2048 elements. For FILLWF, the INTEGER and REAL arrays must be 2048 elements.

Delete Variables/Arrays

Defined variables, arrays, and subroutines are stored in volatile memory, while instrument states can be stored in volatile or continuous memory. Use SCRATCH to delete defined arrays, variables and subroutines from volatile memory.

Figure 7-1. Variables/Arrays Overview

Some typical program statements follow to define variables. Note that lines 60 and 70 are equivalent.

```
        .
        .
40  OUTPUT 709;"REAL A"        !Defines REAL variable A
50  OUTPUT 709;"INTEGER B"     !Defines INTEGER variable B
60  OUTPUT 709;"C=4"           !Assigns 4.0 to REAL variable C
70  OUTPUT 709;"LET C=4"       !Assigns 4.0 to REAL variable C
        .
        .
```

Defining Arrays

Arrays can be defined with a **REAL**, **DIM**, **INTEGER**, or **MEM** command. Arrays use base 0, so the first element in the array is element 0. Both **DIM** and **REAL** define REAL arrays, which **INTEGER** defines an INTEGER array.

Executing **DIM** *array_name (max_index)* [,*array_name (max_index)*,..] or **REAL** *name*[(*max_index*)] [,*name*[(max_index)],...] defines a REAL array and fills all elements with zeros. Executing **INTEGER** *name* [(*max_index*)] [,(*name* [(*max_index*)], ...] defines an INTEGER array and fills all elements with zeros.

The *array_name* or *name* parameter specifies the array name, while the *max_index* specifies the number of elements in the array. The valid range for *max_index* is 0 to 32767. Since the lower bound for all HP 3245A arrays is 0, the number of elements in the array is one more than *max_index*. For example, **REAL A(5)** reserves space in memory for REAL array A with six elements (0 through 5). Two typical program statements follow to define arrays.

```
        .
        .
40  OUTPUT 709;"REAL A(2047)"    !Defines REAL array A of 2048 elements
50  OUTPUT 709;"INTEGER B(5)"    !Defines INTEGER array B of 6 elements
        .
        .
```

NOTE

Arrays A0 through A9 can be defined using the front panel **DIM Array** *key. Refer to Chapter 6 - Front Panel Operation for details.*

Entering Array Values

When an array has been defined with an **DIM**, **REAL**, or **INTEGER** command, you can use one of several commands to enter values into the array. Use the **FILL** to fill an array with decimal values or use the **FILLBIN** command to fill an array with binary values.

You can also use the **FILLAC**, **FILLRP**, and **FILLWF** commands to scale (precompute) and store values in an array for faster output. Refer to Chapter 12 - Programming Arbitrary Waveforms for details on the **FILLAC**, **FILLRP**, and **FILLWF** commands.

NOTE

Values can also be entered into arrays using a **FOR..NEXT** *loop within a subroutine. Refer to "Subroutines" for details.*

Entering Decimal Values Using the FILL Command

The **FILL** *array_name, list* enters the values specified by *list* into the array specified by *array_name*. The first item in *list* is entered into element 0, the second item in element 1, etc. For example, the following program seqment enters -5.0 in element 0 of array VOUT, 0.0 in element 1, +5.0 in element 2 and 10.0 in element 3.

```
        .
40  OUTPUT 709;"DIM VOUT(3)"                  !Define REAL array of 4 elements
50  OUTPUT 709;"FILL VOUT -5.0,0.0,5.0,10.0   !Enter values into array
        .
```

Entering Binary Values Using the FILLBIN Command

The **FILL** command enters decimal values into the defined array. To place binary values into a previously defined array, use the **FILLBIN** command. **FILLBIN** *array_name, block_data* enters binary data into the array specified by *array_name*. The *block_data* parameter consists of a header which includes the #sign, the letter A, and a 16-bit integer which indicates the number of bytes of data followed by the data to be stored. Refer to the **FILLBIN** command in the HP 3245A Command Reference Manual for an example program using **FILLBIN**.

Reading Variable/Array Values

When a variable or array has been defined, you can read the element values by using **FETCH** or **VREAD**. Use the **FETCH** or **VREAD** command to read the value of a variable. Use **VREAD** to read the value of an array element or of the entire array. Use **SIZE?** to return the number of elements in an array.

Reading Variable Value Using FETCH

The **FETCH** *variable* command returns the value of the specified variable. For example, if A is a REAL variable with value 300, **FETCH A** returns 300.0. (Note that **FETCH** is not a front-panel command.) **FETCH** can also be used to return expressions or quoted ASCII strings. Refer to the **FETCH** command in the HP 3245A Command Reference Manual for example programs.

Reading Variable/Array Value Using VREAD

Use **VREAD** *variable* to read the value of a variable. For example, **VREAD A** reads the value of variable A. Use **VREAD** *array_name* [(*element*)] to read the value of an array or a specified element. When *element* is not specified, **VREAD** returns the values of all elements in the array. When *element* is specified, **VREAD** returns the value of the specified element. For example, **DIM VOUT(3)** defines a 4-element array. Then, **VREAD VOUT(2)** returns the value of element #2, while **VREAD** would return the value of each of the elements in VOUT. Refer to the **VREAD** command in the HP 3245A Command Reference Manual for an example program.

Reading Array Size Using SIZE?

To read the number of elements in a defined array, use the **SIZE?** *array_name* command. **SIZE?** returns the number of elements in the specified array. The number is one more than the index number of the array since the first element number is 0. For example, **INTEGER VOLTS(9)** defines a 10-element array with elements 0, 1, ...,9. **SIZE? VOLTS** then returns 10.

Purging Variables/ Arrays

The HP 3245A has two memory storage modes: volatile and continuous (non-volatile). Variables, arrays, and subroutines are stored in volatile memory, while instrument states and other information is stored in continuous (nonvolatile) memory. When power is turned off, all information in volatile memory is deleted, but information in continuous memory is retained. You can use the **SCRATCH** command to delete variables and arrays from memory or use the **MEMAVAIL?** command to read the amount of available memory.

Purging Variables and Arrays Using SCRATCH

Use the **SCRATCH** command to purge (scratch) all user-defined variables, arrays, and subroutines from volatile memory. Note that **SCRATCH** deletes all arrays, variables, subroutines, and stored states from memory and removes the name definitions. (Contrast with **DELSUB** in the next section "Subroutines".)

Reading Available Memory Using MEMAVAIL?

As HP 3245A memory blocks are created and purged, memory becomes fragmented into many small blocks. **MEMAVAIL?** returns the size of the largest block of volatile memory available. The **MEMAVAIL?** command returns the largest block (number of bytes) of volatile memory available. If no memory has been used, approximately 80,248 bytes of volatile memory are available.

Using Memory Mode

With memory mode OFF (**MEM OFF**), a query command such as **VREAD** or **SIZE?** returns the results to the output buffer or to the front panel display, depending on where the command originated (HP-IB or front panel). However, by turning memory mode ON with an **MEM** command, the data returned by the query command can be stored in a specified variable or array.

Using Variables in Memory Mode With MEM variable

Unless previously defined as an INTEGER variable, **MEM** *variable* creates a REAL variable. When **MEM** is executed, the next command to return data stores the results in the variable defined by **MEM** rather than sending the result to the output buffer or front panel display. For example, **MEM A** creates REAL variable A. The next data-generating command will store the result in A.

Using Arrays in Memory Mode With MEM array

Unless previously defined as an INTEGER array, **MEM** *array_name* [(*start_index*)] creates a REAL array. When **MEM** is executed, the next command to return data stores the results in the array defined by **MEM** rather than sending the result to the output buffer or front panel display.

The optional *start_index* parameter (range is 0 to 32767) allows you to begin storing data at a specific location in the array (default *start_index* = 0) For example, with **MEM A(4)** set, the results from the next data-generating command are stored in A(4), A(5), etc. Refer to the **MEM** command in the HP 3245A Command Reference Manual for an example program.

NOTE

*If memory mode is set, always turn memory mode OFF (**MEM OFF**) before using a **VREAD** command. Otherwise, **VREAD** attempts to write information into the memory, rather than returning the information to the output buffer, and an error occurs. Memory mode is automatically disabled as required for variable storage, but NOT for array storage.*

Subroutines

The HP 3245A can store and execute BASIC language subroutines which are downloaded into memory from a controller such as an HP 9000 Series 200/300 (or equivalent). This section shows how to define, execute, and delete subroutines and how to use conditional statements. It includes:

- Defining/deleting subroutines
- Executing subroutines
- Using conditional statements

Subroutines Overview

An HP 3245A subroutine is a set of commands beginning with **SUB** and ending with **SUBEND**. **SUB** *sub_name* assigns a subroutine name which is used to execute the subroutine at a later time. Subroutines are stored in HP 3245A volatile memory. Any HP 3245A command may be stored and executed inside a subroutine. Three conditional and looping commands are provided for use within subroutines.

The number of subroutines which can be stored in HP 3245A memory depends on the sizes of the subroutines. A typical subroutine containing 10 commands (including **SUB** and **SUBEND**) might require about 600 bytes. Subroutines can be nested up to 10 deep. With nested subroutines, one subroutine can call (execute) another subroutine. Figure 7-2 summarizes subroutine commands by function (definition/deletion, execution, and conditional statements).

NOTE

Subroutines, variables, and arrays are destroyed from memory when power is removed or when certain commands are executed. Therefore, we recommend you write your subroutines on your controller and then store them on-disc or tape for later use.

Defining/Deleting Subroutines

As shown in Figure 7-2, seven commands (**SUB**, **SUBEND**, **LIST**, **CAT**, **DELSUB**, **SCRATCH**, and **COMPRESS**) are associated with defining/deleting subroutines. A summary description of these commands follows. Refer to the appropriate command in the HP 3245A Command Reference Manual for an example program using the command.

Defining Subroutines (SUB/SUBEND)

Subroutines are defined with an **SUB** *name* command. Executing the **SUB** command instructs the HP 3245A to store all subsequent commands (until a **SUBEND** command is reached) in the specified subroutine.

Defining Subroutine Names

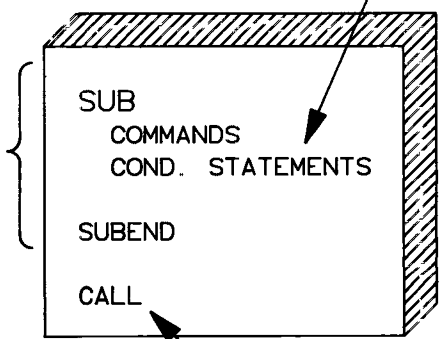
Subroutines names may contain up to 10 characters. The first character must be a letter (A-Z), but the remaining nine characters can be letters, numbers (0-9), the underscore character ("_"), or the question mark ("?"). Subroutine names must not be the same as HP 3245A commands or parameters, previously defined array or variable names, or stored state names.

Use Conditional Statements

```
FOR...NEXT          - loop until loop counter
                      passes specified value.
IF...END IF         - conditional branching
                      within subroutine.
WHILE...END WHILE   - loop as long as numeric
                      expression is TRUE.
```

Define/Delete Subroutine

```
SUB/SUBEND  - define subroutine.
LIST        - list lines of a subroutine.
CAT         - list all subroutine titles.
DELSUB      - delete specified subroutine.
SCRATCH     - delete all subroutines.
COMPRESS    - convert text to machine code.
```



Execute Subroutine

```
CALL       - execute CALL subroutine.
RUN        - execute RUN subroutine.
RUNNING?   - check status of RUN subroutine.
STEP       - single step through subroutine.
PAUSE      - pause a RUN subroutine.

PAUSED?    - check status of RUN subroutine.
CONT       - continue paused/stepped subroutine.
ABORT      - halt execution of RUN subroutine.
RETURN     - return to caller (CALL subroutine)
             or halt execution (RUN subroutine)
```

Figure 7-2. Subroutine Command Summary

When a subroutine is entered, the HP 3245A checks for syntax errors If the syntax is not correct, an error is generated and the command is not stored in the subroutine. You must then edit the subroutine in the controller and re-download the program. The HP 3245A stores subroutines in volatile memory.

If you create or download a subroutine using a subroutine name which already exists in HP 3245A memory, the new subroutine overwrites the previous subroutine. Also, a subroutine will not be stored if a subroutine nesting error exists when the **SUBEND** command is executed (e.g., if one of the called subroutines does not exist in HP 3245A memory).

Subroutine Structure

Each HP 3245A subroutine must contain a **SUB** and **SUBEND** command, and the **SUB** *name* command must be the first line in each subroutine The **SUB** command identifies the subroutine beginning and assigns the name to the subroutine. When the **SUB** command is executed, the HP 3245A begins storing the subroutine in volatile memory.

SUBEND must be the last line in the subroutine. If identifies the subroutine ending and terminates subroutine entry. Commands listed between the **SUB** and **SUBEND** commands are executed, in order, each time the subroutine is executed. Only one **SUB** and one **SUBEND** are allowed in a subroutine. Additional **SUB** or **SUBEND** commands will generate errors. For example, the following lines define subroutine BEEPER which causes the HP 3245A to BEEP once and then return control to the main program. (The subroutine must be called with a **CALL** or **RUN** command before the subroutine will execute)

```
          .
          .
40   OUTPUT 709;"SUB BEEPER"        !Define subroutine BEEPER
50   OUTPUT 709;"  BEEP ONCE"       !Beep once
60   OUTPUT 709;"SUBEND"            !End subroutine
          .
          .
```

Listing Subroutines (CAT/LIST)

When subroutines have been defined and downloaded, you can use the **CAT** command to list the name and size of all subroutines. You can use the **LIST** command to list the program lines of a specified subroutine to the controller CRT.

Listing Subroutine Names (CAT)

The **CAT** (catalog) command lists the names of all HP 3245A subroutines, variables, and arrays presently stored in HP 3245A memory. **CAT** also displays the number of bytes required to store each subroutine and the type and number of elements in each array.

If a subroutine has been defined and then deleted using **DELSUB**, it is listed with a size of zero. When there are no more arrays or subroutines to be listed, the **CAT** command returns the word "DONE". Table 7-6 shows the format for the **CAT** command.

The **CAT** command returns two "size" values for subroutines The first value is the size, in bytes, of the subroutine's source code This is the HP 3245A machine code version of the subroutine. The second value is the size, in bytes, of the subroutine text code which you can list and single-step.

When you compress a subroutine (using the **COMPRESS** command), the TEXT SIZE goes to "0" but the machine code SIZE remains the same. The size value returned for arrays is the actual number of elements in the array.

Table 7-6. CAT Command Data Formats

Subroutines	SUB sub_name SIZE dddd TEXT SIZE dddd
INTEGER Variables	INT variable_name
REAL Variables	REAL variable_name
INTEGER Arrays	IARRAY array_name SIZE dddd
REAL Arrays	RARRAY array_name SIZE dddd

Listing Subroutine Lines (LIST)

The **LIST** command allows you to list the specified subroutine to the controller CRT. The subroutine lines are listed, one at a time, with **SUBEND** as the last line. Note that subroutines which have been compressed (with the **COMPRESS** command) cannot be listed.

Deleting Subroutines (COMPRESS/ DELSUB/ SCRATCH)

Three commands can be used to delete subroutines: **DELSUB**, **SCRATCH**, and **COMPRESS**. (**COMPRESS** does not actually delete a subroutine, but the compressed subroutine cannot be listed or stepped.)

Compressing a Subroutine (COMPRESS)

The **COMPRESS** command removes the text of the specified subroutine from memory. This saves space in memory but eliminates the ability to list (**LIST** command) or single-step (**STEP** command) the subroutine. The **COMPRESS** command should be used only after the subroutine has been debugged and tested. Compressed subroutines continue to appear in the catalog listing (**CAT**) with machine size (SIZE) given, but text size = 0 (TEXT SIZE 0).

Deleting a Subroutine (DELSUB)

The **DELSUB** (delete subroutine) command deletes the specified subroutine from memory but does not delete the subroutine name itself from the catalog listing of subroutines (**CAT** command) Deleting a subroutine recovers the memory space and the subroutine cannot be called (with **CALL**) or run (with **RUN**). The deleted subroutine name still appears in the **CAT** listing (with size 0) and cannot be redefined as anything other than a subroutine. (Use **SCRATCH** to remove definitions of all user-defined names.)

Deleting All User-Defined Names (SCRATCH)

The **SCRATCH** command deletes (scratches) all HP 3245A subroutines, variables, and arrays from volatile memory. It also deletes all name definitions from the catalog listing (**CAT** command). If **SCRATCH** is executed when a subroutine is running, an error is generated but the subroutine is not purged from memory.

Executing Subroutines

As shown in Figure 7-2, nine commands (**CALL**, **RUN**, **RUNNING?**, **STEP**, **PAUSE**, **PAUSED?**, **CONT**, **ABORT**, and **RETURN**) are associated with defining/deleting subroutines. A summary description of these commands follows. Refer to the appropriate command in the HP 3245A Command Reference Manual for an example program using the command

Calling Subroutines (CALL/RUN/ RUNNING?)

There are three commands associated with calling subroutines: **CALL**, **RUN**, and **RUNNING?**.

Calling a Subroutine (CALL)

The **CALL** *sub_name* command executes the named subroutine and waits for the subroutine to complete before executing other commands. This means that no further commands are accepted (either from HP-IB or the front panel keyboard) until the subroutine finishes. The specified subroutine must be stored in HP 3245A memory before executing **CALL**.

CALL may also be used in a subroutine to call another subroutine to provide expanded capability of "nested" subroutines When using nested subroutines, the calling subroutine is suspended so that only one subroutine is running at a time. Subroutines can be nested up to 10 deep.

If **USE** is executed inside a subroutine called by **CALL**, the HP 3245A retains that assignment after the subroutine completes. For example, if channel A is selected as the use channel within a subroutine called by **CALL**, channel A remains the use channel after the subroutine completes. For example, the following program lines call subroutine BEEPER.

```
        .
        .
40   OUTPUT 709;"SUB BEEPER"        !Define subroutine BEEPER
50   OUTPUT 709;"  BEEP ONCE"       !Beep once
60   OUTPUT 709;"SUBEND"            !End subroutine
70   OUTPUT 709;"CALL BEEPER"       !Call subroutine BEEPER
        .
        .
```

A subroutine may call a second (nested) subroutine for execution before the first subroutine finishes execution. When the second subroutine executes the **SUBEND** command, the first subroutine continues with the next command following the embedded **CALL** command. (Do not use the **RUN** command to nest subroutines.)

There are two requirements for nesting subroutines. First, the subroutine called from another subroutine must be stored in memory before the subroutine doing the calling is stored. Second, subroutines may not be nested more than 10 levels deep. You cannot place a subroutine inside another subroutine.

Running a Subroutine (RUN)

The **RUN** *sub_name* command executes the named subroutine in parallel with other commands so the HP 3245A executes the subroutine as it finds time between executing other commands. Commands executed from the keyboard or from the controller temporarily interrupt execution of the subroutine. Do not use the **RUN** command to nest subroutines.

If **USE** is executed within a subroutine executed with **RUN**, the HP 3245A retains that assignment ONLY during the subroutine. After the subroutine completes, the assignment reverts back to the channel in use before the subroutine was executed. For example, the following program lines execute the **RUN** subroutine BEEPER. Note that this subroutine will run only when the HP 3245A is not busy executing other commands.

```
        .
        .
40   OUTPUT 709;"SUB BEEPER"        !Define subroutine BEEPER
50   OUTPUT 709;"  BEEP ONCE"       !Beep once
60   OUTPUT 709;"SUBEND"            !End subroutine
70   OUTPUT 709;"RUN BEEPER"        !Runl subroutine BEEPER
        .
        .
```

Query Subroutine Running Status (RUNNING?)

The **RUNNING?** query command returns a "1" if a subroutine called by **RUN** is currently running or paused (see **PAUSE**) or returns a "0" if the subroutine is not running. (Only subroutines executed with the **RUN** command can be paused.)

Pausing Subroutines (STEP/PAUSE/PAUSED?/CONT)

Four commands are associated with pausing subroutines: **STEP**, **PAUSE**, **PAUSED**, and **CONT**. **PAUSE**, **PAUSED?**, and **CONT** apply only to run subroutines called by **RUN** .

Pausing a Subroutine (PAUSE)

The **PAUSE** command pauses a subroutine executed with the **RUN** command. Once a subroutine has been paused, you can use the **STEP** (single-step) or **CONT** (continue) command to resume execution. The **CONT** command allows the subroutine to continue running to completion, starting with the command following the **PAUSE** command. **PAUSE** cannot be used with a subroutine called by the **CALL** command. An error is generated if you attempt to execute **CONT** or **STEP** when a subroutine is not paused or if the specified subroutine does not exist.

Subroutine Pause Status (PAUSED?)

The **PAUSED?** command returns a "1" if the current **RUN** subroutine is paused or returns a "0" if the subroutine is running (or finished running). Again, only subroutines executed with a **RUN** command can be paused.

Single Step Through Subroutine (STEP)

The **STEP** [*sub_name*] command allows you to step through the specified subroutine, line by line, to verify its operation. To single-step through a downloaded subroutine, send **STEP** [*sub_name*] repeatedly. Each time **STEP** is executed, the next subroutine command is displayed and executed. If the command generates data, the data is displayed.

For example, if TEST1 is a defined subroutine, each time `OUTPUT 709;"STEP TEST1"` is executed, the next line of the subroutine is displayed. You can use the **CONT** command to cease stepping and continue normal execution of the subroutine from the last line stepped. Note that compressed subroutines (**COMPRESS** command) cannot be stepped.

Continuing a Subroutine (CONT)

CONT resumes execution of a **RUN** subroutine that is currently paused (**PAUSE** command) or is in the single-step mode (**STEP** command). (Only subroutines executed with the **RUN** command can be paused or stepped.) When executed, **CONT** allows the subroutine to execute to completion, starting with the command after **PAUSE** or after the last command to be stepped.

Exiting Subroutines (ABORT/ RETURN)

Two commands are associated with exiting subroutines: **ABORT** and **RETURN**.

Aborting a Subroutine (ABORT)

The **ABORT** command halts execution of any subroutine executed with **RUN** and returns control to the controller or to the front panel keyboard **ABORT** can only halt an executing (**RUN**) subroutine following execution of the current subroutine command. **ABORT** will also halt any nested subroutines, but will do nothing if a subroutine is not running at the time **ABORT** is executed.

ABORT cannot be used for subroutines executed with **CALL**. To abort a called subroutine, press the front panel **Clear** key or send a **CLR** command. Also, do not use **ABORT** inside a subroutine.

Exiting a Subroutine (RETURN)

Unless interrupted, a subroutine called by **CALL** will continue execution until it reaches the **SUBEND** command However, the **RETURN** command can also be used to end the **CALL** subroutine. **RETURN** must be used in a subroutine and is generally useful only when used in an **IF..END IF** loop

When **RETURN** is executed within a **CALL** subroutine, program execution is returned to the program line following the **CALL** command. For subroutines executed using the **RUN** command, **RETURN** terminates execution of the subroutine.

Using Conditional Statements

The HP 3245A provides three BASIC language statements for conditional branching and looping: **FOR..NEXT**, **WHILE..END WHILE**, and **IF..END IF**, which are used ONLY within HP 3245A subroutines. These statements are similar to those in an enhanced BASIC language, except HP 3245A subroutines do not have line numbers or GOTO statements for branching. Looping and conditional branching statements may be nested seven deep.

FOR..NEXT Loops

The **FOR..NEXT** command defines a loop which is repeated until a loop counter passes a specified value. The syntax for the **FOR..NEXT** loop follows. Note that the **FOR..NEXT** loop is valid only in a subroutine.

FOR *counter* = *initial_value* **TO** *final_value* **[STEP** *step_size***]**

program segment

NEXT *counter*

The *counter* parameter is a variable name which acts as the loop counter. The *initial_value* parameter and *final_value* parameter may be numbers, numeric variables, or numeric expressions. The [*step_size*] parameter may be a number or a numeric expression which specifies the amount the loop counter is incremented for each pass through the loop.

A negative value for *step_size* decrements the loop counter. The program segment is repeatedly executed until the loop counter exceeds the *final_value*.

Example 7-1: FOR..NEXT Loop (FRNXT7)

This program decrements the loop counter from 20 to 1 in increments of -1 (20, 19, ..., 1) and displays the results on the front panel display.

```
10   !file FRNXT7
20   !
30   CLEAR 709                              !Clear HP 3245A
40   OUTPUT 709;"RST"                       !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER I"                 !Define INTEGER variable I
70   OUTPUT 709;"SUB LOOP"                  !Begin subroutine
80   OUTPUT 709;"  FOR I=20 TO 1 STEP -1"   !Begin loop
90   OUTPUT 709;"    DISP I"                !Display value
100  OUTPUT 709;"    WAIT 0.5"              !Wait 0.5 sec
110  OUTPUT 709;"  NEXT I"                  !Next value
120  OUTPUT 709;"SUBEND"                    !End subroutine
130  OUTPUT 709;"CALL LOOP"                 !Call subroutine
140  END
```

WHILE..END WHILE Loops

The **WHILE..END WHILE** construct defines a loop which is repeated as long as the specified numeric expression is true. The syntax for the **WHILE..END WHILE** construct follows. Note that **WHILE..END WHILE** is valid only within a subroutine.

> **WHILE** *expression*
>
> program segment
>
> **END WHILE**

The **WHILE..END WHILE** operation depends on the result of a test performed at the start of the loop. If the test is TRUE (not equal to zero), the program segment between the **WHILE** and **END WHILE** statements is executed and a branch is made back to the **WHILE** statement. If the test is FALSE (equal to zero), program execution continues with the statement following the **END WHILE** statement.

Example 7-2: WHILE..END WHILE Loop (WHEND7)

This program uses subroutine PWRTWO with a **WHILE..END WHILE** loop to display the powers of 2 less than 1000 (1, 2, . . ., 512) on the front panel display.

```
10   !file WHEND7
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER I"         !Define INTEGER variable I
70   OUTPUT 709;"SUB PWRTWO"        !Define subroutine PWRTWO
80   OUTPUT 709;"  I=1"             !Set I = 1
90   OUTPUT 709;"  WHILE I <= 1000" !Continue until 2*I >= 1000
100  OUTPUT 709;"    DISP I"        !Display value of I
110  OUTPUT 709;"    I=(2*I)"       !Assign next value to I
120  OUTPUT 709;"    WAIT 1"        !Wait 1 second
130  OUTPUT 709;"  END WHILE"       !End when 2*I >= 1000
140  OUTPUT 709;"  DISP 'Done'"     !Display message when done
150  OUTPUT 709;"SUBEND"            !End subroutine
160  OUTPUT 709;"CALL PWRTWO"       !Call subroutine PWRTWO
170  END
```

IF..END IF Branching

The **IF..END IF** construct provides conditional branching within HP 3245A subroutines. The syntax for the **IF..END IF** construct follows. Note that **IF..END IF** is valid only within an HP 3245A subroutine.

IF *expression* **THEN**

program segment

[ELSE]

[program segment]

END IF

The **END IF** statement must follow the **IF..THEN** statement somewhere in the subroutine. **ELSE** is an optional statement, but if used must appear before the **END IF** statement. All commands after the **IF..THEN** statement and before the **ELSE** and **END IF** statements will be executed if the expression evaluates to TRUE (not equal to zero).

If the expression is TRUE, execution continues with the program segment between **IF..THEN** and **ELSE**. If the expression is FALSE, execution continues with the segment after **ELSE**. In either case, when the program segment is completed (assuming there are no other loops or conditional branches) program execution continues with the statement following the **END IF** statement.

Example 7-3: IF..END IF Branching (IFEND7)

This program uses **IF..END IF** within subroutine IFSQR to display the square root of a number or to indicate if the number is less than 0. The first time the subroutine is called, the value 2.0000000E+00 appears on the front panel display. The second time the subroutine is called, the number is less than 0, so the HP 3245A BEEPs once and "Number <0" appears on the front panel display.

```
10   !file IFEND7
20   !
30   CLEAR 709                          !Clear HP 3245A
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60   OUTPUT 709;"REAL R"                !Define REAL variable
70   OUTPUT 709;"SUB IFSQR"             !Define subroutine IFSQR
80   OUTPUT 709;"  IF R<0 THEN"         !Start IF..END IF loop
90   OUTPUT 709;"    BEEP"              !BEEP once
100  OUTPUT 709;"    DISP 'Number <0'"  !Display "Number <0" if R<0
110  OUTPUT 709;"  ELSE"                !Do 120-130 only if R>=0
120  OUTPUT 709;"    DISP SQR(R)"       !Display square root of R
130  OUTPUT 709;"  END IF"              !End IF..END IF loop
140  OUTPUT 709;"SUBEND"                !End subroutine
150  WAIT 1                             !Wait 1 second
160  OUTPUT 709;"LET R=4"               !Set R = 4
```

```
170  OUTPUT 709;"CALL IFSQR"            !Call subroutine IFSQR
180  WAIT 1                             !Wait 1 second
190  OUTPUT 709;"LET R=-1"              !Set R = -1
200  OUTPUT 709;"CALL IFSQR"            !Call subroutine IFSQR
210  END
```

Contents

Chapter 8
Input/Output Operations

Chapter 8
Input/Output Operations

Chapter Contents

This chapter shows how to use input/output operations for the HP 3245A. The chapter contents are:

- **HP-IB Communication** shows how to use echo checks; how to reset, clear, and test the HP 3245A; how to set the HP 3245A HP-IB address; how to set the use channel; and how to set beeper mode.

- **Setting Local/Remote Operation** shows how to set local (front panel) and remote (controller) operation. It also shows how to lock out front panel operation.

- **Input/Output Buffering** shows how to select input and output buffering mode; how to set the output format, how to clear the output buffer; how to query the "ready" status of the HP 3245A; and how to set EOI termination.

- **Interrupts and Service Requests** describes the status register and RQS mask; shows how to read the status word and byte; shows how to generate interrupts to the controller; and shows how to enable programmed and power-on service requests.

HP-IB Communication

This section summarizes HP-IB communication for the HP 3245A. It includes the following topics:

- Using echo check
- Resetting/clearing/testing the HP 3245A.
- Setting/reading HP-IB address.
- Setting/reading use channel
- Enabling beeper mode.

Using Echo Check (ECHO)

One way to check communication between your controller and the HP 3245A is to use the **ECHO** *string* command, where *string* is any set of printable ASCII characters enclosed in either double quotation marks (""*string*"") or single apostrophes ('*string*'). When the **ECHO** message is sent, the HP 3245A returns the message to the controller for display.

NOTE

If your controller does not use cr lf, refer to the **END** *command in the HP 3245A Command Reference Manual for details on asserting EOI*

To use the **ECHO** command, first check the HP 3245A HP-IB address using the **Address** command on the HP 3245A front panel. (The factory set address is 09.) Then, send a desired message from your controller using the HP-IB address to make the echo check. Refer to the **ECHO** command in the HP 3245A Command Reference Manual for an example program.

Reset/Clear/Test HP 3245A (RST/CLR/TEST)

When communication has been established between your controller and the HP 3245A, you can use the **RST** and/or **CLR** commands to set the HP 3245A to a known state. As desired, you can also perform a confidence self-test of the instrument with the **TEST** command.

Resetting the HP 3245A (RST)

The **RESET** (or **RST**) command resets the HP 3245A to a known (power-on) state. **RESET** [*ch*] without the *ch* parameter resets the entire HP 3245A, while **RESET** *ch* resets the specified channel only.

Use **RESET 0** or **RESET CHANA** to reset channel A, use **RESET 100** or **RESET CHANB** to reset channel B, or use **RESET** to reset the HP 3245A. Refer to the **RESET** command in the HP 3245A Command Reference Manual for a list of command actions and the power-on state of the HP 3245A.

Clearing the HP 3245A (CLR)

The **CLR** command places the HP 3245A in the same state as set by the HP-IB **CLEAR** (device clear or selected device clear) command. Refer to the HP 3245A **CLR** command and/or to the HP-IB **CLEAR** command in the HP 3245A Command Reference Manual for a list of actions performed by **CLR** or **CLEAR**.

Confidence Self-Test (TEST)

The **TEST** (or **TEST**) command performs a confidence test on the HP 3245A or on a specified channel. **TEST** does not change the hardware or software state, but does set the output to 0 V on the specified channel(s). Use **TEST 0** or **TEST CHANA** to perform a confidence test on channel A, use **TEST 100** or **TEST CHANB** to perform the test on channel B, or use **TEST** to perform the test on the instrument.

TEST returns "PASS" if all tests pass or "FAIL" if one or more tests fail. Any failures which occur during the test are displayed on the front panel and the first four errors are stored in the error register. (Use the **ERR?** or **ERRSTR?** to read the errors.) An example program using **TEST** follows.

Example 8-1: Testing the HP 3245A (TEST8)

This program tests the HP 3245A, channels A and B, and the front panel. If all tests pass, "Test Passed" appears on the controller CRT. If one or more tests fail, "Test Failed" appears on the controller CRT and the first four errors are displayed.

```
10   !file TEST8
20   !
30   DIM Err$[60]                  !Dimension controller array
40   OUTPUT 709;"TEST"             !Test HP 3245A and front panel
50   ENTER 709;A$                  !Enter test result (pass/fail)
60   IF A$ = "FAIL" THEN           !Enter loop
70     PRINT "Test Failed"         !Display message if test fails
80     FOR I = 1 TO 4              !Error loop
90       OUTPUT 709;"ERRSTR?"      !Read error string
100      ENTER 709;Err$            !Enter string
110      PRINT Err$                !Display  string
120    NEXT I                      !Loop until error register is empty
130  ELSE
140    PRINT "Test Passed"         !Display message if test passes
150  END IF                        !End loop
160  END
```

Setting HP-IB Address (ADDRESS)

To communicate with the HP 3245A, you will need to know the HP-IB address of the instrument. You can read the HP-IB address with the **ADDRESS?** (or **ADDR?**) command and set the HP-IB address (if required) with the **ADDRESS** command. The HP 3245A is factory-set for address 09.

If required, you can change the HP 3245A HP-IB address (in the range from 0 to 30) by sending **ADDRESS** *address* over HP-IB. You can also display the instrument address using the front panel **Address** key. An example program to set the HP-IB address follows.

Example 8-2: Setting HP-IB Address (ADDR8)

This program changes the HP-IB address of an HP 3245A from "17" to "09" and returns the new address to the controller CRT.

```
10   !file ADDR8
20   !
30   CLEAR 717                     !Clear HP 3245A
40   OUTPUT 717;"RST"              !Reset HP 3245A
50   OUTPUT 717;"SCRATCH"          !Clear HP 3245A memory
60   OUTPUT 717;"ADDRESS 9"        !Set new address
70   OUTPUT 709;"ADDRESS?"         !Read current HP-IB address
80   ENTER 709;A                   !Enter address
90   PRINT "Address =";A           !Display address
100  END
```

A typical return is:

```
Address = 9
```

Setting Use Channel (USE)

The HP 3245A has two versions: the standard version which has one channel (called channel A) and Option 001 which has two channels (channel A and channel B). Some HP 3245A commands require that a use channel be specified before the command can be executed.

You can specify the use channel (the channel to receive subsequent commands) by sending the **USE** *ch* command, where **USE 0** (or **USE CHANA**) sets channel A and **USE 100** (or **USE CHANB**) sets channel B as the use channel. At power-on or following a reset, channel A is the use channel.

When a use channel is set, all commands following the **USE** *ch* command are sent to the use channel. The use channel designation remains in effect until another **USE** *ch* command is sent, the HP 3245A is reset, or power is cycled.

You can read the current use channel (channel A or B) by sending the **USE?** command. If channel A is the use channel, "0" is returned. If channel B is the use channel, "100" is returned. Refer to the **USE** command in the HP 3245A Command Reference Manual for an example program to set the use channel.

Enabling Beep Mode (BEEP)

The **BEEP** [*mode*] command enables or disables the HP 3245A beeper mode. When the beeper mode is enabled, a beep occurs when an error is generated. When beeper mode is disabled, error messages are displayed on the front panel but the beep is suppressed. The **BEEP** *mode* parameters are OFF, ON, and ONCE. OFF disables beeper mode and ON/ONCE enable beeper mode. Power-on *mode* = ON and default *mode* = ONCE.

BEEP ONCE overrides the present ON/OFF mode of **BEEP** and produces a single beep. The beep occurs even if the beeper mode is disabled (**BEEP OFF**). After a single beep, the beeper mode returns to the previous mode (ON or OFF). The beeper mode is stored in continuous memory and is retained when power is removed from the HP 3245A. The HP 3245A signals the end of its power-on test with a tone. The **BEEP** command does not affect this tone.

Setting Local/Remote Operation

This section shows how to set the HP 3245A for local or remote operation. When the HP 3245A is in the local state, the user has access to all front panel commands and functions available. In remote state, the HP 3245A is under remote (HP-IB) control and only a subset of the front panel commands can be used.

In remote state, the state of the HP 3245A cannot be changed, but query and monitor commands can be used. Figure 8-1 shows the four local/remote states for the HP 3245A and shows the HP 3245A and/or HP-IB commands which are required to go from one state to another.

Setting Local Operation (LOCAL)

In Local State (LOCS) and Local with Lockout State (LWLS), the front panel REM annunciator is OFF and the HP 3245A accepts commands from both the front panel keyboard and from the controller. The HP 3245A is set to Local State at power-on.

NOTE

Executing **LOCK ON** *totally disables the front panel keyboard. With* **LOCK ON**, *commands cannot be entered from the front panel regardless of the state (Local/Remote) of the HP 3245A. Send* **LOCK OFF** *followed by* **LOCAL** *to restore front panel operation.*

Local State (LOCS) to Local with Lockout State (LWLS)

To go from Local State to Local with Lockout State (LWLS), execute the HP-IB **LOCAL LOCKOUT** command. In LWLS, the HP 3245A front panel **Local** key cannot be used to restore local control. However, when LWLS is entered from LOCS, the user can still use the front panel keyboard. If LOCS was entered from LWLS or RWLS, **REMOTE 7** is required prior to **LOCAL LOCKOUT 7** in order to return to LWLS

Remote State (REMS) to Local State (LOCS)

To return to Local State from REMS, use OUTPUT 709;"LOCAL", press the front panel **Local** key, or use **LOCAL 7** or **LOCAL 709**. Note that the **Local** key performs the same action as the **LOCAL** command, except that the **Local** key does not clear the **RMT** (local lockout) command if enabled.

Remote with Lockout State (RWLS) to Local State (LOCS)

To return to Local State (LOCS) from RWLS, execute a **LOCAL 7** command.

Remote w/Lockout State (RWLS) to Local w/Lockout State (LWLS)

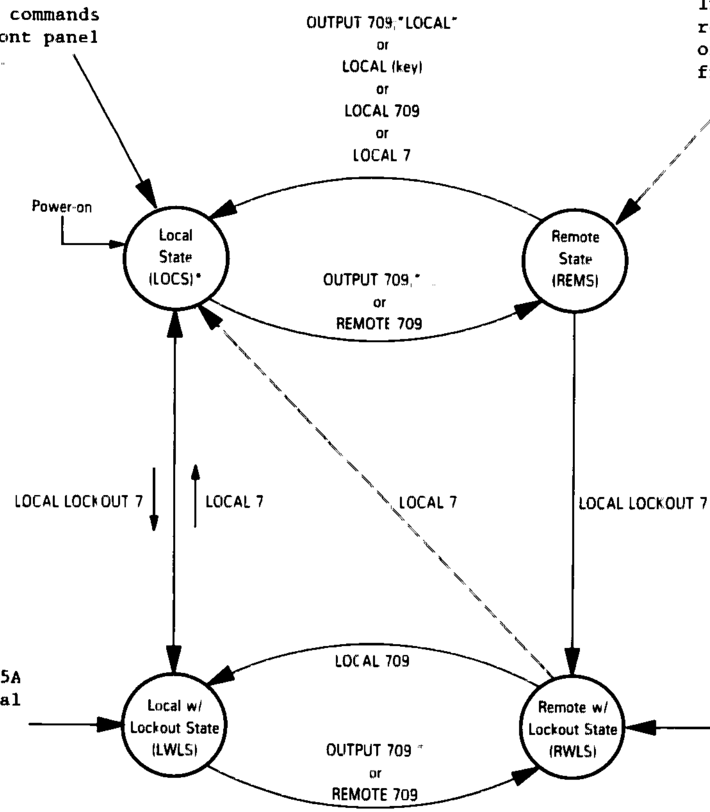
To return from RWLS to LWLS, execute a **LOCAL 709** command.

Local State (LOCS)

In Local State, the user has local control of the HP 3245A and access to all commands and functions allowed from the front panel. Local State is the power-on state.

Remote State (REMS)

In Remote State, the HP 3245A is under remote (HP-IB) control and only a subset of the front panel commands can be entered from the front panel keyboard.



Local with Lockout State (LWLS)

In Local with Lockout State, the HP 3245A Local key cannot be used to restore local control. However, when LWLS is entered from Local State (LOCS), the user has control of the HP 3245A until the instrument is placed in the Remote with Lockout State (RWLS).

Remote with Lockout State (RWLS)

In Remote with Lockout State, the HP 3245A Local key cannot be used to restore local control (as with LWLS). However, in RWLS the HP 3245A is under remote (HP-IB) control.

*REMOTE 7 followed by an OUTPUT statement or REMOTE 709 is required to go to REMS if you entered LOCS from LWLS or RWLS

REMOTE 7 is required prior to LOCAL LOCKOUT 7 in order to return to LWLS if you entered LOCS from LWLS or RWLS

Figure 8-1. Remote/Local States

Setting Remote Operation (REM)

In Remote State (REMS) or in Remote with Lockout State (RWLS), the front panel REM annunciator is ON and the HP 3245A executes commands from the controller. The HP 3245A can also execute commands from the front panel keyboard as long as the command does not change the hardware state of the instrument.

Since most controllers set the HP-IB REN (remote enable) line true at power-on or after a reset operation, executing any command (except **LOCAL**) from the controller will also place the HP 3245A in Remote State. Table 8-1 lists front panel commands which are allowed when the HP 3245A is in Remote State (REMS) or in Remote with Lockout State (RWLS).

Table 8-1. Front Panel Commands Allowed in Remote

ADDRESS?	DRIVETBn?	OUTPUT?	SRQ
APPLY?	DUTY?	PANG?	STA?
ARANGE?	ERRSTR?	PAUSED?	STB?
BEEP	FREQ?	RANGE?	SYNCOUT?
CALEN?	HELP	READY?	TBn?
CALL	ID?	REFIN?	TERM?
CALNUM?	IDN?	REFOUT?	TIME
CALSTR?	IMP?	REV?	TRIGIN?
CAT	LIST	RQS?	TRIGMODE?
CTYPE?	LOCAL	RUNNING?	TRIGOUT?
DCOFF?	MEMAVAIL?	SER?	USE?
DCRES?	MON	SIZE?	WAIT
DELAY?			

Local State (LOCS) to Remote State (REMS)

To go from Local State to Remote State (REMS), execute an HP-IB **REMOTE 709** command or send an OUTPUT 709;"... command. In Remote State, the state of the HP 3245A cannot be changed, but you can query the instrument or monitor the output. (If LOCS is entered from LWLS or RWLS, **REMOTE 7** followed by an OUTPUT 709 statement or **REMOTE 709** is required to go to REMS.)

Remote State (REMS) to Remote with Lockout State (RWLS)

To change from Remote State (REMS) to Remote with Lockout State (RWLS), send HP-IB command **LOCAL LOCKOUT 7**. In RWLS, the HP 3245A **Local** key cannot be used to restore local control and the instrument is under remote (HP-IB) control.

Input/Output Buffering

This section describes input and output buffering for the HP 3245A, including:

- Input buffering.
- Output buffering.
- Data storage/reads.
- Output data formats.

Figure 8-2 summarizes input/output operation for the HP 3245A. Commands issued from the controller with an OUTPUT 709;"... command are sent to the input buffer. **INBUF** enables or disables the buffer. If a data-generating (query) command is sent, the resulting data is sent either to the output buffer or to HP 3245A memory, depending on the **MEM** command.

Data stored in the output buffer is transferred to the controller by an ENTER 709;A type statement if the data-generating command was sent by the controller or to the front panel display if the command was entered from the front panel keyboard. Data stored in memory must first be transferred to the output buffer with a **VREAD** command and then output to the controller or front panel.

Data is sent to the controller in ASCII or binary format, depending on the **OFORMAT** command. For binary data, **BLOCKOUT** specifies whether or not a data header precedes the data. (Only ASCII data can be displayed on the front panel display.)

Input Buffering (INBUF/READY?)

For HP 3245A input buffering, **INBUF** *mode* enables or disables the HP 3245A input buffer. With **INBUF OFF**, the input buffer is disabled and commands are accepted only when the HP 3245A is not busy. With **INBUF ON**, the input buffer is enabled and commands are stored in the buffer, thus releasing the HP-IB interface immediately. With **INBUF ON**, you can use the **READY?** command to determine when the HP 3245A is ready to accept new commands.

Input Buffering Disabled (INBUF OFF)

When input buffering is disabled (**INBUF OFF**), when the HP 3245A is addressed to listen (accept commands from the controller), commands are accepted from the HP-IB interface one character at a time, as the HP 3245A is ready for them. Thus, the interface bus is held during the time the HP 3245A is processing a command.

While the HP 3245A is processing a character, it holds the HP-IB interface. After processing the character, the HP 3245A completes the handshake and accepts the next character. When the HP 3245A receives a valid terminator, it executes the entire command.

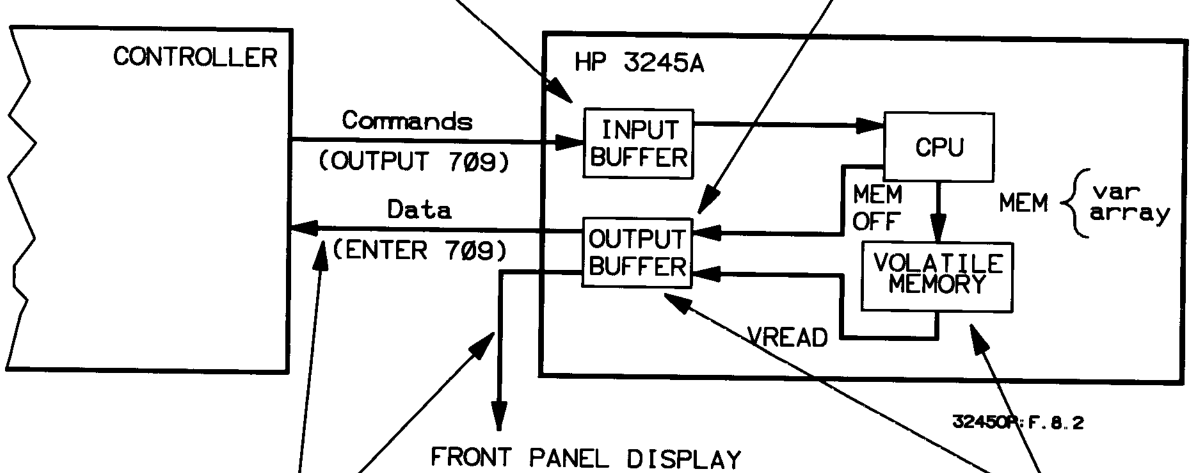
Valid command terminators include semicolons (;), carriage returns (*cr*), line feeds (*lf*), and EOI sent concurrently with the last character send. Most controllers send *cr* and *lf* after each OUTPUT statement. The first terminator (*cr*) initiates command execution. The final terminator (*lf*) is held off until command execution is complete. When the command is complete, the HP 3245A accepts the *lf* and frees the HP-IB interface.

Input Buffering (INBUF/READY?)

INBUF enables/disables the input buffer. With INBUF ON, commands are stored in buffer, releasing the bus immediately. With INBUF OFF, commands are accepted when the HP 3245A is not busy. With INBUF ON, use READY? to determine when the HP 3245A is ready to accept new commands.

Output Buffering (OUTBUF/CLROUT)

OUTBUF enables/disables the output buffer. With OUTBUF ON, new data written into the buffer is appended to existing data in the buffer. With OUTBUF OFF, new data written into the buffer overwrites existing data in the buffer. Use CLROUT (or CLR, RST, or HP-IB Device Clear) to clear the output buffer (erase data).



Output Data Formats (END/OFORMAT/BLOCKOUT)

Data stored in the output buffer is transferred to the controller or to the front panel display, depending on where the data-generating command originated. As required, use END ON or END OFF to assert or suppress the HP-IB EOI signal

Data is transferred to the controller in ASCII or binary format. OFORMAT ASCII specifies ASCII format, OFORMAT BINARY specifies binary format For binary data, BLOCKOUT specifies whether a data header precedes the data.

Data Storage (MEM/VREAD)

With MEM not set, data from query commands is stored directly in the output buffer. With MEM set, data from query commands is stored in the variable or array specified by MEM. Use VREAD to transfer data from the variable/array to the output buffer.

Figure 8-2. Input/Output Operation

The HP 3245A executes HP-IB bus commands, such as Group Execute Trigger (GET) or Device Clear (DCL), in sequence with other commands sent over the HP-IB interface. Thus, it is normally not necessary to add programmed delays in the controller to allow more time for commands to execute.

Input Buffering Enabled (INBUF ON)

When input buffering is enabled (**INBUF ON**) and the HP 3245A is addressed to accept commands from the controller, the HP 3245A stores incoming command characters in a 256-character first-in-first-out (FIFO) circular buffer. This technique minimizes bus transfer time and allows your controller to communicate with other instruments while the HP 3245A is processing commands. If the buffer becomes full, the HP 3245A holds off HP-IB bus communication until it can process commands and open up space in the buffer.

Determining HP 3245A Ready Status (READY?)

With **INBUF ON**, synchronization between the HP 3245A and the controller is lost. However, you can use the **READY?** command to monitor the ready bit in the HP 3245A status register and thus determine when the contents of the input buffer have been executed. (**READY?** is useful only when the input buffer is enabled with **INBUF ON**).

The **READY?** command returns a "1" when the instrument is ready to accept additional commands. If the HP 3245A is presently executing a command or subroutine, the instrument waits until the command or subroutine is complete before returning a "1" to indicate a ready state. The HP 3245A remains "busy" until **READY?** completes. Refer to the **READY?** command in the HP 3245A Command Reference Manual for an example program using the **READY?** command.

Output Buffering (OUTBUF/ CLROUT/END)

Data generated by the HP 3245A is sent to the output buffer (or to memory and then to the output buffer if **MEM** is used). The **OUTBUF** command enables or disables the output buffer. Use **CLROUT** to clear the output buffer (erase data in the buffer). For some controllers, EOI termination (using the **END** command) may be required to output data to the controller.

Disabling the Output Buffer (OUTBUF OFF)

When a data-generating command is entered from the HP 3245A front panel keyboard, the response is sent to the front panel display. If the command is executed from the controller, the response is sent to the output buffer or to memory. The **OUTBUF** command enables or disables the HP 3245A output buffer. The output buffer capacity is 2048 bytes of data. When the output buffer is disabled (**OUTBUF OFF**), any new command which generates data first clears the buffer and then writes the new data into the buffer. This is the power-on state for the output buffer.

NOTE

Depending on the controller, if **END ON** *is not used you may need to specify an image statement to ignore terminators between data strings when entering data into the controller, since cr lf is sent after each data string.*

Enabling the Output Buffer (OUTBUF ON)

When the output buffer is enabled (**OUTBUF ON**), new data is appended to the previous data stored in the output buffer. In this mode, the HP 3245A sends data to the output buffer in groups but does not overwrite data in the buffer. With **OUTBUF ON**, when the buffer fills (2048 bytes), the HP 3245A holds the HP-IB interface and does not complete the present command until there is sufficient space in the buffer for all output data from the command.

If the controller addresses the HP 3245A to send data over HP-IB but the output buffer is empty, all further HP-IB activity stops until data becomes available. If no command or subroutine has been executed which will eventually generate data, the HP-IB bus remains "busy" until the controller times out.

If the controller requests more data than the output buffer can hold (or is currently in the output buffer), the HP 3245A will hold off HP-IB communications until the data becomes available and is sent to the controller. (The HP 3245A will not accept any new commands during this time.)

If the output buffer is full when the HP 3245A generates new output data, the HP 3245A processor releases the input buffer (if it was held) and waits for the controller to read the output buffer. You must read the output buffer or execute an HP-IB Device Clear before the HP 3245A will process new commands. The HP 3245A processor sets the status register DATA AVAILABLE bit (bit 0) when output data is available in the output buffer. When the output buffer is empty, this bit is cleared. Refer to "Interrupts and Service Requests" for details on SRQ.

Clearing the Output Buffer (CLROUT)

The **CLROUT** command clears (erases) the data in the output buffer. When the output buffer is enabled (**OUTBUF ON**), **CLROUT** is useful to ensure that the next query (data-generating) command gives a predictable response. With **OUTBUF OFF** (output buffer disabled), it is usually not necessary to clear the buffer, since new data overwrites the existing data in the buffer.

NOTE

With **INBUF ON** *(input buffer enabled), commands following* **CLROUT** *may be accepted and stored in the buffer before* **CLROUT** *is executed. If these commands generate data, the data returned may be old data, new data, or a combination of old and new data.*

EOI Termination (END)

Normally, data messages are sent over HP-IB using standard ASCII codes which are terminated by a *cr lf*. However, since some controllers cannot terminate data input on *cr lf*, the *End or Identify* (EOI) line in the HP-IB interface may be used to mark the end of the data message. The **END** command enables or disables the EOI function. With the EOI function enabled (**END ON**), the HP 3245A sets the EOI line TRUE concurrent with the last character of the message to terminate the data transfer. (The *lf* is still sent even with **END ON**.) At power-on, the EOI function is disabled (**END OFF**) so the EOI line is always suppressed.

With **OUTBUF ON**, data sent to the output buffer overwrites data currently in the buffer. Thus, asserting EOI is effective only for controllers which do not recognize *lf* as a terminator. With **OUTBUF ON**, data from multiple commands is appended to existing data in the buffer. With **END ON**, EOI is asserted with the last byte (*lf*) of the data for each command, which requires multiple ENTER statements. With **END OFF**, EOI is suppressed and a single ENTER statement can be used.

Data Storage/ Reads (MEM/ VREAD)

When an HP 3245A command generates data, the data is either sent directly to the output buffer or is stored in an array or variable in HP 3245A memory, as specified by the **MEM** command. Data stored in a variable or array must be transferred to the output buffer with a **VREAD** command before it can be sent to the controller.

Using Memory Mode (MEM)

The **MEM** command specifies a variable or array to store data from commands which return numeric results. The HP 3245A cannot store ASCII (string) results. The syntax for the **MEM** command is:

```
     OFF
MEM  variable
     array_name [(start_index)]
```

With **MEM OFF** (power-on), results are sent to the output buffer and are not stored in memory. When **MEM** *variable* is used, the next numeric command result is stored in the specified variable and the memory mode is then automatically turned off (**MEM OFF**). Memory mode is also disabled by the **CLR** command, with the front panel **Clear** key, or by an HP-IB Device Clear message.

When **MEM** *array_name* is specified, all subsequent numeric data is stored in the specified array. The *start_index* parameter is used with the *array_name* parameter to specify a starting location of the array (store numeric results at a specific array element). The range is any integer from 0 to 32767. Default *start_index* = 0 (begin storing data at the first [0th] element of the array).

When the memory storage mode is ON, numeric output data is sent once to the specified variable or continuously to the specified array. If data is to be stored in a variable or array, the variable must be defined using **REAL** or or **INTEGER** or the array must be dimensioned using **DIM**, **REAL**, or **INTEGER**. An example to store data in a variable follows.

Example 8-3: Storing Data in Memory (MEMS8)

This program uses **MEM** to store the current HP-IB address of the HP 3245A in variable Address and uses **VREAD** to transfer the value to the output buffer. (**MEM OFF** is used in this program but is optional since the memory mode is automatically disabled after storing variable values.)

```
10   !file MEMS8
20   !
30   CLEAR 709                          !Clear HP 3245A
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER Adrs"          !Define variable Adrs
70   OUTPUT 709;"MEM Adrs"              !Store results in Adrs
80   OUTPUT 709;"ADDRESS?"              !Read current HP-IB address
90   OUTPUT 709;"MEM OFF"               !Turn memory mode OFF
100  OUTPUT 709;"VREAD Adrs"            !Trans value in Adrs to out buffer
110  ENTER 709;A                        !Enter address
120  PRINT "Address =";A                !Display address
130  END
```

A typical return for HP-IB address 709 follows. If memory mode is not set (line 70 is deleted), the program returns "0", since **VREAD** reads the value in Adrs (set to 0 by **SCRATCH**) and overwrites the previous value (09) placed in the output buffer by **ADDRESS?**.

```
Address = 9
```

Read Data From Memory (VREAD)

The **VREAD** command Returns the value of the specified variables, expressions, or arrays. The **VREAD** command syntax is:

VREAD *variable*
expression
array_name [(*element*)]

The *variable* parameter specifies the variable name. The *expression* parameter specifies any valid number or numeric expression (enclosed in parentheses). The *array_name* [(*element*)] parameter specifies the array name and starting element.

When *element* is not specified, **VREAD** returns the values of all elements in the array. When *element* is specified, **VREAD** returns the value of the specified element. For example, if VOUT is a four-element array, the element numbers are 0, 1, 2, and 3. Then, **VREAD VOUT(2)** returns the value of element #2, while **VREAD VOUT** returns the values of all four elements in VOUT.

VREAD transfers variable or array values from memory to the output buffer. Always turn memory mode OFF (**MEM OFF**) before executing **VREAD**. Otherwise, **VREAD** attempts to write data into memory and an error occurs. Memory mode is automatically turned off as required for variables, but not for arrays.

Example 8-4: Reading Array Value (VRED8)

This program stores the current address of the HP 3245A in array A (defined as a 3-element INTEGER array), turns memory mode OFF, uses **VREAD** to transfer the value in element A(0) to the output buffer, and then enters the address into the controller CRT.

```
10   !file VRED8
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER A(2)"      !Define array A
70   OUTPUT 709;"MEM A"             !Store results in A
80   OUTPUT 709;"ADDRESS?"          !Read current HP-IB address
90   OUTPUT 709;"MEM OFF"           !Turn memory mode OFF
100  OUTPUT 709;"VREAD A(0)"        !Trans value in A(0) to out buffer
110  ENTER 709;A                    !Enter address
120  PRINT "Address =";A            !Display address
130  END
```

A typical return for this program follows. Note that if line 90 is deleted, the program will not complete since **VREAD** (line 100) attempts to write data into array A, rather than reading data from the array.

```
Address = 9
```

Data Formats (OFORMAT/ BLOCKOUT)

When an HP 3245A command generates data, the data is sent to the front panel display if the command or subroutine was entered from the front panel keyboard. If the command or subroutine was executed from the controller, the data is sent to the output buffer (**MEM OFF**) or to the HP 3245A memory (using the **MEM** command). Some controllers may require EOI termination (use the **END** command).

Selecting Output Data Format (OFORMAT)

The HP 3245A returns numeric results in either ASCII or binary format, depending on the **OFORMAT** command. At power-on, the ASCII output format is selected. In ASCII format, multiple results from one command are separated by commas. The **OFORMAT** *format* parameter specifies ASCII or binary output format, as shown in Table 8-2 The power-on *format* is ASCII.

In ASCII mode, multiple results from one command are separated by commas. A carriage return and line feed are sent after the final result. If **END** is ON, an EOI is sent with the line feed. The binary mode is useful if your controller uses 16-bit integer and 64-bit IEEE-754 format for its internal binary representation The **BLOCKOUT** command determines whether the IEEE-728 Block A header is sent before each output.

Table 8-2. Output Data Formats

format	Definition
ASCII	6-digit signed notation (INTEGER results). 8-digit scientific notation (REAL results).
BINARY	16-bit, 2's complement notation (INTEGER results). IEEE-754 64-bit notation (REAL results).

Setting Binary Mode Header (BLOCKOUT)

The **BLOCKOUT** command enables or disables the block output mode At power-on, the block output mode is enabled (**BLOCKOUT ON**) and binary outputs are preceded with an IEEE-728 Block A header followed by a *cr lf*. With the blockout mode disabled (**BLOCKOUT OFF**), only the binary number is output (no header or *cr lf*).

Binary mode is useful if your controller uses 16-bit integer and/or 64-bit IEEE-754 format for internally storing data. Binary data outputs are preceded with an IEEE-728 Block A header followed by a *cr lf*. The Block A header consists of four bytes: the "#" sign, the letter "A", and two bytes which indicate the number of bytes of data to follow. These two bytes represent one 16-bit number; the first byte represents bits 15 through 8 and the second byte represents bits 7 through 0. Refer to the **BLOCKOUT** command in the HP 3245A Command Reference Manual for an example program using **BLOCKOUT**.

Interrupts and Service Requests

This section describes status requests and service requests for the HP 3245A. It includes:

- Status register/RQS mask description.
- Reading the status register.
- Enabling service requests.

Interrupts/ Service Requests Overview

The HP 3245A can be programmed to interrupt the controller when certain specified events or conditions occur. (Of course, the controller must be programmed to respond to the interrupt.) The Service Request (SRQ, a function of the HP-IB interface) implements the interrupt action and is independent of all other HP-IB activity.

Interrupt Events/Conditions

The HP 3245A can be programmed to interrupt for one or more of five events/conditions shown in Table 8-3.

Table 8-3. Interrupt Events/Conditions

DATA AVAILABLE	The HP 3245A can interrupt the controller when data is available in the output buffer.
USER SERVICE REQUEST	The HP 3245A can interrupt when an SRQ command is executed from the controller, from within an HP 3245A subroutine, or when the command is entered from the front panel (via the MENU keys).
LOCAL	The HP 3245A can interrupt at power-on, when it is reset, or when it enters the Local mode.
READY	The HP 3245A can interrupt when it is in the READY state. That is, when the input buffer is empty and the HP 3245A is not executing a command or a CALL subroutine.
ERROR	The HP 3245A can interrupt when an error condition occurs.

Status and Service Requests Operation

Figure 8-3 summarizes HP 3245A operation for status and service requests. The HP 3245A status register monitors the five defined events and conditions (DATA AVAILABLE, USER SERVICE REQUEST, etc.). When a defined event or condition occurs, the corresponding bit in the status register is set (1). The **STA?** and **STB?** commands can be used to determine the bits which are currently set.

Even though a status register bit is set, a service request (SRQ) to the controller is not generated unless that bit is "unmasked". The **RQS** command unmasks the bit(s) in the RQS mask register so that specified event(s)/condition(s) can generate an SRQ. The **RQS?** command returns the decimal value of the unmasked bits in the RQS mask.

For example, status register bit 0 monitors DATA AVAILABLE. When data is available in the output buffer, status register bit 0 is set. If bit 0 is also unmasked by **RQS**, an SRQ is generated when data becomes available. (The controller must be programmed to take action as a result of the SRQ).

Two types of service requests can be generated by the HP 3245A: programmed (front panel) requests with the **SRQ** command or power-on SRQ with the **PONSRQ** command. With programmed service requests, an SRQ is sent to the controller when an **SRQ** command is executed from the controller, from an HP 3245A subroutine, or from the front panel (if bit 2 is unmasked). When power-on SRQ is enabled, RQS mask bit 3 is unmasked, so an SRQ is generated at power-on.

Status Register/ RQS Mask (RQS/RQS?)

The HP 3245A status register monitors and records the status of the five interrupt conditions (DATA AVAILABLE, etc.). The RQS mask determines which interrupt conditions will actually interrupt the controller when they occur. Figure 8-4 shows the status register and RQS mask bits, bit definitions, decimal values, and set/clear conditions.

The Status Register

As shown in Figure 8-4, although the HP 3245A status register is a 16-bit register, only bits 0, 2, 3, 4, 5, and 6 are used. Each bit has an associated decimal value (bit 0 = 1, bit 2 = 1, ..., bit 6 = 64). Note that bit 2 is not used. When the event or condition associated with a bit occurs, the bit is set (1). For example, when data is available in the output buffer, bit 0 is set. You can use the **STA?**, **STB?**, or HP-IB **SPOLL** command to read the current state of the status register bits (refer to "Reading Status Register (STA?/STB?/SPOLL for details)

The RQS Mask

The RQS mask determines which interrupt condition(s) will interrupt the controller. The **RQS** *unmask_value* command "unmasks" the status register bit(s) to generate an interrupt when the condition(s) occur. The bit definitions for the RQS mask are the same as for the status register, so only bits 0, 2, 3, 4, and 5 can be unmasked (bit 1 is not used and bit 6 is always unmasked). When status register bit 0, 2, 3, 4, or 5 is set AND the bit is unmasked, bit 6 in the RQS mask is set. This, in turn, sets bit 6 in the status register (SRQ SENT). When bit 6 in the status register is set (TRUE), the HP 3245A asserts the HP-IB SRQ line and the front panel SRQ annunciator turns ON.

Set/Read RQS Mask Register (RQS/RQS?)

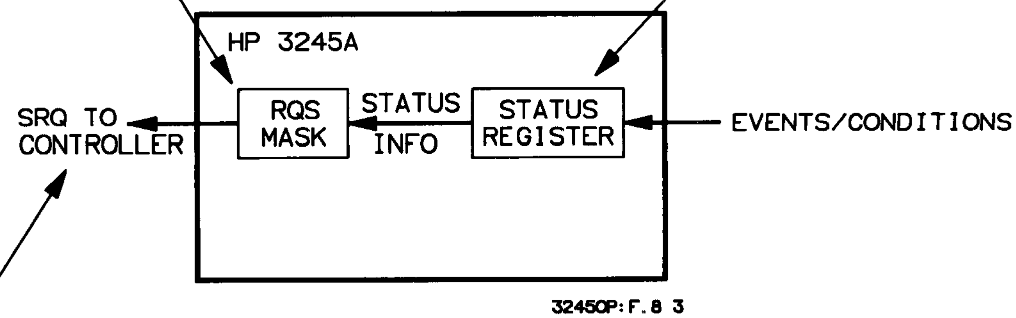
RQS sets bit(s) in the RQS mask register to determine what event(s) or condition(s) will generate an SRQ to the controller. When an event monitored by the status register occurs and the corresponding bit is "unmasked" (by RQS), an SRQ is sent to the controller.

Only status register bits 0, 2, 3, 4, and 5 can be unmasked. RQS? returns the decimal value of the unmasked bits in the RQS mask.

Read Status Register State (STA?/STB?)

The status register constantly monitors specified events and/or conditions. When the specified event or condition occurs, the associated bit in the status register (0, 2, 3, 4, 5, or 6) is set (1).

Both STA? and STB? return the weighted sum of all bits set in the status register. However, STA? clears (0) bits 2 and 3 when the command executes, while STB? does not clear these bits.



Generate SRQ to Controller (SRQ/PONSRQ)

When status register bit 2 (user service request) is unmasked by RQS, executing SRQ (usually from the front panel) sets the HP-IB SRQ line TRUE to tell the controller the HP 3245A has requested service.

PONSRQ (power-on SRQ) enables or disables assertion of the HP-IB SRQ line at power-on. When power-on SRQ is enabled (PONSRQ ON), the RQS mask is set to a value of "8" at power-on, thus allowing SRQ to be asserted. With PONSRQ OFF, the mask register is set to "0" at power-on.

Figure 8-3. Status/Service Requests Operation

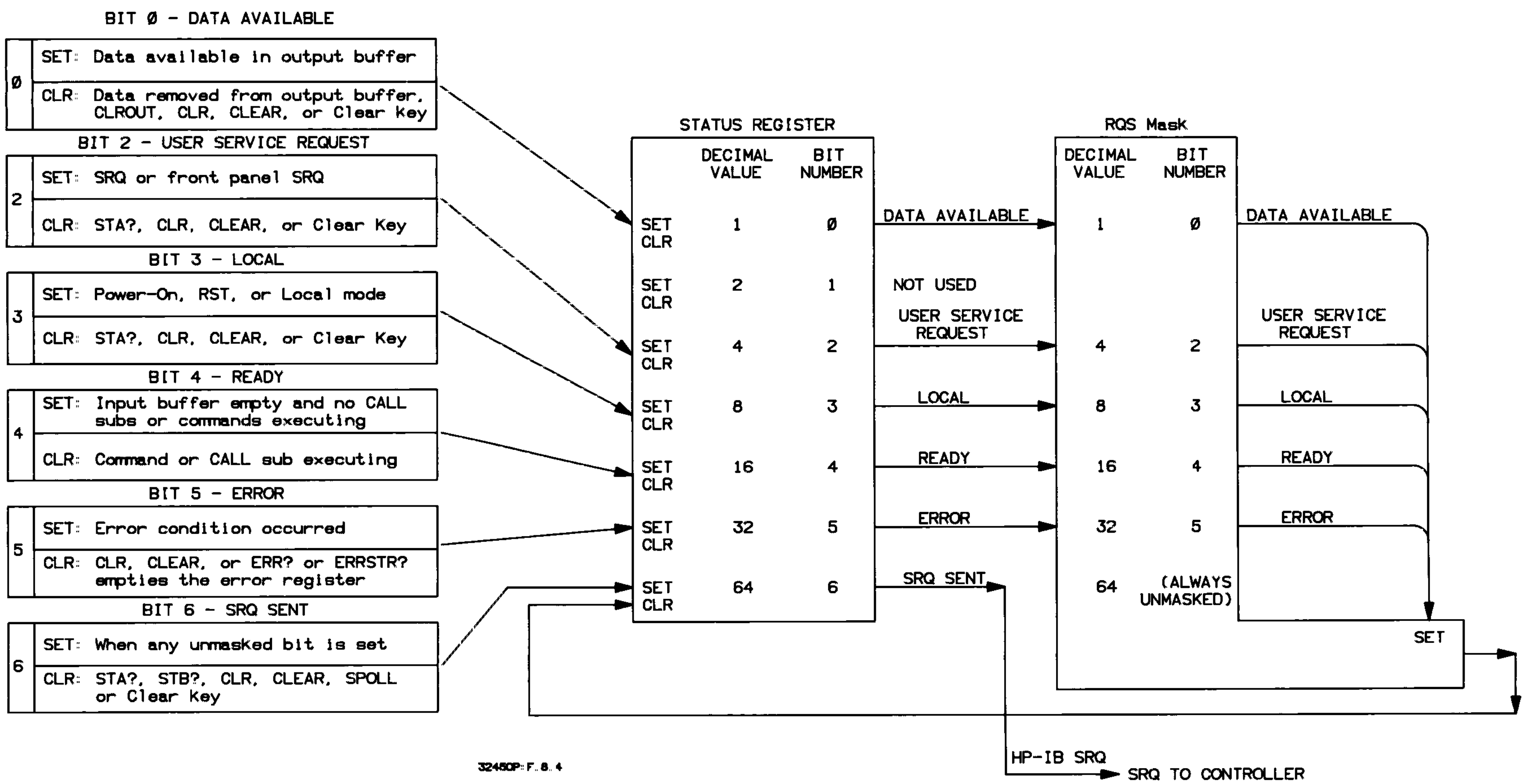


Figure 8-4. Status Register/RQS Mask

To set the RQS mask, use **RQS** *unmask_value* where *unmask_value* is a decimal value equal to the sum of the bit values you want to unmask. For example, to unmask bit 0 (DATA AVAILABLE, value = 1) and bit 5 (ERROR, value = 32), use **RQS 33** (32 + 1). Then, whenever data is available in the output buffer or when an error condition occurs, the HP 3245A will attempt to interrupt the controller.

Reading the RQS Mask Value

The **RQS?** command returns a decimal value equal to the sum of the bit values which are unmasked. For example, if bit 0 (value = 1), bit 4 (value = 16), and bit 5 (value = 32) are unmasked, **RQS?** returns 49. The following program uses **RQS?** to return the current value of the RQS mask. (Note that you will need to use **STA?**, **STB?**, or **SPOLL** to read the status register bits.)

Example 8-5: Reading RQS Mask Value (RQSQ8)

This program uses **RQS?** to display the unmasked bit(s) in the RQS mask.

```
10   !file RQSQ8
20   !
30   OUTPUT 709;"RQS 48"                !Unmask RQS mask bits
40   OUTPUT 709;"RQS?"                  !Query RQS mask value
50   ENTER 709;A                        !Enter value
60   PRINT "RQS Mask Value = ";A        !Display value
70   PRINT "Unmasked bits are: "        !Header
80   PRINT                              !Space
90   IF BINAND(A,1) THEN PRINT "DATA AVAILABLE"      !Bit 0
100  IF BINAND(A,4) THEN PRINT "USER SERVICE REQUEST" !Bit 2
110  IF BINAND(A,8) THEN PRINT "LOCAL"    !Bit 3
120  IF BINAND(A,16) THEN PRINT "READY"   !Bit 4
130  IF BINAND(A,32) THEN PRINT "ERROR"   !Bit 5
140  END
```

Since line 30 unmasks bits 4 and 5, a typical return is:

```
RQS Mask Value = 48
Unmasked bits are:

READY
ERROR
```

Controller Response to a Service Request

When an SRQ message is sent, the controller conducts a serial poll (SPOLL) of each instrument on the bus which is enabled to assert SRQ. When an instrument is polled, it returns its (8-bit) status register value. If bit 6 is TRUE, the controller then knows that the instrument interrupted and looks at other bits to determine the cause of the interrupt.

Status register bit 6 (SRQ SENT) is set whenever any unmasked bit (0, 2, 3, 4, or 5) in the register is set. Bit 6 is cleared when the unmasked bits are cleared, when the controller executes a Serial Poll, or when the event setting a status register bit is cleared (for example, reading the error register to empty the error register and clear bit 5).

NOTE

The **CLR** *command and the HP-IB* **CLEAR** *clear certain bits in the status register. Refer to Table 8-4 for details.*

Reading Status Register (STA?/STB?/ SPOLL)

You can use the **STA?**, **STB?**, or HP-IB **SPOLL** command to read the current state of the status register. **STA?**, **STB?**, and **SPOLL** return a decimal value which is the decimal sum of the status register bits which are true (set). (A "0" is returned if no status register bits are set.)

In addition to reading the status register bits set, **STA?**, **STB?**, and **SPOLL** clear certain bits depending of the conditions set when the command is executed. Table 8-3 shows the status register bits, decimal value, definition, and events/conditions which set and clear the bits.

Reading/Clearing Status Register With STA?

The **STA?** command both reads the status register bits set and clears certain bits in the register after the read has been made. The **STA?** command returns a decimal value equal to the sum of the decimal values of all status register bits set (both masked and unmasked). For example, if bit 0 (value = 1) and bit 4 (value = 16) are set, **STA?** returns 17.

STA? also clears status register bits after the read has been made. From Table 8-4, **STA?** clears bit 2 (USER SERVICE REQUEST) and bit 3 (LOCAL) after the command executes. For example, if both of these bits are unmasked, and both bits are set, status register bit 6 (SRQ SENT) is also set and an SRQ is sent to the controller. In this case, **STA?** returns 76 (64 + 8 + 4) and clears bits 2, 3, and 6.

Executing **STA?** clears the bits 2 and 3 which are set by an event (USER SERVICE REQUEST and LOCAL). Bits 4 and 5 which reflect state (condition) (DATA AVAILABLE, READY, and ERROR) are not cleared by **STA?**. Bit 6 (SRQ SENT) and the HP-IB SRQ line are cleared ONLY if no unmasked bits remain set.

NOTE

When **STA?** *or* **STB?** *is executed, bit 4, READY, will always show cleared since the HP 3245A is busy executing the command. Use* **READY?** *or* **SPOLL** *to check the ready status of the HP 3245A.*

At power-on, both **STA?** and **STB?** return the value "8" if power-on SRQ is disabled (**PONSRQ OFF**). If power-on SRQ is enabled (**PONSRQ ON**), "72" is returned and the HP-IB SRQ line is asserted at power-on. Refer to "Enabling Service Requests (SRQ/PONSRQ)" for details.

Table 8-4. Status Register Bits

Bit	Value	Definition	Set by:	Cleared by:
0	1	DATA AVAILABLE	Data available in the output buffer.	CLR, CLEAR, CLROUT, or Clear key or when data is removed from output buffer
1	2	NOT USED		
2	4	USER SERVICE REQUEST	SRQ command from controller or from HP 3245A.	STA?, CLR, CLEAR, or Clear key.
3	8	LOCAL	Power-on, RST, or Local mode set.	STA?, CLR, CLEAR, or Clear key.
4	16	READY	Input buffer is empty and no command or CALL subroutine is executing.	Command or CALL sub-routine is executing.
5	32	ERROR	An error condition occurring.	CLR, CLEAR or when ERR? or ERRSTR? clears error register.
6	64	SRQ SENT	When any unmasked bit (0, 2, 3, 4, or 5) in status register is set.	STA?, STB?, CLR, CLEAR, SPOLL, or Clear key.*
7-16		NOT USED		

* = Not cleared by STA?/STB? for every condition.

Example 8-6: Reading Status Register (STAQ8)

This program uses **STA?** to read the state of the status register. The value returned is the weighted decimal sum of the status register bits which are set.

```
10  !file STAQ8
20  !
30  OUTPUT 709;"STA?"                    !Read status register
40  ENTER 709;A                          !Enter value
50  PRINT "Status reg bits set = ";A     !Display value
60  END
```

If bit 2 (value = 4) and bit 6 (value = 64) are set, the sum of the values (68) is returned, as shown. For this program, since bit 4 or bit 5 was not set **STA?** clears both bit 2 and bit 6 when the command executes.

```
Status reg bits set = 68
```

Example 8-7: Interrupt Controller on Error (INTR8)

This program interrrupts the controller when an HP 3245A error occurs. To execute the program, run the program and then press the **Freq** key followed by the **Enter** key on the front panel to introduce a deliberate error. When the error is generated, the display shown after the program appears on the controller CRT.

```
10   !file INTR8
20   !
30   CLEAR 709                          !Clear HP 3245A
40   DIM C$[256]                        !Dim controller array
50   ON INTR 7 GOTO 110                 !Set SRQ interrupt
60   ENABLE INTR 7;2                    !Enable cont interrupt
70   OUTPUT 709;"RST"                   !Reset HP 3245A
80   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
90   OUTPUT 709;"RQS 32"                !Unmask bit 5 (ERROR)
100  GOTO 100                           !Loop until interrupt
110  OUTPUT 709;"STA?"                  !Read status register
120  ENTER 709;A                        !Enter value
130  PRINT "Bits set after error =";A   !Display value
140  OUTPUT 709;"STA?"                  !Read status register
150  ENTER 709;B                        !Enter value
160  PRINT "Bits set after STA?  =";B   !Display value
170  OUTPUT 709;"ERRSTR?"               !Read error register
180  ENTER 709;C$                       !Enter error string
190  PRINT "Error:";C$                  !Display error string
200  OUTPUT 709;"STA?"                  !Read status register
210  ENTER 709;D                        !Enter value
220  PRINT "Bits set after ERRSTR? =";D !Display value
230  END
```

A typical return for this program follows. When the error occurred, bit 5 (ERROR) was set. Since bit 5 was unmasked by **RQS 32** (line 90), bit 6 was also set and an SRQ sent to the controller. Then, **STA?** (line 110) returned the weighted decimal value of the bits set (96 = 32 + 64).

Next, to demonstrate that **STA?** does not clear bits 5 or 6, **STA?** was executed again (line 140) to return the same result (96). Then, **ERRSTR?** (line 170) was executed to clear the error register and display the error on the controller CRT. Since the error register was cleared (only one error was committed) by **ERRSTR?**, status register bit 5 was cleared. This, in turn, cleared bit 6 since no unmasked bits remain set.

```
Bits set after error = 96
Bits set after STA?  = 96
Error:     2,"SYNTAX -- FREQ ;     Expected: a number ,(,-,+"
Bits set after ERRSTR? = 0
```

Reading/Clearing Status Register with STB?/SPOLL

As noted, **STA?**, **STB?**, and **SPOLL** all return the weighted sum of the status register bit which are set (both masked and unmasked), but **STA?** also clears bit 2 (USER SERVICE REQUEST) and bit 3 (LOCAL). If desired, you can use **STB?** to read the status register bits without clearing bit 0, 2, 3, 4, or 5.

However, since **STB?** interrupts the HP 3245A microprocessor, when **STB?** is used the HP 3245A appears to be busy (bit 4 clear). Since **SPOLL** reads the status register without interrupting the microprocessor, you can use **SPOLL** to monitor the ready state of the HP 3245A.

If the HP-IB SRQ line is set TRUE when **SPOLL** is sent, all bits in the status register are cleared if the event or condition which set the bit(s) is no longer present. If the SRQ line is FALSE when **SPOLL** is sent, the status register contents are not changed.

If there is data in the output buffer when **SPOLL** is sent, the data remains intact. In contrast, if there is data in the output buffer when **STB?** is sent, that data is overwritten by the status data (unless **OUTBUF ON** is set). An example program using **SPOLL** to read the status register follows.

Example 8-8: Reading Status Using SPOLL (SPOL8)

This program reads the status register and returns the weighted value of the bits set.

```
10  !file SPOL8
20  !
30  P=SPOLL (709)     !Perform serial poll, put result in P
40  PRINT P           !Display decimal value of bits set
50  END
```

Enabling Service Requests (SRQ/PONSRQ)

In addition to the five interrupt events/conditions, you can also program the HP 3245A for two types of service requests (SRQ): programmed (front panel) SRQ using the **SRQ** command or power-on SRQ using the **PONSRQ** command.

Programmed Service Requests (SRQ)

When the **SRQ** command is sent from the controller, from within an HP 3245A subroutine, or from the front panel (via the MENU keys), a programmed SRQ is generated. If status register bit 2 (USER SERVICE REQUEST) is unmasked (by **RQS 4**), executing **SRQ** sets the HP-IB SRQ line TRUE to signal the controller that the HP 3245A has requested service.

Example 8-9: Front Panel SRQ Interrupt (SRQI8)

This program uses **SRQ** entered from the front panel to generate an interrupt to the controller. The program loops continuously at line 70 until an **SRQ** command is executed from the front panel. When **SRQ** is executed, the front panel SRQ annunciator turns ON and the message shown following the program is displayed on the controller CRT. To execute the program, run the program and then enter **SRQ** using the MENU key operation.

```
10   !file SRQI8
20   !
30   CLEAR 709                       !Clear HP 3245A
40   ON INTR 7 GOTO 100              !Specify intr routine
50   ENABLE INTR 7;2                 !Enable controller interrupt
60   OUTPUT 709;"RST"                !Reset HP 3245A
70   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
80   OUTPUT 709;"RQS 4"              !Unmask bit 2 (USER SERV REQ)
90   GOTO 90                         !Loop until SRQ executed
100  PRINT "HP 3245A Interrupt"      !Display interrupt message
110  OUTPUT 709;"STA?"               !Read status register
120  ENTER 709;A                     !Enter bits set
130  PRINT "Status reg bits set =";A !Display bits set
140  END
```

When SRQ is executed from the front panel, a typical display is as follows. The returned value of 68 shows that bit 6 (SRQ SENT: value = 64) and bit 2 (USER SERVICE REQUEST: value = 4) are set (68 = 64 + 4).

```
HP 3245A Interrupt
Status reg bits set = 68
```

Enabling Power-On SRQ Interrupt (PONSRQ)

You can use the **PONSRQ** (power-on SRQ) command to program the HP 3245A to interrupt the controller when power is applied to the instrument. **PONSRQ ON** enables the HP 3245A to interrupt the controller when power is applied to the instrument, while **PONSRQ OFF** disables the instrument from interrupting at power-on.

With **PONSRQ ON** (power-on SRQ enabled), the RQS mask is set to value "8" at power-on which unmasks bit 3 (LOCAL) and enables an SRQ when the instrument is turned on. With **PONSRQ OFF** (power-on SRQ disabled), the RQS mask is set to value "0" at power-on, so a power-on SRQ is not generated.

Example 8-10: Power-On SRQ Interrupt (PONS8)

This program shows one way to use **PONSRQ** for power-on SRQ interrupt.

```
10   !file PONS8
20   !
30   OUTPUT 709;"RST"                    !Reset HP 3245A
40   OUTPUT 709;"STA?"                   !Clear possible pending SRQ
50   ENTER 709;Temp                      !Read value
60   OUTPUT 709;"PONSRQ ON"              !Enable power-on SRQ
70   ON INTR 7 GOTO 110                  !Set interrupt routine
80   ENABLE INTR 7;2                     !Enable cont interrupt
90   PRINT "Cycle HP 3245A Power"
100  GOTO 100                            !Loop until interrupt
110  OUTPUT 709;"STA?"                   !Read status register
120  ENTER 709;A                         !Enter value
130  IF BIT(A,3) THEN                    !Test for power-on SRQ
140    PRINT "HP 32454A Power-On SRQ"
150  END IF
160  END
```

A typical return is:

```
HP 3245A Power-On SRQ
```

Contents

Chapter 9
Other Instrument Functions

Chapter 9
Other Instrument Functions

Chapter Contents

This chapter describes general instrument functions for the HP 3245A. The chapter contents are:

- **Front Panel Functions** shows how to redefine front panel keys; enable and disable the display; and monitor channel status and condition.

- **General Query Commands** shows how to use the HELP function; how to read error codes and messages; and how to use identification queries.

- **Timing Functions** shows how to set and read the HP 3245A internal clock time and how to set a wait period for a specified time or trigger event.

- **Storing/Recalling States** shows how to store and recall local states; set and recall remote states; and store the state of a channel into another channel.

- **Memory Management** shows how to set memory mode; describes HP 3245A internal task priority; and gives guidelines for efficient memory usage.

- **Test/Calibration** summarizes test and calibration procedures for the HP 3245A.

Front Panel Functions

This section describes some HP 3245A front panel functions, including redefining front panel keys, enabling/disabling the display, and monitoring states/conditions. Refer to Chapter 6 - Front Panel Operation for additional information and keystroke functions/operation.

Redefine Keys (DEFKEY)

The **DEFKEY** *key*, *string* command allows a user-defined string to be assigned to a NUMERIC (**0** - **9**) key on the HP 3245A front panel. **DEFKEY?** *key* returns the current definition for the specified NUMERIC key.

The *key* parameter can be any of the front panel NUMERIC keys (**0** through **9**) When redefining a NUMERIC key from the controller, the *key* parameter must be enclosed in double quotation marks (" "*key*" ") or apostrophes ('*key*') Quotation marks/apostrophes are not required when redefining a key from the front panel.

The *string* parameter (up to 75 characters) is assigned to the specified NUMERIC *keys*. When **DEFKEY** is executed from the front panel, the *string* parameter must be enclosed in quotation marks ("*string*"). Use `OUTPUT 709;"DEFKEY 1 '''"` to return a NUMERIC key to its default definition.

Redefined NUMERIC keys may contain complete commands or command fragments. Multiple commands may be stored in a single key when separated by a semicolon (;). Up to 75 characters can be stored in a NUMERIC key string. The strings defined by the **DEFKEY** command are stored in continuous memory and are retained when power is removed from the HP 3245A An HP-IB example follows. Refer to Chapter 6 - Front Panel Operation for a front panel keystroke sequence to redefine a key.

Example 9-1: Redefining Front Panel Key (DEFKY9)

This program redefines NUMERIC Key 5 with the string APPLY DCV .1 and returns the redefined function to the controller CRT. When NUMERIC key 5 is pressed, the HP 3245A outputs 0.1 DCV from the use channel. To return key 5 to its original definition, use `OUTPUT 709;"DEFKEY 5 ''"`. Note that line 80 places the HP 3245A in Local mode With Local mode, you can press the **5** key followed by the **Enter** key to actually output 0.1 VDC from channel A.

```
10  !file DEFKY9
20  !
30  DIM A$[50]                              !Dimension controller array
40  OUTPUT 709;"DEFKEY 5,'APPLY DCV .1'"    !Redefine NUMERIC key 5
50  OUTPUT 709;"DEFKEY? 5"                  !Read key 5 new definition
60  ENTER 709;A$                            !Enter definition
70  PRINT A$                                !Display definition
80  LOCAL 709                               !Place HP 3245A in Local
90  END
```

A typical return is:

```
APPLY DCV .1
```

Enable Display (DISP)

DISP (or **DSP**) enables or disables the front panel display. This command can also be used to display a string, the contents of a variable, a number, or a numeric expression. **DISP?** (or **DSP?**) returns a quoted string containing the contents of the front panel display. The command syntax is:

DISP *mode*

or

DISP [MSG] [*message*]

The *mode* parameter enables or disables the front panel display. The parameters are OFF and ON. Power-on *mode* = ON. With **DISP ON**, the front panel display is enabled. With **DISP OFF**, the front panel display is disabled and dashes are displayed.

MSG is an optional parameter which tells the HP 3245A a message is to be displayed on the front panel *message* may be a quoted string of characters (quotation marks are not displayed), the value of a variable, a number, or a numeric expression (enclosed in parentheses).

When **DISP** is executed from the controller, *message* must be enclosed in double quotation marks (" "*message*" ") or apostrophes ('*message*'). When using the controller, the message can be entered in either upper case or lower case letters.

When disabled (**DISP OFF**), the front panel displays all dashes and all annunciators are OFF. The display can be reenabled with **DISP ON** or by pressing any front panel key which changes the display. With the display disabled, command execution speed increases since the HP 3245A does not continually update the front panel display. To clear the front panel display, send "**DISP** ' ' "

DISP? (or **DSP?**) returns a quoted string containing the contents of the front panel display, including all characters outside the 15-character display window (up to 256 characters). When **DISP?** is executed from the controller, the response is sent to the output buffer in the default ASCII format.

NOTE

DISP? *is useful for reading* **MON** *displays. The* **MON** *command normally displays results on the front panel. However, you can use* **DISP?** *to send the results to the controller. Refer to "Monitor States (MON)" for an example program.*

Monitor States (MON)

The **MON** command displays on the front panel display command keywords entered over the HP-IB or the state of channel A or channel B The command syntax is:

MON **HPIB**
STATE *ch*
NONE
OFF

MON HP-IB displays command keywords (with parameters), as the HP 3245A receives and executes commands from the controller, **MON STATE** *ch* displays the state of the specified channel (channel A or channel B), and **MON OFF** or **MON NONE** disables the monitoring mode on the specified use channel

A **MON** command cancels any monitoring processes from previous **MON** commands (only one monitor mode is in effect at a time). For example, **MON HPIB; MON STATE 0** cancels monitoring of the HP-IB inputs **MON STATE 0** monitors channel A, while **MON STATE 100** monitors channel B. To disable the monitor mode, press the front panel **Clear** key or execute **MON OFF**.

With **MON STATE** (power-on or **RST** command), the front panel display contains the elements shown in Table 9-1 (scroll the display with the **Right Arrow** or **Down Arrow** key to read each element in the display).

Table 9-1. Power-On State Display

Information	Description
Arrow to channel A	Channel A is use ch
0.000000E+0DCV	0 V DCV output
FREQ 1000.000 1000.000	Output freq in Hz*
DCOFF 0.000000E+0	DC offset in Volts
DUTY 50.0	Duty cycle percentage
PANG 0.000	Phase angle**
RANGE 1.0	Voltage/current range
ARANGE ON	Autorange setting
TERM FRONT	Output terminal
IMP 0	Output impedance
DCRES HIGH	DC resolution mode
TRIGMODE OFF	Triggering mode
TRIGIN HIGH	Trigger input source
TRIGOUT OFF	Output trigger mode
SYNCOUT OFF	SYNC destination
REFIN INT	Ref freq input source
REFOUT EXT	Ref freq output dest
DELAY 0.04	Output delay in sec

* = second value applies to dual-freq mode only.
** = applies to synchronized mode only.

Example 9-2: Monitoring Channel State (MONST9)

This program monitors the state of channel A and displays the results on the controller CRT. (Note that the state of channel A is also displayed on the front panel display.)

```
10   !file MONST9
20   !
30   DIM Message$[256]            !Dimension string array
40   OUTPUT 709;"RST"             !Reset HP 3245A
50   OUTPUT 709;"MON STATE 0"     !Monitor channel A state
60   OUTPUT 709;"DISP '' "        !Clear front panel display
70   WAIT 1                       !Wait for state on display
80   OUTPUT 709;"DISP?"           !Read front panel display
90   ENTER 709;Message$           !Enter message
100  PRINT Message$               !Print message
110  END
```

Since **RST** sets channel A to its power-on condition, a typical return is as follows. Refer to Table 9-1 for a description of the state elements.

```
"0.000000E+0DCV, FREQ 1000.000 1000.000, DCOFF 0.000000E+0,
DUTY 50.0, PANG 0.000, RANGE 1.0, ARANGE ON, TERM FRONT, IMP
0, DCRES HIGH, TRIGMODE OFF, TRIGIN HIGH, TRIGOUT OFF,
SYNCOUT OFF, REFIN INT, REFOUT EXT, DELAY 0.04"
```

General Query Functions

This section describes general query functions for the HP 3245A, including the HELP function, error code/message queries, and identification (ID) queries.

HELP Function (HELP)

The **HELP** [*topic*] command provides brief descriptions of HP 3245A commands, where *topic* is the keyword for the command specified. **HELP** also provides explanations of some command parameters. For example, **HELP ABORT** provides information on the **ABORT** command. Note that the **HELP** command can be entered from the front panel or via HP-IB. Use the front panel arrow keys to scroll the message displayed on the front panel display.

Error Messages (ERR?/ ERRSTR?)

The HP 3245A has an error register which can store up to four error messages. You can use the **ERR?** command to return the error code of the most recent error or can use the **ERRSTR?** command to return the error code and string of the most recent error

Error Handling

Whenever an error is generated, the HP 3245A front panel displays the error number, a brief description of the error, and (if a command syntax error occurred), the string that was in error. Errors are stored in a first-in-first-out list which holds the first four errors which occurred.

You can read the list, one at a time, using **ERR?** or **ERRSTR?**. **ERR?** returns the numeric code of the error, while **ERRSTR?** returns the error code number and the error message When an error occurs, the front panel **ERR** annunciator is turned ON. Refer to the HP 3245A Command Reference Manual for a list of error messages.

Any error sets the ERROR bit (bit 5) in the status register (refer to Chapter 8 - Input/Output Operations). Power-on, **RESET**, **CLR**, HP-IB**CLEAR** or reading the entire error list clears the ERROR bit (HP-IB **SPOLL** does not clear the ERROR bit). **RESET**, **CLR**, or **CLEAR** also clear the entire error register. There are four main categories of errors: syntax, execution, math, and hardware errors. Table 9-2 summarizes these errors and shows HP 3245A response to the errors.

Reading Error Codes (ERR?)

The **ERR?** command returns the error code of the most recent error, deletes that error code from the error register, and clears the status register error bit (bit 5) after all errors are read Refer to Chapter 8 - Input/Output Operations for a description of the HP 3245A status register

One integer is returned corresponding to the most recent error in the error register. Errors are read in a first-in-first-out fashion (i.e., the first error to occur is the first error to be read). If no error codes are in the error register, "0" is returned. Up to four errors will be stored in the error register.

Table 9-2. HP 3245A Error Handling

Syntax Errors

Definition	HP 3245A receives illegal command from controller or from front panel. Typical errors are illegal characters, misspelled headers or parameters, or illegal number formats.
Response	For syntax errors, HP 3245A skips over characters until it reaches a command terminator (cr, lf, or EOI). The processor then resumes execution of following commands. For errors in a subroutine statement, the processor ignores the command, logs the error, and stores the rest of the subroutine. For errors in a subroutine execution statement, the subroutine is not stored.

Execution Errors

Definition	A syntactically correct command includes a parameter range error or other data which is outside the defined parameter range.
Response	Aborts the command and continues with the next command. If the error occurs during a subroutine execution, the HP 3245A logs the error and aborts the sub.

Math Errors

Definition	A math error such as dividing by zero trig functions out of range, etc.
Response	Aborts the command and continues with the next command. If the error occurs during subroutine execution, the HP 3245A logs the error and aborts the sub.

Hardware Errors

Definition	Errors which cause the HP 3245A to power-up incorrectly or not at all.
Response	Displays hardware errors which occur.

ERR? reads the error messages and clears the error register one error at a time Errors that occur while the register is full are displayed as usual, but are not stored in the error register. Reading an error message makes space available in the register for the next error. The status register error bit (bit 5) and the front panel ERR annunciator are cleared when the error register is empty. Refer to the **ERR?** command in the HP 3245A Command Reference Manual for an example program using **ERR?**.

Reading Error Messages (ERRSTR?)

The **ERRSTR?** command returns the error code and string of the most recent error, deletes that error from the error register, and clears the status register error bit after all errors are read. The status register error bit (bit 5) and the front panel ERR annunciator are cleared when all errors have been read from the error register.

ERRSTR? returns a number, a comma, and a quoted string which may contain up to 256 characters. Errors are returned in a first-in-first-out fashion (i.e., the first error to occur is the first error to be read). If no error codes are in the error register, `" 0, NO ERROR"` is returned.

ERRSTR? reads the error messages and clears the error register one error at a time. Errors that occur while the register is full are displayed as usual, but are not stored in the error register. Reading an error message makes space available in the register for the next error. An example program using **ERRSTR?** follows.

Example 9-3: Reading Error Messages (ERRMG9)

This program uses the **ERRSTR?** command within a loop to read and print all error messages in the error register.

```
10  !file ERRMG9
20  !
30  DIM Message$[256]            !Dimension controller array
40  REPEAT                       !Begin loop
50    OUTPUT 709;"ERRSTR?"       !Read error register
60    ENTER 709;Code,Message$    !Enter code, message
70    PRINT Code,Message$        !Display code, message
80  UNTIL Code=0                 !Loop until all errors are read
90  END
```

A typical return for no errors follows.

```
0, "NO ERROR"
```

ID Queries (ID?/IDN?/CTYPE?/REV?)

Four commands apply to HP 3245A identification. The **ID?** command returns the HP 3245A model number, the **IDN?** command returns the instrument identity, the **CTYPE** command returns the channel type, and **REV?** returns the firmware revision code.

Model/Identity Queries (ID?/IDN?)

ID? returns the HP 3245A model number (HP3245), while **IDN?** returns the manufacturer's name, the model number, the serial number, and the firmware revision number for the HP 3245A. **IDN?** returns four lines of information. Line 1 is the manufacturer's name (Hewlett-Packard), line 2 is the model number (3245), line 3 is the instrument serial number (always 0), and line 4 is the firmware revision number.

The firmware revision number is a four-digit year and date code. The code has the form "yyww", where "yy" is the year minus 1960, and "ww" is the week number of that year. For example, 2833 means that the latest firmware revision was the 33rd week of 1988 (1988 - 1960 = 28). An example program using **IDN?** follows.

Example 9-4: Reading HP 3245A Identity (IDNQR9)

This program demonstrates the use of the **IDN?** command to determine the HP 3245A model number and firmware revision code.

```
10  !file IDNQR9
20  !
30  OUTPUT 709;"IDN?"     !Read HP 3245A identity
40  FOR I = 1 TO 4        !Set loop counter
50    ENTER 709;A$        !Enter one string at a time
60    PRINT A$            !Print one string
70  NEXT I                !Increment counter
80  END
```

A typical return is.

```
HEWLETT PACKARD    (Manufacturer's Name)
3245               (Model Number)
0                  (Serial Number - Always 0)
2830               (Firmware Revision Code)
```

Channel Type Query (CTYPE?)

The **CTYPE?** *channel* command returns either "21" or "0". **CTYPE? 0 or CTYPE? CHANA** always returns "21". **CTYPE 100** or **CTYPE CHANB** returns "21" for a two-channel instrument or returns "0" for a single-channel (channel A) instrument.

Firmware Revision Query (REV?)

The **REV?** command returns the HP 3245A firmware revision date code. The code has the form "yyww", where "yy" is the year minus 1960, and "ww" is the week number of that year. For example, 2833 means that the latest firmware revision was the 33rd week of 1988 (1988 - 1960 = 28). Refer to the **REV?** command in the HP 3245A Command Reference Manual for an example program using **REV?**.

Timing Functions

This section describes HP 3245A timing functions, including setting/reading the internal clock and waiting for a specified time/event.

Setting Clock (SET TIME/ TIME)

The HP 3245A has an internal clock which can be set with the **SET TIME** command. The time (in seconds since midnight) can be read with the **TIME** command. **SET TIME** *seconds* sets the HP 3245A internal clock in number of seconds since midnight. Use the **TIME** command to return the current clock setting in seconds past midnight.

The **SET TIME** *seconds* parameter is the number of seconds since midnight. The valid range is 0 through 86399.9 (resolution = 0.01 seconds). At power-on, the HP 3245A internal clock is set to 0.0 seconds. The time set by **SET TIME** is stored in volatile memory and is lost when power is removed. You can calculate a specific time in seconds past midnight by using Time = (hours*3600) + (minutes*60) + seconds. For example, 9:45:30 A.M. = (9*3600) + (45*60) + 30 = 35130 seconds.

HP 9000 Series 200/300 controllers use two commands to convert time formats: **TIME** and **TIME$**. The **TIME** command converts a formatted time-of-day string (hh:mm:ss) into a numeric value of seconds since midnight. The **TIME$** command converts the number of seconds value into a string representing the time-of-day format. Therefore, the number of seconds past midnight can also be specified by: `OUTPUT 709;"SET TIME";TIME ("hh:mm:ss")`. For example, to specify 9:45:30 A.M. as the time to be set use `OUTPUT 709;"SET TIME";TIME(9:45:30)`. An example to set the HP 3245A clock follows.

Example 9-5: Setting HP 3245A Clock (STIME9)

This program sets the HP 3245A internal clock to 25000 seconds since midnight (06:56:40 hours) and verifies that the clock is running by reading the time five seconds after the clock is set.

```
10   !file STIME9
20   !This program demonstrates the SET TIME command.
30   !
40   CLEAR 709                              !Clear HP 3245A
50   OUTPUT 709;"RST"                       !Reset HP 3245A
60   OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
70   OUTPUT 709;"SET TIME";TIME("6:56:40")  !Set clock for 6:56:40 A.M.
80   WAIT 5                                 !Wait five seconds
90   OUTPUT 709;"TIME"                      !Read clock time
100  ENTER 709;A                            !Enter time
110  PRINT TIME$(A)                         !Convert time and display
120  END
```

A typical return is:

```
06:56:45
```

Wait For Time/Event (WAIT/WAITFOR)

The HP 3245A can be set to wait for a specified number of second with the **WAIT** *seconds* command or can be set to wait for a specified trigger (edge) from a specified source with the **WAITFOR** *source, edge* command.

Wait For Specified Time (WAIT)

With **WAIT** *seconds*, the HP 3245A waits the specified number of seconds before continuing. The valid range is 0 through 21,474,836 seconds (about 248 days). Power-on *seconds* = 0. Resolution for the **WAIT** command is 1 msec for values $\leq$30 msec and 10 msec for values >30 msec. For example, `OUTPUT 709;"WAIT 2E-3"` pauses the HP 3245A for 2 msec.

Wait For Trigger (WAITFOR)

The HP 3245A can be programmed to wait for a specified trigger (edge) on backplane trigger bus TB0 or TB1 by using the **WAITFOR** *source, edge* command. The *source* parameters are **TB0** (wait for edge on trigger bus TB0) and **TB1** (wait for edge on trigger bus TB1). The *edge* parameters are **HL** (high-to-low edge) and **LH** (low-to-high edge). To consistently detect the transitions, each level must be maintained for at least 100 µsec. For example, `OUTPUT 709;"WAITFOR TB0,HL"` causes the program statement to wait for a falling edge on trigger bus TB0. Or, `OUTPUT 709;"WAITFOR TB1,HL;WAITFOR TB1,LH"` causes the program statement to wait for a high-to-low-to-high pulse (>130 µsec wide) on trigger bus TB1.

Storing/Recalling States

This section shows how to store and recall channel and **DRIVETBn** states, including storing/recalling local states, storing/recalling remote states, and transferring states from channels.

Local Storage/Recall (SSTATE/RSTATE)

The hardware state of a channel (such as DCV output, the arbitrary waveform contents, frequency = 1 kHz, DC offset = 2 VDC, etc.) can be stored in continuous (nonvolatile) memory with the **SSTATE** or **SSTATEn** command (**DRIVETBn** is stored by **SSTATE** only). The stored state of the channel can then be recalled (and the state of the channel restored) with the **RSTATE** or **RSTATEn** command (**DRIVETBn** is recalled by **RSTATE** only). Since the states stored in continuous memory are not destroyed when power is removed, you can do a single **RSTATE** command to restore the channel to the state existing before power-down, rather than having to re-enter all the commands required to set the channel parameters.

NOTE

If a power failure occurs while a state is being stored in continuous memory, that state becomes invalid. When power is restored, the invalid partial state is purged. However, states stored in continuous memory are not purged with a **SCRATCH** *command nor are stored states lost at power-down.*

Single-Channel Instrument Operation

Figure 9-1 summarizes state storage and recall operation for a single-channel (channel A) HP 3245A. For a single-channel instrument, the current state of channel A is initially stored in volatile memory. Use **SSTATE** *number* or **SSTATEA** *number* to store the state of channel A into the state number specified by *number*. For example, **SSTATE 3** or **SSTATEA 3** stores the existing state of channel A into state #3.

For a single-channel instrument, you can store up to 14 states (0 through 13) in continuous memory using **SSTATE** or **SSTATEA** *number*. When a state is stored, use **RSTATE** *number* or **RSTATEA** *number* to recall the state from continuous to volatile memory and thus restore the channel state. For example, **RSTATE 2** or **RSTATEA 2** transfers the state stored in continuous memory state #2 into volatile memory.

Example 9-6: Store/Recall State (SSTAT9)

This program sets channel A to output a 2.0 VDC signal, stores the channel A state in state #1 in continuous memory (with **SSTATE 1**), then pauses and displays "Cycle Power" on the front panel. When the program pauses, turn the HP 3245A power OFF, then ON, and then press the controller Continue (or equivalent) key to complete the program.

When the program resumes, **RSTATE 1** recalls the channel A state from continuous memory and thus restores the channel state existing before power was cycled. The channel A output (2.0 VDC) is displayed to verify that the state before power-down was restored. (Note that **SCRATCH** clears volatile memory but does not clear continuous memory.)

```
10   !file SSTAT9
20   !This program demonstrates the SSTATE and RSTATE commands.
30   !
40   CLEAR 709                           !Clear HP 3245A
50   OUTPUT 709;"RST 0"                  !Reset channel A
60   OUTPUT 709;"SCRATCH"                !Clear HP 3245A memory
70   !
80   OUTPUT 709;"APPLY DCV 2.0"          !Output 2.0 VDC from ch A
90   OUTPUT 709;"SSTATE 1"               !Store ch A state in state 1
100  WAIT 1                              !Wait 1 second
110  OUTPUT 709;"DISP 'Cycle Power'"     !Display message
120  PAUSE                               !Pause program
130  !
140  OUTPUT 709;"RSTATE 1"               !Recall state 1
150  OUTPUT 709;"OUTPUT?"                !Query ch A output
160  ENTER 709;A                         !Enter output
170  PRINT "Channel A output =";A;"VDC"  !Display output
180  END
```

A typical return is:

```
Channel A output = 2 VDC
```

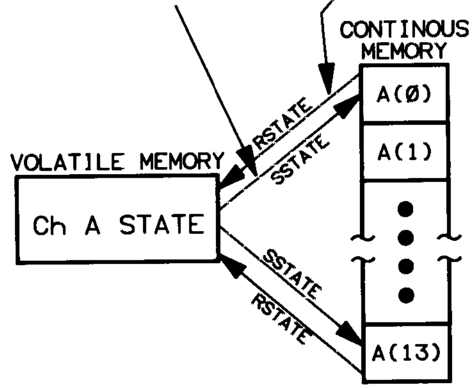
Store States (SSTATE/SSTATEA)

For a single-channel instrument, use SSTATE number or SSTATEA number to store channel A state into the continuous memory state number (0 through 13) set by number. (DRIVETBn is stored by SSTATE only and SSTATEB returns an error.)

For single-channel instrument operation, store the channel A state in memory state A(0) through A(13). States stored in continuous memory are not purged by SCRATCH or lost at power-down.

Recall States (RSTATE/RSTATEA)

Use RSTATE number or RSTATEA number to recall the state stored in continuous memory by SSTATE or SSTATEA. (DRIVETBn is recalled by RSTATE only and RSTATEB returns an error for single-channel operation.) States must be stored by SSTATE or SSTATEA before RSTATE or RSTATEA can be used.



32450P:F 9.1

Figure 9-1. Store/Recall States - Single-Channel Instrument

Two-Channel Instrument Operation

Figure 9-2 summarizes state storage/recall for a two-channel (channels A and B) HP 3245A. With two-channel operation, 7 states (0 through 6) are specified for each channel. That is, state 0 stores information in state A(0) and B(0), state 1 in A(1) and B(1), etc.

When **SSTATE** *number* is specified, the state of channel A is stored in A(n) and the state of channel B in B(n). If **SSTATEA** is used, only the state of channel A is stored. If **SSTATEB** is used, only the state of channel B is stored. For example, **SSTATE 1** stores the state of channels A and B in continuous memory in state 1 [A(1) and B(1)]. However, **SSTATEA 1** stores only the state of channel A in state 1 [A(1)].

Similarly, **RSTATE** *number* recalls the states of channels A and B from the specified continuous memory state, **RSTATEA** *number* recalls the state of channel A, and **RSTATEB** recalls the state of channel B.

In addition, for two-channel instrument operation, you can use the **STOREATOB** command to copy the state of channel A into channel B or use **STOREBTOA** to copy the state of channel B into channel A (both previous states are still stored in continuous memory). Then, you can use **SSTATEA** or **SSTATEB** to transfer the state to continuous memory.

Note that the driving channel and/or state of the Freq Ref connector may be changed by **STOREATOB**, **STOREBTOA**, or an **RSTATE(A,B)**. If, for example, a 2-channel HP 3245A is in its power-on/reset state of channel A:**REFOUT EXT** and channel B: **REFOUT OFF**, **STOREBTOA** will set both channels to **REFOUT OFF** and they will no longer by synchronized. Or, if channel A is set to **REFOUT EXT**, **STOREATOB** will set channel B to **REFOUT EXT** (which sets channel A to **REFOUT OFF**). A similar situation will result when TB0 or TB1 is driven with **SYNCOUT** or **REFOUT**.

Example 9-7: Transfer Channel States (TSTAT9)

This program initially sets channel A for 2.0 VDC output and channel B for 0 VDC output. Next, **STOREATOB** is used to transfer the state of channel A to channel B. Then, **SSTATE 1** is used to store the states of channels A and B into state 1 in continuous memory. After power is cycled, the states of channels A and B are recalled from continuous memory and displayed to show that the channel states are identical (2.0 VDC output).

NOTE

When the front panel display indicates "Cycle Power", press the ON/OFF switch OFF then ON and then press the controller Continue (or equivalent) key to continue the program

---

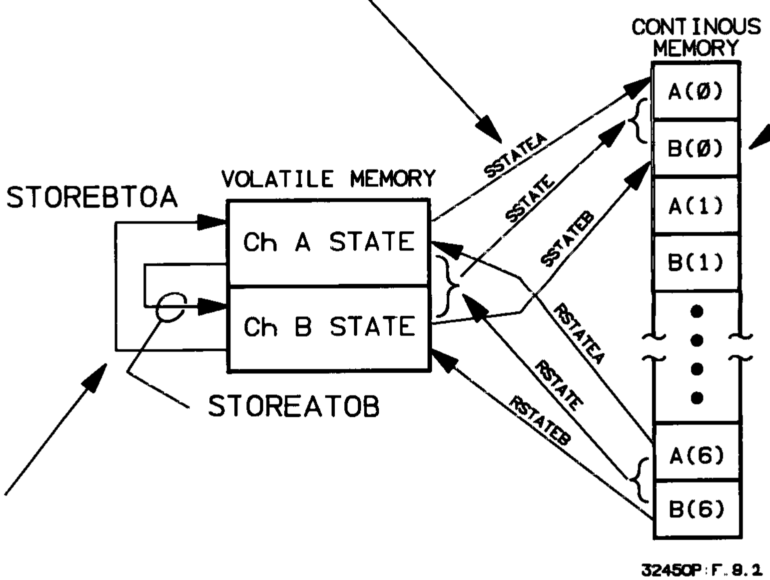
Store States (SSTATE/SSTATEA/SSTATEB)

For a two-channel instrument, use SSTATE number to store the states of channels A and B into the continuous memory state (0 through 6) set by number. Use SSTATEA number to store channel A state only or use SSTATEB number to store channel B state only (DRIVETBn is stored by SSTATE only.)

For two-channel operation, store channel A state in A(0) through A(6) and store channel B state in B(0) through B(6). States stored in continuous memory are not purged by SCRATCH or lost at power-down

Recall States (RSTATE/RSTATEA/RSTATEB)

Use RSTATE number recall the state of both channel A and B as stored in continuous memory by SSTATE, SSTATEA, or SSTATEB. Use RSTATEA number to recall channel A state or RSTATEB number to recall channel B state. (DRIVETBn is recalled by RSTATE only.) States must be stored by SSTATE, SSTATEA, or SSTATEB before RSTATE, RSTATEA, or RSTATEB can be used



Transfer Channel States (STOREATOB/STOREBTOA)

Use STOREATOB to transfer (copy) the state of channel A to channel B. Use STOREBTOA to transfer (copy) the state of channel B to channel A. After STOREATOB or STOREBTOA the states remain in volatile memory

Use SSTATE, SSTATEA, or SSTATEB to store states in continuous memory. Executing STOREATOB, STOREBTOA, or RSTATE (A,B) may change the driving channel and/or state of the Freq Ref connector and lose channel synchronization.

Figure 9-2. Store/Recall States - Two-Channel Instrument

```
10   !file TSTAT9
20   !This program demonstrates the SSTATE, STOREATOB, and
30   !RSTATE commands.
40   !
50   CLEAR 709                             !Clear HP 3245A
60   OUTPUT 709;"RST"                      !Reset HP 3245A
70   OUTPUT 709;"SCRATCH"                  !Clear HP 3245A memory
80   !
90   OUTPUT 709;"USE 0"
100  OUTPUT 709;"APPLY DCV 2.0"            !Output 2.0 VDC from ch A
110  OUTPUT 709;"STOREATOB"                !Copy ch A state to ch B
120  OUTPUT 709;"SSTATE 1"                 !Ch A/B states to state 1
130  WAIT 1                                !Wait 1 second
140  OUTPUT 709;"DISP 'Cycle Power'"       !Display message
150  PAUSE                                 !Pause program
160  !
170  OUTPUT 709;"RSTATE 1"                 !Recall ch A and B states
180  OUTPUT 709;"USE 0"                    !Use channel A
190  OUTPUT 709;"OUTPUT?"                  !Query channel A output
200  ENTER 709;A                           !Enter channel A output
210  PRINT "Channel A output =";A;"VDC"    !Display channel A output
220  !
230  OUTPUT 709;"USE 100"                  !Use channel B
240  OUTPUT 709;"OUTPUT?"                  !Query channel B output
250  ENTER 709;B                           !Enter channel B output
260  PRINT "Channel B output =";B;"VDC"    !Display channel B output
270  END
```

A typical return is:

```
Channel A output = 2 VDC
Channel B output = 2 VDC
```

Remote Storage/Recall (SET/SET?)

As an added feature, you can store the state of the HP 3245A in your controller using the **SET?** command. Then, if power is lost, you can restore the state of the entire HP 3245A to that existing when **SET?** was executed by using the **SET** command

Or, if you want to store the state of an HP 3245A into another HP 3245A, you can store the state in the controller with **SET?** and transfer the state to the second HP 3245A with **SET**. Note that **SET** and **SET?** transfer the state of the HP 3245A, not individual channel states.

SET/SET? Operation

SET *block_data* instructs the HP 3245A to accept a binary block of data from the controller which specifies the HP 3245A configuration. Before executing **SET**, **SET?** must be executed to read the current HP 3245A configuration into the controller. The *block_data* parameter is a block in IEEE-728 Block A format which specifies the configuration. The contents of the block must be identical to the block returned by the **SET?** command.

Block A Format

The **SET** *block_data* parameter is in IEEE-728 Block A format. The block A format header consists of four bytes: the # sign, the letter A, and two bytes which indicate the number of bytes of data to follow. When the HP 3245A configuration is transferred to the controller (with **SET?**) and is then returned to the HP 3245A (with **SET**), the Block A format is automatically provided. Refer to the **SET** command in the HP 3245A Command Reference Manual for a program using **SET** and **SET?**.

Memory Management

This section gives guidelines to use HP 3245A memory efficiently, including HP 3245A task priorities, memory storage, and using memory efficiently.

Task Priorities

Three independent tasks affect command execution in the HP 3245A. In priority order, HP 3245A tasks are:

- 1: Commands executed from the front panel keyboard.
- 2: Commands executed from the controller via HP-IB.
- 3: Subroutines executed using the **RUN** command.

Table 9-3 shows how the HP 3245A executes its defined tasks (commands or subroutines).

Table 9-3. HP 3245A Task Execution Rules

. Any task can execute only one command or subroutine at a time.
. The HP 3245A always executes the highest priority task which has something to do.
. Any task which executes a wait command >30 msec allows the next-lower priority task to execute some or all of its command(s).
. If a task requires a resource (such as a trigger bus) which another task is using, the second task waits until the first task finishes with the resource, even if the second task is higher priority.

Memory Storage (MEMAVAIL?)

The HP 3245A has approximately 80 Kbytes of volatile memory. An additional 64 Kbytes of continuous (battery-backed) memory is reserved for the 14 store states, redefining front panel keys, etc. Volatile memory available depends on the number of variables, arrays, and subroutines stored.

Volatile/Continuous Memory Storage

Volatile memory is the largest block of HP 3245A user memory. Since volatile memory is not battery-backed, any information stored in volatile memory is lost when power is removed from the HP 3245A. Volatile memory is used to store arrays, variables, and subroutines. Continuous memory is used to store the 14 store states and other information such as power-on SRQ setting and front panel display status. Table 9-4 lists the information stored in continuous memory.

NOTE

The contents of continuous memory can be lost if power fails while the HP 3245A is executing a command which stores data in continuous memory. Therefore, all stored states should also be stored in the controller as a back-up.

Table 9-4. Data Stored in Continuous Memory

Item	Description
Power-On SRQ	Power-on SRQ setting. When enabled, RQS mask is set to "8" at power-on thus allowing SRQ to be asserted.
HP-IB Address	Current HP 3245A HP-IB address. Factory setting is 09.
Recall Key Buffer	Recall last command(s) entered.
Beep Mode	Enable/disable state of beeper mode. When enabled, a beep occurs when an error occurs.
Recall States	States recalled with Recall key are stored.
Front Panel Key States	Current definition of each front panel key, including keys redefined with DEFKEY.
Stored States	States stored by SSTATE, SSTATEA, or SSTATEB.
Security Code	Calibration security code assigned. Factory assignment is 3245.

Returning Memory Size (MEMAVAIL?)

MEMAVAIL? returns the size (in bytes) of the largest volatile memory blocks available in HP 3245A memory. The command returns the largest block (number of bytes) of volatile memory. Refer to the **MEMAVAIL?** command in the HP 3245A Command Reference Manual for an example program using **MEMAVAIL?**

Using Memory Efficiently

When an item is purged from volatile memory, a "hole" remains where that data was stored since the HP 3245A does not pack the remaining data. (However, two adjacent holes are combined into one large hole.) Thus, volatile memory can become fragmented by leaving holes in the memory.

The HP 3245A will not allocate space for an item (such as an array or a stored state) over multiple holes - it must have one continuous block. (Use **MEMAVAIL?** to return the size of the largest hole in volatile memory.) Table 9-5 lists some guidelines to help avoid memory fragmentation and thus ensure maximum amount of available volatile memory.

Table 9-5. Volatile Memory Management Guidelines

- Define the most permanent items first (such as variable names, subroutine names, etc.). Define items that are subject to change last (such as arrays that will be redimensioned and temporary subroutines).
- Purge items in the opposite order that they are created, since two adjacent holes are combined into one large hole.
- Download and compress each debugged subroutine individually rather than downloading and compressing several subroutines at a time. If a subroutine has more than 100 lines, consider dividing it into two or three smaller subroutines.
- Use INTEGER arrays wherever possible, since they use less memory than REAL arrays.

Test/Calibration

You can use the **TEST**, **FTEST**, or **DTEST** command to test the HP 3245A operation. Or you can use the **CAL** command for HP 3245A calibration.

Test Procedures (TEST/FTEST/DTEST)

The **TEST** command performs a self-test on the HP 3245A. When the self-test passes, PASS is displayed on the front panel display. To perform a data test (**DTEST** command), execute **DTEST 0** to test channel A or execute **DTEST 100** to test channel B. When the device test passes, 0.0000000E+00 is displayed on the front panel display

To perform a full test on the HP 3245A, use the **FTEST** command. To test the front panel Output, connect a BNC cable between the front panel Output connector and the Trigger (I/O) connector on the front panel and then execute **FTEST** *ch*. When the full test successfully completes, PASS is shown on the front panel display.

For example, to perform a full test on channel A, connect a BNC cable between the front panel Channel A Output connector and the channel A Trigger (I/O) connector and execute **FTEST 0**. For channel B, make similar connections and execute **FTEST 100.**

You can also perform a full test for the rear panel Output connectors. For example, to test channel A, connect a BNC cable between the rear panel Channel A Output connector and the front panel Channel A Trigger (I/O) connector. Then, execute **FTEST 1.** For channel B, make similar connections and execute **FTEST 101.**

Calibration (CAL)

As required, you can perform a full calibration of the HP 3245A, including operational verification, performance tests, and adjustments. Refer to the HP 3245A Calibration Manual for details.

PART IV
PROGRAMMING
OUTPUTS/WAVEFORMS

Contents

Chapter 10
Programming DC Outputs

Chapter 10
Programming DC Outputs

Introduction

This chapter shows how to program the HP 3245A for DC voltage (DCV) and DC current (DCI) outputs. Refer to the following chapters for information to generate AC waveforms and other unique signals.

- Chapter 11 - Generating Defined Waveforms.
- Chapter 12 - Generating Arbitrary Waveforms.
- Chapter 13 - Triggering Outputs/Waveforms.
- Chapter 14 - Advanced Programming Topics.

NOTE

All programs in this chapter are stored on the Example Programs disc. Refer to Chapter 1 - Using This Manual for information on using the disc. Refer to the HP 3245A Command Reference Manual for command details. If you use the front panel keyboard to generate DC outputs, refer to Chapter 6 - Front Panel Operation.

Chapter Contents

The chapter contents are:

- **Programming Power-On DC Outputs** shows how to program the HP 3245A for DC voltage and current outputs when power-on conditions are set.

- **Programming Modified DC Outputs** shows how to program the HP 3245A for DC voltage and current outputs when output parameters are modified from their power-on settings.

- **Programming Triggered DC Outputs** shows how to program the HP 3245A for DC voltage and current outputs when triggering is used

DC Outputs Overview

Figure 10-1 shows the three types of DC outputs discussed in this chapter (power-on, modified, and triggered) and shows a suggested set of steps to select the output function, channel parameters, and trigger event as required. For example, for DC outputs with power-on settings, all you need to do is to select the output function (DCI or DCV) and then program the instrument for the desired output. Or, for modified DC outputs, you will need to select the output function and then select the channel parameter(s) which are to be changed from their power-on setting.

Table 10-1 shows power-on conditions for commands which apply to DC outputs. If power-on conditions can be used, refer to "Programming Power-On DC Outputs". If power-on conditions must be modified, but triggering is not required, refer to "Programming Modified DC Outputs". If triggering is required, refer to "Programming Triggered DC Outputs".

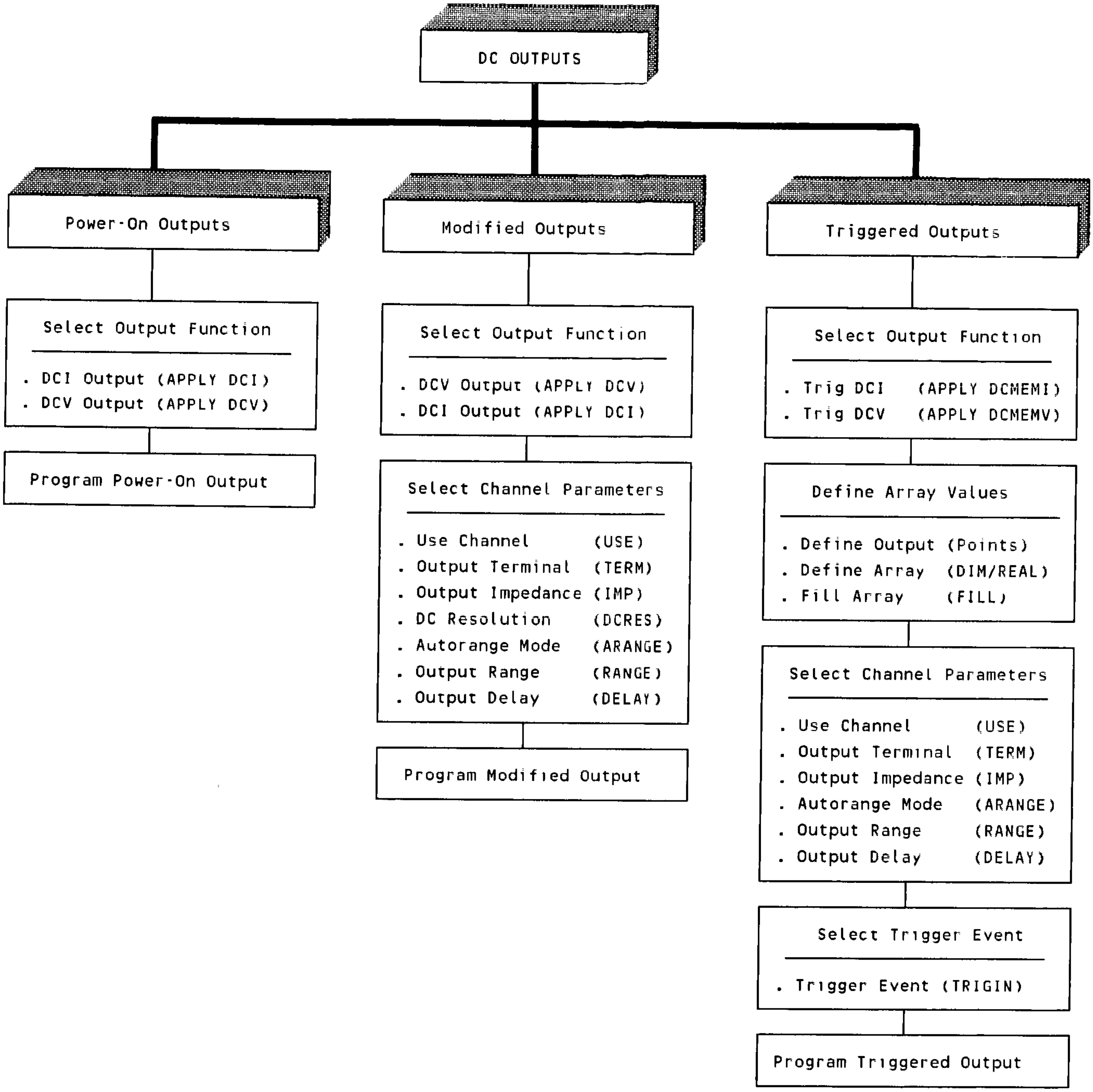


Figure 10-1. DC Outputs - Overview

Table 10-1. DC Outputs - Power-On State

Output Parameter	Command	Power-On State
Use Channel	USE	Channel A
Output Terminal	TERM	Front panel
Output Impedance	IMP	0 Ohms
DC Resolution Mode	DCRES	High-res mode
Autorange Mode	ARANGE	Enabled
Range	RANGE	1V (autorange)
Delay Time	DELAY	0.04 seconds

Programming Power-On DC Outputs

This section show how to generate DC voltage (DCV) or DC current (DCI) outputs when power-on conditions are set. As shown in Figure 10-2, for power-on DC outputs you only need to select the output function (DCV or DCI) required and then program the HP 3245A for the desired output with an **APPLY DCV** or **APPLY DCI** command.

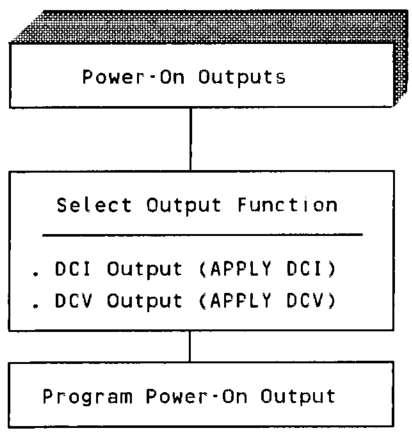


Figure 10-2. Power-On DC Outputs - Selection Steps

Command Summary

Figure 10-3 summarizes channel operation and related commands for power-on DC outputs. The output function (DCV or DCI) is set with an **APPLY DCV** *volts* or an **APPLY DCI** *amps* command With power-on conditions, the output is from channel A (set with **USE 0**), while **TERM FRONT** sets the front panel Channel A Output terminal as the output port.

The channel A output impedance is set to 0 Ω with **IMP 0**, the channel is set for high-resolution mode with **DCRES HIGH**, and autorange is set with **RANGE AUTO**. The delay time between commands is set to 0.04 seconds with **DELAY 0.04**.

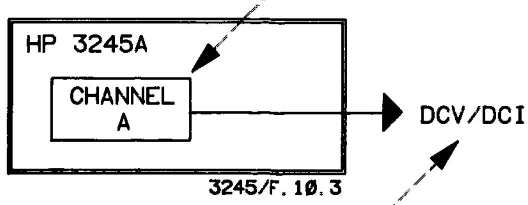
Selecting Output Function

As noted, the only step required for power-on DC outputs is to select the output function with an **APPLY DCV** or **APPLY DCI** command For power-on conditions, channel A is the **USE** (output) channel. Channel A can be set for constant voltage (voltage) DC outputs with **APPLY DCV** *volts* or for constant current (current) DC outputs with **APPLY DCI** *amps*.

For power-on settings, the range for DC voltage outputs is -10.25 VDC to +10.25 VDC, while the range for DC current outputs is -0.1 A to +0.1 A. You can use the function query command (**APPLY?**) to return the function setting (DCV or DCI) or you can use the **OUTPUT?** command to return the output value. Voltage output values are returned in DC Volts, current output values in DC Amps.

Power-On Channel Settings

USE 0	Channel A is USE channel.
TERM FRONT	Front panel output.
IMP 0	0 Ω output impedance.
DCRES HIGH	High-res (6 digits) mode.
ARANGE ON	Autorange function enabled
RANGE AUTO	Autorange set.
DELAY 0 04	Output delay is 0.04 seconds.



Power-On Output Functions

Set DC voltage output with APPLY DCV volts (-10.25 VDC to + 10.25 VDC).
Set DC current output with APPLY DCI amps (-0.1 A to +0.1 A).

Figure 10-3. Power-On DC Outputs - Command Summary

Programming Examples

When you have selected the output function and output level, program the HP 3245A for the desired output. Figure 10-4 shows the suggested programing sequence. Example programs follow to output DC voltages and currents with power-on conditions.

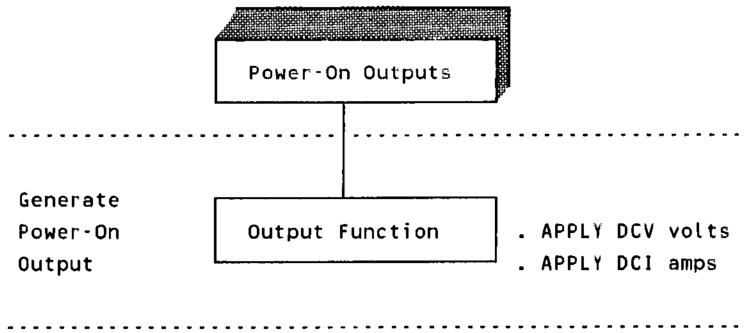


Figure 10-4. Power-On DC Outputs - Programming Sequence

Example 10-1: Power-On DC Voltage Output (DCVPO10)

This program outputs 3.5 VDC from channel A and returns the output function to the controller. **RST** places the HP 3245A in its power-on state with the DC voltage function selected, autorange enabled, high-resolution mode selected, output impedance set to 0 Ω, and front panel Channel A Output connector selected.

```
10   !file DCVPO10
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   OUTPUT 709;"APPLY DCV 3.5"     !Output DC voltage @3.5 VDC
70   OUTPUT 709;"APPLY?"            !Query output function
80   ENTER 709;A$                   !Enter output value
90   PRINT "Function =";A$          !Display output value
100  END
```

A typical return is:

```
Function = DCV
```

Example 10-2: Power-On DC Current Output (DCIPO10)

This program outputs 3.25 mA from channel A and returns the output value to the controller. **RST** places the HP 3245A in its power-on state with the DC voltage function selected, autorange enabled, high-resolution mode selected, output impedance set to 0 Ω, and front panel Channel A Output connector selected.

```
10   !file DCIPO10
20   !
30   CLEAR 709                              !Clear interface
40   OUTPUT 709;"RST"                       !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                   !Delete vars, arrays, subs
60   OUTPUT 709;"APPLY DCI 3.25E-3"         !Output DC current @ 3.25 mA
70   OUTPUT 709;"OUTPUT?"                   !Query output value
80   ENTER 709;A                            !Enter output value
90   PRINT "Output =";A;"Amps"              !Display output value
100  END
```

A typical return is:

```
Output = .00325 Amps
```

Programming Modified DC Outputs

This section shows how to program the HP 3245A for DC outputs with parameter(s) modified from their power-on settings. As shown in Figure 10-5, there are two steps to select functions and parameters for modified DC outputs: first, select output function (DCV or DCI), and then select channel parameter(s) to be modified.

Command Summary

Figure 10-6 summarizes channel operation and related commands for modified DC outputs. **USE** sets the output channel (A or B) and **TERM** sets the output terminal (front or rear). **APPLY DCV** sets DCV output, while **APPLY DCI** sets DCI output. **IMP** sets 0 Ω or 50 Ω output impedance, **DCRES** sets high-resolution (6-digits) mode or low-resolution (3-digits) mode, and **RANGE/ARANGE** can be used to set autorange or fixed range. **DELAY** sets the delay time (from 0 to 9.99 seconds) between output commands.

Selecting Output Function

The first step to program modified DC outputs is to select the output function (DCV or DCI). **APPLY DCV** *volts* sets DCV on the **USE** channel, while **APPLY DCI** *amps* sets DCI on the **USE** channel. The range for DC voltage outputs is -10.25 VDC to +10.25 VDC for **IMP 0** or -5.125 VDC to +5.125 VDC with **IMP 50** (and the output is terminated in a 50 Ω load). The range for DC current outputs is -0.1 A to +0.1 A for either impedance setting.

Use **APPLY?** to return the output function (DCV or DCI) or use **OUTPUT?** to return the output value. Voltage output values are returned in Volts, current output values in Amps. The power-on/reset setting is **APPLY DCV 0** (Autorange mode is enabled when the output function is changed.)

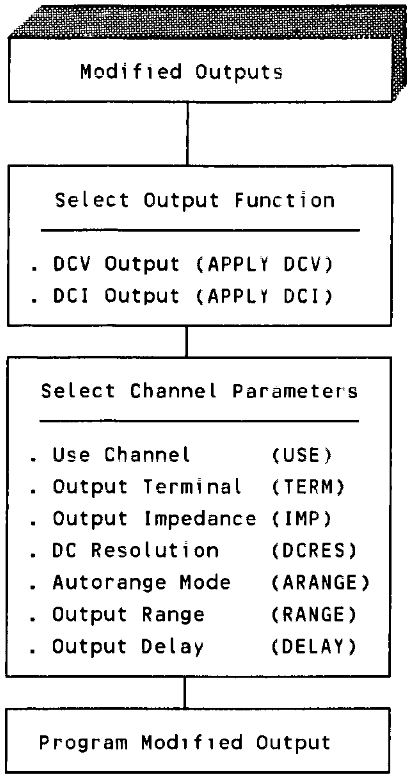


Figure 10-5. Modified DC Outputs - Selection Steps

Selecting Channel Parameters

When you have selected the output function, select the channel parameters required. As shown in Figure 10-5, there are seven commands which can be used to modify DC outputs: **USE**, **TERM**, **IMP**, **DCRES** (does not apply to triggered DC outputs), **ARANGE**, **RANGE**, and **DELAY**.

Use Channel (USE)

At power-on, the **USE** channel (the channel to receive output commands) is set to channel A. If you have a two-channel version of an HP 3245A, you can use **USE 0** or **USE CHANA** to select channel A or use **USE 100** or **USE CHANB** to select channel B. **USE?** returns the current **USE** channel (channel A or B). **USE?** returns "0" if channel A is the **USE** channel or returns "100" if channel B is the **USE** channel.

NOTE

You can simultaneously have two **USE** *channels - one for commands entered from the front panel and the other for commands entered over HP-IB (e.g.,* **USE 0** *over HP-IB and* **USE 100** *entered from the front panel).*

Output Terminal (TERM)

Select the output terminal for the **USE** channel with the **TERM** [*mode*] command. **TERM FRONT** sets the front panel Output terminal and **TERM REAR** sets the rear panel Output terminal for the **USE** channel. **TERM OFF** or **TERM OPEN** disconnects all outputs. The power-on/reset setting is **TERM FRONT**. **TERM?** returns the current output terminal setting for the **USE** channel. The output must be from the front panel or from the rear panel. The signal cannot be output simultaneously from both output ports.

Use Channel (USE)

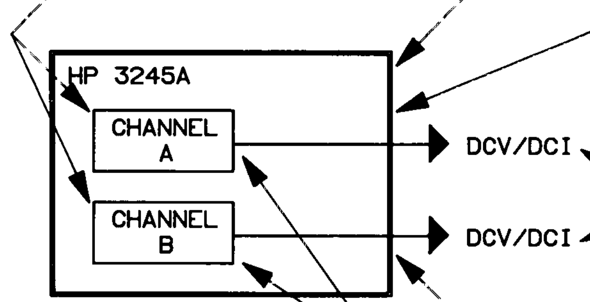
USE ch sets use (output) channel. USE 0 or USE CHANA sets channel A, USE 100 or USE CHANB sets channel B.

Output Terminal (TERM)

TERM [mode] sets the output terminal. TERM FRONT sets front panel output, TERM REAR sets rear panel output, TERM OFF/OPEN disconnects outputs

Output Impedance (IMP)

IMP mode sets the output impedance on the use channel. IMP 0 sets 0Ω impedance, IMP 50 sets 50Ω impedance.



Output Function (APPLY DCV/DCI)

APPLY DCV volts sets DC voltage output (-10.25 VDC to +10.25 VDC). APPLY DCI amps sets DC current output (-0.1 A to +0.1 A).

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Refer to Table 10-4 for default delay times.

DC Range (RANGE)

RANGE [max_output] sets fixed output range. DC voltage and current output resolution depends on the output impedance (set by IMP). RANGE AUTO sets autorange.

DC Resolution (DCRES)

DCRES mode sets high-resolution or low-resolution output mode on the use channel. DCRES HIGH sets high-resolution (6 digits) mode, DCRES LOW sets low-resolution (3 digits) mode.

Figure 10-6. Modified DC Outputs - Command Summary

NOTE

EXCESSIVE TERMINAL VOLTAGE ERROR. If an external signal is applied to the selected Output terminal and is large enough to damage the instrument (>± 15 Vpeak), the HP 3245A automatically opens the appropriate output relay. When an output relay opens, the "EXCESSIVE TERMINAL VOLTAGE" error is generated. To continue operation, remove the external signal and reset the HP 3245A.

Output Impedance (IMP)

For DC (and AC) voltage outputs, use the **IMP** *mode* command to set the output impedance on the **USE** channel (**IMP** does not apply to current outputs). **IMP 0** selects the 0 Ω mode, while **IMP 50** selects the 50 Ω mode. The power-on/reset setting is **IMP 0**. The **IMP?** command returns the impedance setting on the **USE** channel.

In 0 Ω mode, the **USE** channel voltage output equals the programmed output voltage with or without load termination. In 50 Ω mode, actual output voltage equals the programmed output voltage ONLY when the channel is terminated with a 50 Ω load. For example, if the output is programmed for 2 VDC in the 50 Ω mode, the output will double to 4 VDC with no load. The voltage will be 2 VDC when the output is terminated with a 50 Ω load.

If the present DC output voltage is generated in the 0 Ω mode and is greater than the maximum allowable output in the 50 Ω mode, selecting the 50 Ω mode will generate an "OUT OF RANGE" error. When the output impedance is changed from 0 Ω to 50 Ω, or vice-versa, autorange mode is automatically enabled.

DC Resolution (DCRES)

For DC voltage or current outputs, use **DCRES** *mode* to set output resolution from the **USE** channel. **DCRES HIGH** sets high-resolution mode, while **DCRES LOW** sets low-resolution mode. The power-on/reset setting is **DCRES HIGH**. **DCRES?** returns the setting (HIGH or LOW) on the **USE** channel. When the DC resolution is change from **HIGH** to **LOW**, or vice-versa, autorange is automatically enabled. Table 10-2 shows DC resolution modes.

Resolution Affected by Other Settings

For DC voltage outputs, actual resolution depends on the output impedance, on the DC resolution mode, and on the DC range. For DC current outputs, resolution depends only on the DC resolution mode and the range settings.

Use High-Resolution Mode for Greatest Accuracy

For greatest accuracy, use the high-resolution mode which provides 6.5 digits of resolution. For increased settling time with lower (3.5 digit) resolution, use the low-resolution mode.

Table 10-2. DC Outputs - Resolution Modes

mode	Description
HIGH	. High-resolution mode . 40 msec settling time (@ 0.0001% resolution) . 6.5 digits (24 bits) of resolution. . 2 DC voltage ranges . 4 DC current ranges
LOW	. Low-resolution mode. . 100 µsec settling time (@ 0.05% resolution) . 3.5 digits of resolution. . 7 DC voltage ranges . 4 DC current ranges

Autorange Mode (ARANGE)

The autorange mode is enabled or disabled with the **ARANGE** command. With autorange mode enabled (**ARANGE ON**), the HP 3245A automatically selects the lowest voltage or current range which will provide maximum accuracy for the desired output level. An error is generated if you attempt to set the output level greater than the maximum output of the highest voltage or current range available

With autorange mode disabled (**ARANGE OFF**), the HP 3245A retains its present range regardless of the programmed output value. If the programmed value attempts to exceed the maximum output of the present range, an error is generated. Autorange mode is enabled at power-on. Also, autoranging is automatically enabled when commands which change the DC resolution mode (**DCRES** command), output impedance (**IMP** command), or output function (**APPLY** commands) are executed.

DC Range (RANGE)

At power-on/reset, autorange is enabled (**ARANGE ON**) When autorange is enabled, the lowest voltage or current range is automatically selected which will provide maximum accuracy for the desired output Use **RANGE** [*max_output*] to select a fixed output range. **RANGE?** returns the current range setting on the **USE** channel.

DC Voltage/Current Ranges

Table 10-3 shows DC voltage outputs for high-resolution and low-resolution modes (set with **DCRES**) and DC current outputs for both high-resolution and low-resolution modes. In the table, "Maximum Programmed Output" is the maximum value you can use in the **APPLY DCV** *pp_ampl* or **APPLY DCI** *pp_ampl* for the range, resolution, and output impedance selected.

Selecting a Range

To select a voltage or current range, select **RANGE** [*max_output*] as the maximum expected output voltage (in Volts) or current (in Amps) The channel then selects the correct range, based on the previous mode (DCI, DCV, ACI, or ACV) of the **USE** channel. To enable autorange, use **RANGE AUTO** or default the [*max_output*] parameter.

Table 10-3. DC Outputs - Output Ranges

DC Current Outputs			
Range (Amps DC)	Maximum Programmed Output	Resolution High-Res Mode	Resolution Low-Res Mode
0.0001A	0.0001 A	0.1 nA	50 nA
0.001A	0.001 A	1 nA	500 nA
0.01A	0.01 A	10 nA	5 µA
0.1A	0.1 A	100 nA	50 µA

DC Voltage Outputs (High-Resolution Mode)			
Output Impedance (Ohms)	Range (VDC)	Maximum Programmed Output	Resolution
0	1V	1.25 V	1 µV
0	10V	10.25 V	10 µV
50	0.5V	0.625 V	0.5 µV
50	5V	5.125 V	5 µV

DC Voltage Outputs (Low-Resolution Mode)			
Output Impedance (Ohms)	Range (VDC)	Maximum Programmed Output	Resolution
0	0.15625V	0.15625 V	78 µV
0	0.3125V	0.3125 V	156 µV
0	0.625V	0.625 V	312 µV
0	1.25V	1.25 V	625 mV
0	2.5V	2.5 V	1.25 mV
0	5V	5 V	2.5 mV
0	10V	10 V	5 mV
50	0.078125V	0.078125 V	39 µV
50	0.15625V	0.15625 V	78 µV
50	0.3125V	0.3125 V	156 µV
50	0.625V	0.625 V	312 µV
50	1.25V	1.25 V	625 mV
50	2.5V	2.5 V	1.25 mV
50	5V	5 V	2.5 mV

Program RANGE Command Last

Since the output range values differ based on the DC resolution mode, output impedance, and output function selected, the HP 3245A automatically enables the autorange function when any of these parameters are changed. Therefore, when selecting fixed ranges with **RANGE**, program the **RANGE** command AFTER the DC resolution mode, output impedance, and output function have been chosen. To avoid a potential "OUT OF RANGE" error, do not execute **RANGE** until you have ensured that the desired output level can be generated on the selected range.

Using Autorange Mode

Output range values differ according to the DC resolution mode, output impedance, and output function selected. When any of these parameters are changed, the channel automatically enables the autorange function. When autorange is enabled (**ARANGE ON**), an error is generated you attempt to set the output level greater than the maximum output of the highest voltage or current range available.

When autorange is disabled (**ARANGE OFF**), the channel retains the present range regardless of the programmed output value. If the programmed value attempts to exceed the maximum output of the present range, an error is generated. The MNRG annunciator on the front panel is ON when a fixed range is set (autorange disabled) on the channel presently being monitored and is OFF when autoranging is enabled.

Output Delay (DELAY)

As required, you can change the delay time between output commands with the **DELAY** command. **DELAY** [*time*] specifies a time interval during which outputs from the **USE** channel are allowed to settle before the next command is executed. The **DELAY?** command returns the present delay time (in seconds). If no delay time has been programmed with the **DELAY** command, **DELAY?** returns the default delay time based on the values shown in Table 10-4.

Setting Delay Time

The **DELAY** [*time*] parameter specifies the settling time (range = 0 to 999 seconds) with 0.01 second resolution. At power-on/reset, the default *time* is automatically determined by the function, range, DC resolution, and output impedance set on the **USE** channel.

Default Delays

If a delay time is not specified, the HP 3245A determines a default delay based on present function, range, DC resolution, and output impedance. If the output function, range, resolution, or output impedance are changed without specifying a new delay, the default delay time is updated based on the new selection. Table 10-4 shows default delay times for all output functions.

Changing Delay Times

Once a channel's delay time is set with **DELAY**, the value does not change until another value is specified. A delay time can be set which is shorter than the default value, but the resultant settling time may be too short to allow the channel to meet accuracy specifications for that output function.

Table 10-4. DC Outputs - Default Delay Times

Change In:	Default Delay
Output Function:	
High-res DCV/DCI (from other function)	1 sec
All other function changes	30 msec
Output Value:	
High-resolution mode	40 msec
Low-resolution mode (same range)	0
Triggered DCV/DCI (first value)	30 msec
Triggered DCV/DCI (after first value)	0
All other modes	30 msec
Range:	
High-resolution mode	40 msec
All other modes	30 msec
Output Impedance (Voltage Only):	
High-resolution mode	40 msec
All other modes	30 msec

Commands Affected by DELAY

The **DELAY** command affects commands which control the output function (**APPLY** commands); the output range (**ARANGE** and **RANGE**); DC resolution (**DCRES**); and output impedance (**IMP**). When the **USE** channel is operating in the DC triggered-sequence mode (**APPLY DCMEMI** or **APPLY DCMEMV**), only the first value in the sequence is delayed. Refer to "Programming Triggered DC Outputs" for triggered-sequence mode operation.

Programming Examples

When you have selected the output function and channel parameter settings required, program the HP 3245A for the desired output. Figure 10-7 shows a suggested sequence of commands to program the **USE** channel for modified DC outputs. Note that the programming sequence for commands is NOT the same as the sequence of steps to select the parameters shown in Figure 10-5.

Also, in the command sequence note that the **RANGE** command is programmed after the **APPLY DCV** or **APPLY DCI** command so that the correct range (based on the previous mode of the channel) is used. Example programs follow to generate modified DC outputs.

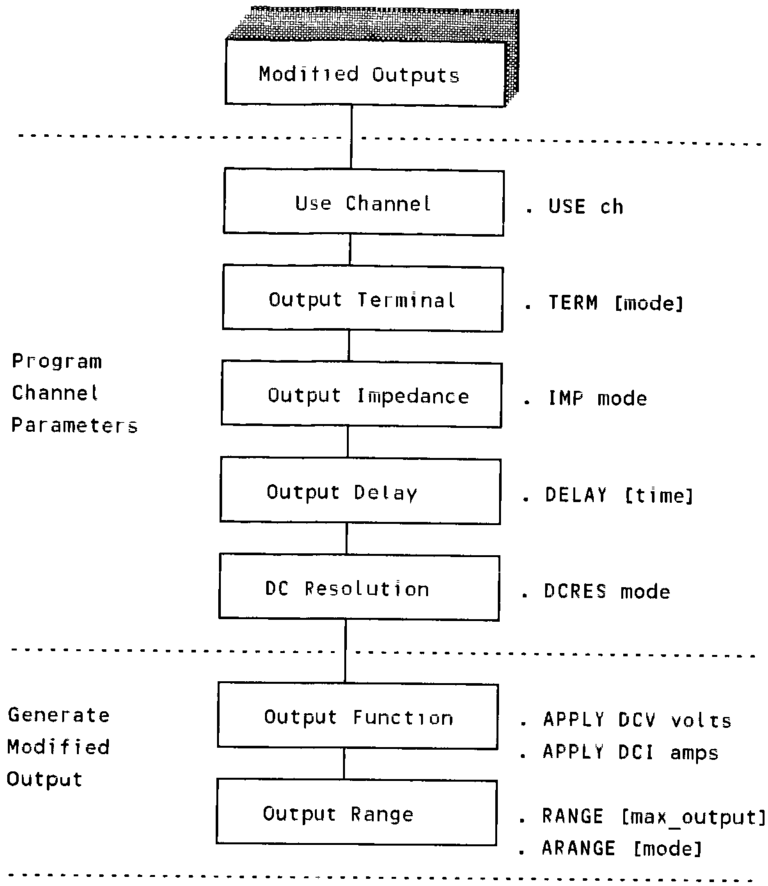


Figure 10-7. Modified DC Outputs - Programming Sequence

Example 10-3: Selecting Output Terminal (TERMF10)

This program selects the rear panel Channel A Output connector as the channel A output destination and returns the output terminal setting. (Use **TERM FRONT** to return the output to the front panel.)

```
10   !file TERMF10
20   !
30   CLEAR 709                          !Clear interface
40   OUTPUT 709;"RST"                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"               !Delete vars, arrays, subs
60   OUTPUT 709;"TERM REAR"             !Ch A output from rear terminal
70   OUTPUT 709;"TERM?"                 !Query output terminal
80   ENTER 709;A$                       !Enter output terminal
90   PRINT "Output Terminal = ";A$      !Display output terminal
100  END
```

A typical return is:

```
Output Terminal = REAR
```

Example 10-4: Selecting DC Current Range (RANGE10)

This program outputs 0.06 Amps (PP) at 100 nA resolution. Since **RST** sets DCV mode and autorange on channel A, the channel must first be set to the DCI mode with **APPLY DCI** (line 60) before **RANGE** is executed. If this is not done, the range will remain set at 1V, based on the previous mode setting of DCV.

```
10   !file RANGE10
20   !
30   CLEAR 709                    !Clear interface
40   OUTPUT 709;"RST"             !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"         !Delete vars, arrays, subs
60   OUTPUT 709;"APPLY DCI 0.06"  !Set ch A for DCI mode
70   OUTPUT 709;"RANGE .05"       !Set 0.1A range
80   END
```

Example 10-5: Setting Output Delay Time (DELAY10)

This program outputs two DC voltages with delays from channel A. Line 70 outputs 5 VDC followed by a two-second delay and then line 80 outputs 2.5 VDC.

```
10   !file DELAY10
20   !
30   CLEAR 709                    !Clear interface
40   OUTPUT 709;"RST"             !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"         !Delete vars, array, subs
60   OUTPUT 709;"DELAY 2"         !Set 2 second delay
70   OUTPUT 709;"APPLY DCV 5"     !Output 5 VDC for 2 seconds
80   OUTPUT 709;"APPLY DCV 2.5"   !Output 2.5 VDC
90   END
```

Programming Triggered DC Outputs

This section shows how to program the HP 3245A for triggered DC outputs. As shown in Figure 10-8, there are four steps to select triggered DC output parameters:

- Select output function.
- Define output array values.
- Select channel parameters.
- Select trigger event.

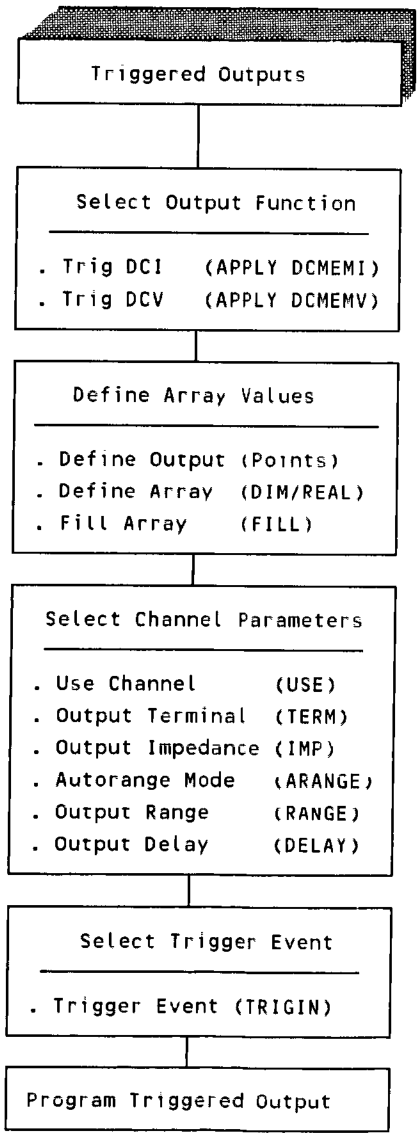


Figure 10-8. Triggered DC Outputs - Selection Steps

Command Summary

Figure 10-9 summarizes commands for triggered DC outputs. **USE** sets the **USE** channel (A or B) and **TERM** sets the output terminal (front or rear panel). **IMP** sets 0 Ω or 50 Ω output impedance, **RANGE** sets fixed range or autorange, and **DELAY** sets output delay time. Note that triggered DC outputs operate only in low-resolution (**DCRES LOW**) mode.

For triggered DC output operation, the user specifies a set of output values (from 2 to 2048) and stores the values in an array (defined by **DIM** or **REAL**) using the **FILL** command. **APPLY DCMEMV** sets DCV outputs and **APPLY DCMEMI** sets DCI outputs. When **APPLY DCMEMV** or **APPLY DCMEMI** is executed, the first value in the array is immediately output without a trigger.

Then, the next value in the array is output when a trigger event set by **TRIGIN** occurs. Each time a trigger event occurs, the next value in the array is output. After the last value in the array is output, the next trigger outputs the first value in the array (wraparound).

For example, with three DCV output values (-1.0, 0, 2.0) stored in an array, **APPLY DCMEMV** execution immediately outputs -1.0 VDC. Then, the first trigger event changes the output level to 0.0 VDC, the second trigger changes the output to 2.0 VDC, and the third trigger changes the output level to -1.0 VDC.

Selecting Output Function

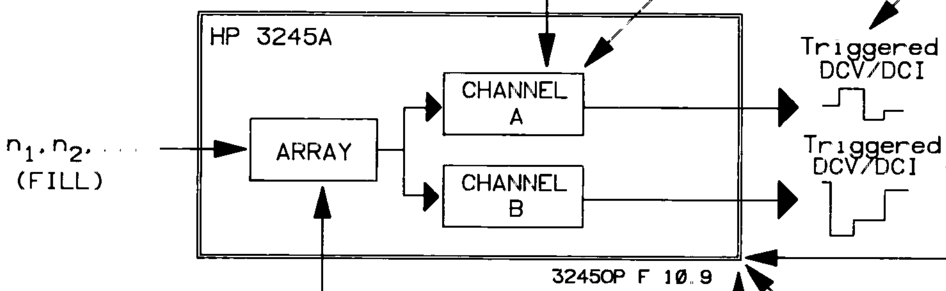
The first step to select triggered DC output parameters is to select the output function with **APPLY DCMEMV** or with **APPLY DCMEMI**. After one of the commands is executed, **OUTPUT?** returns of value of the element with the largest magnitude. You can use the **APPLY?** command to return the output function (DCMEMV or DCMEMI).

Triggered DCV Output (APPLY DCMEMV)

Use **APPLY DCMEMV** *length, array_name* to generate triggered-sequence DC voltage outputs, where *length* is the number of DC voltage values to be output (range is 2 to 2048) and *array_name* is the name of the array in which the output values are stored. Use **DIM** or **REAL** to define the array size and use **FILL** to store the (user-specifed) DC voltage values in the array. Refer to "Defining Array Values" for details on defining/filling arrays.

The valid range for triggered DC voltage outputs is -10.25 VDC to +10.25 VDC with 0 Ω output impedance or -5.125 VDC to +5.125 VDC with 50 Ω output impedance. (The channel should be terminated with 50 Ω in the 50 Ω mode.) With triggered DCV outputs, the first value in the array is output immediately when **APPLY DCMEMV** is executed. Then, the other values in the array are output, one at a time, when a trigger event (specified by the **TRIGIN** command) occurs.

For example, if VOUT is a 4-element array with -1.0, 0, 2.0, and 3.0 as array element values and **APPLY DCMEMV 4,VOUT** is executed, the first element value (-1.0 VDC) is immediately output without a trigger. Then, when the first trigger event (set by **TRIGIN**) occurs, the second array value (0.0) is output, etc. On the fourth trigger, the first array element value (-1.0) is again output, etc.



DC Range (RANGE)

RANGE [max_output] sets fixed output range. DC voltage and current output resolution depends on the output impedance (set by IMP). RANGE AUTO sets autorange.

Use Channel (USE)

USE ch sets use (output channel). USE 0 or USE CHANA sets channel A, USE 100 or USE CHANB sets channel B.

Trigger Event (TRIGIN)

TRIGIN [event] sets the event which generates DCV/DCI outputs <u>after</u> the first output has been generated with an APPLY DCMEMV/DCMEMI command

Output Function (APPLY DCMEMV/DCMEMI)

APPLY DCMEMV volts sets triggered DC voltage output (-10.25 VDC to +10.25 VDC)
APPLY DCMEMI amps sets triggered DC current output (-0.1 A to +0.1 A)

Output Terminal (TERM)

TERM [mode] sets the output terminal
TERM FRONT sets front panel output,
TERM REAR sets rear panel output,
TERM OFF/OPEN disconnects outputs.

Define/Fill Array (DIM/REAL/FILL)

DIM or REAL defines a REAL array to store output values. FILL stores user-specified values in defined array. (You can also use the Edit Array key.)

Output Impedance (IMP)

IMP mode sets the output impedance on the use channel. IMP 0 sets 0Ω impedance, IMP 50 sets 50Ω impedance.

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Default delay time = 0.04 sec

Figure 10-9. Triggered DC Outputs - Command Summary

Triggered DCI Outputs (APPLY DCMEMI)

Use **APPLY DCMEMI** *length, array_name* to generate triggered-sequence DC current outputs, where *length* is the number of DC current values to be output (range is 2 to 2048) and *array_name* is the name of the array in which the output values are stored. Use **DIM** or **REAL** to define the array size and use **FILL** to store the (user-specifed) DC voltage values in the array. Refer to "Defining Array Values" for details on defining/filling arrays.

The valid range for triggered DC current outputs is -0.1 Amps to +0.1 Amps with either 0 Ω or 50 Ω output impedance (and the channel is terminated with 50 Ω in the 50 Ω mode). With triggered DCI outputs, the first value in the array is output immediately when **APPLY DCMEMI** is executed. Then, the other values in the array are output, one at a time, when a trigger event (specified by the **TRIGIN** command) occurs.

For example, if IOUT is a 4-element array with -0.005, 0, 0.002, and 0.003 as array element values and **APPLY DCMEMI 4,IOUT** is executed, the first element value (-0.005 Amps) is immediately output without a trigger. Then, when the first trigger event (set by **TRIGIN**) occurs, the second array value (0.0 Amps) is output, etc. On the fourth trigger, the first array element value (-0.005 Amps) is again output, etc.

Defining Array Values

When you have selected the output function (triggered DCV or triggered DCI), define an array and select the array values required for the triggered output.

NOTE

You can also use the front panel **DIM Array** *and* **Edit Array** *keys to define and fill arrays A0 through A9. Refer to Chapter 6 - Front Panel Operation for details.*

Define Array (DIM/REAL)

To specify varying values for triggered DC outputs, the desired values can be stored in a REAL array defined by **DIM** *array_name* (*max_index*) or by a **REAL** *array_name* (*max_index*). For either command, *name* is the name of the REAL array and *max_index* is one less than the number of elements in the array. The valid range for *max_index* is 0 to 2047 and the lower bound for arrays is always 0. For example, **DIM A(4)** specifies REAL array A of 5 elements [A(0) through A(4)].

Fill Array (FILL)

When an array is defined with **DIM** or **REAL**, you can use the **FILL** *array_name, list* command to enter values into the array. For example, if VOUT is defined as a 4-element REAL array, using **FILL VOUT -1.0, 0.0, 2.0, 3.0** places -1.0 in element VOUT(0), 0.0 in VOUT(1), 2.0 in VOUT(2), and 3.0 in VOUT(3). When **APPLY DCMEMV** or **APPLY DCMEMI** is executed, the value in the first array element is output. Then, each time the trigger event occurs, the next value in the array is output. Again, note that the output value "wraps around" to the first element value after cycling through all element values.

Selecting Channel Parameters

When the output function has been selected and array values defined, select the channel parameters required. Except for DC resolution, commands for triggered DC outputs are the same as for modified DC outputs. Also, **DCRES** has no effect for triggered DC outputs since **APPLY DCMEMI** or **APPLY DCMEMV** operate <u>only</u> in low-resolution mode.

Output range values (selected by the **ARANGE**, **IMP**, or **RANGE** commands) which are specified after **APPLY DCMEMV** or **APPLY DCMEMI** has been executed do not take effect until another **APPLY DCMEMV** or **APPLY DCMEMI** command is executed. (**APPLY DCMEMV** or **APPLY DCMEMI** retains the range which is selected at the time the voltage or current values are transferred to the HP 3245A internal storage buffer.) However, the **MON STATE** display and other query commands are updated as soon as the range values are changed.

Selecting Trigger Event

When the output function, output parameters, and array values have been selected, select the trigger event with the **TRIGIN** [*event*] command. Use **TRIGIN?** to return the trigger event for the **USE** channel. The **TRIGIN** [*event*] parameters are shown in Table 10-5. Power-on *event* = **HIGH**, default *event* = **SGL**. Refer to Chapter 13 - Triggering Outputs/Waveforms for a discussion of trigger events. Software triggering, as set with **TRIGIN SGL**, is used for all example programs in this chapter.

Table 10-5. DC Outputs - Trigger Event Parameters

event	Definition
TBO	Trigger Bus 0. Trigger using the signal on the TBO trigger bus.
TB1	Trigger Bus 1. Trigger using the signal on the TB1 trigger bus.
EXT	Trigger Connector. Trigger using the input to the front panel Trigger connector.
EXTBAR	Inverse of EXT. Trigger using the inverse of input to front panel Trigger connector.
LOW	Low Signal - Software Control. Used with HIGH parameter to internally trigger.
HIGH	High Signal - Software Control. Used with LOW parameter to internally trigger.
HOLD	Same as HIGH parameter.
SGL	Single Trigger. Assure a HIGH then change to LOW and then to HIGH to internally trigger.

Programming Examples

When you have selected the output function, defined the array values, selected the channel parameters, and the trigger event required, program the HP 3245A for the desired output. Figure 10-10 shows a suggested sequence of commands to program triggered DC outputs.

Note that the programming sequence shown in Figure 10-10 is NOT the same as the steps to select the output parameters shown in Figure 10-8. Also, note that the trigger event set by **TRIGIN** is programmed after the **RANGE** and **APPLY** commands to avoid triggering the channel prematurely. Example programs follow to generate triggered DC voltage and current outputs.

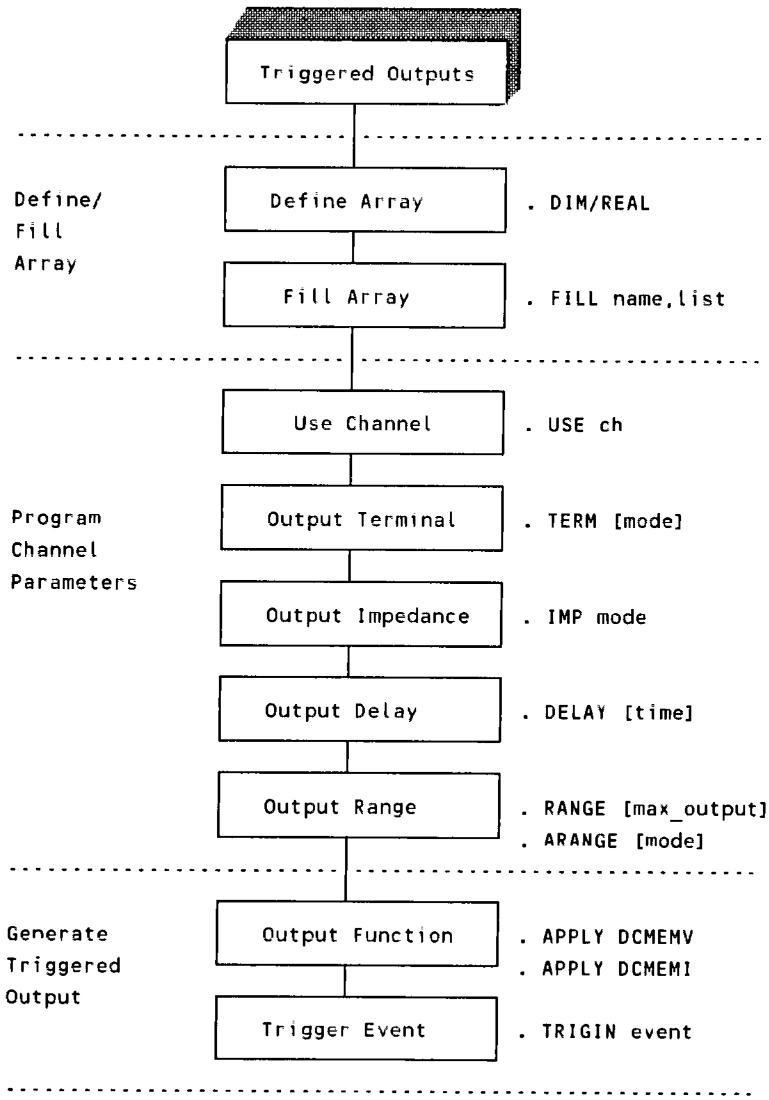


Figure 10-10. Triggered DC Outputs - Programming Sequence

Example 10-6: Triggered DC Voltage Outputs (TRIGV10)

This program outputs a triggered DC voltage sequence from channel A consisting of four voltage values: 1.1, 2.2, 3.3, and 4.4 VDC. When the program executes, **APPLY DCMEMV** outputs the first value in the array (1.1 VDC). The remaining voltages are output, one at a time, when **TRIGIN SGL** (a software trigger) is executed. Note that the first output (1.1 VDC) is generated immediately without a trigger.

```
10   !file TRIGV10
20   !
30   CLEAR 709
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Delete vars, arrays, subs
60   OUTPUT 709;"DIM VOUT(3)"         !Dimension 4-element array
70   OUTPUT 709;"FILL VOUT 1.1,2.2,3.3,4.4"    !Enter array values
80   OUTPUT 709;"APPLY DCMEMV 4,VOUT"  !Output = 1.1 VDC
90   OUTPUT 709;"TRIGIN SGL"          !Output = 2.2 VDC
100  OUTPUT 709;"TRIGIN SGL"          !Output = 3.3 VDC
110  OUTPUT 709;"TRIGIN SGL"          !Output = 4.4 VDC
120  OUTPUT 709;"TRIGIN SGL"          !Output = 1.1 VDC (wrap-around)
130  OUTPUT 709;"TRIGIN SGL"          !Output = 2.2 VDC
140  END
```

Example 10-7: Triggered DC Current Outputs (TRIGI10)

This program outputs a triggered DC current sequence from channel A consisting of four current values: 0.001, 0.002, 0.003, and 0.004 Amps. When the program executes, **APPLY DCMEMI** outputs the first value in the array (0.001 Amps). The remaining current levels are output, one at a time, when **TRIGIN SGL** (a software trigger) is executed. Note that the first output (0.001 Amps) is generated immediately without a trigger.

```
10   !file TRIGI10
20   !
30   CLEAR 709
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Delete vars, arrays, subs
60   OUTPUT 709;"DIM IOUT(3)"         !Dimension 4-element array
70   OUTPUT 709;"FILL IOUT .001,.002,.003,.004"    !Enter array values
80   OUTPUT 709;"APPLY DCMEMI 4,IOUT"  !Output = 0.001 Amps
90   OUTPUT 709;"TRIGIN SGL"          !Output = 0.002 Amps
100  OUTPUT 709;"TRIGIN SGL"          !Output = 0.003 Amps
110  OUTPUT 709;"TRIGIN SGL"          !Output = 0.004 Amps
120  OUTPUT 709;"TRIGIN SGL"          !Output = 0.001 Amps (wrap-around)
130  OUTPUT 709;"TRIGIN SGL"          !Output = 0.002 Amps
140  END
```

Contents

Chapter 11
Programming Defined Waveforms

Chapter 11
Programming Defined Waveforms

Introduction

This chapter shows how to program the HP 3245A for defined (sine, ramp, and square wave) AC voltage and current waveforms. Refer to the following chapters to generate DC outputs or arbitrary AC waveforms and other unique signals.

- Chapter 10 - Generating DC Outputs.
- Chapter 12 - Generating Arbitrary Waveforms.
- Chapter 13 - Triggering Outputs/Waveforms.
- Chapter 14 - Advanced Programming Topics.

Chapter Contents

The chapter contents are:

- **Programing Power-On AC Waveforms** shows how to program the HP 3245A for defined AC voltage and current waveforms when power-on conditions are set.

- **Programming Modified AC Waveforms** shows how to program the HP 3245A for defined AC voltage and current waveforms when waveform parameters are modified from their power-on settings.

- **Programming Triggered AC Waveforms** shows how to program the HP 3245A for defined AC voltage and current waveforms when waveform parameters are modified and/or triggering is used.

NOTE

All programs in this chapter are stored on the Example Programs disc. Refer to Chapter 1 - Using This Manual for details on using the disc. Refer to the HP 3245A Command Reference Manual for command details. If you are using the front panel keyboard to generate defined AC waveforms, refer to Chapter 6 - Front Panel Operation.

Defined Waveforms Overview

Figure 11-1 shows the three modes of defined AC waveforms discussed in this chapter (power-on, modified, and triggered) and shows a suggested set of steps to select the waveform function, the channel and waveform parameters, and the trigger mode/event.

Table 11-1 shows power-on conditions for commands which apply to AC waveforms. If power-on conditions can be used, refer to "Programming Power-On AC Waveforms". If power-on condtions must be modified, but triggering is not required, refer to "Programming Modified AC Waveforms". If power-on conditions must be modified and/or triggering is required, refer to "Programming Triggered AC Waveforms".

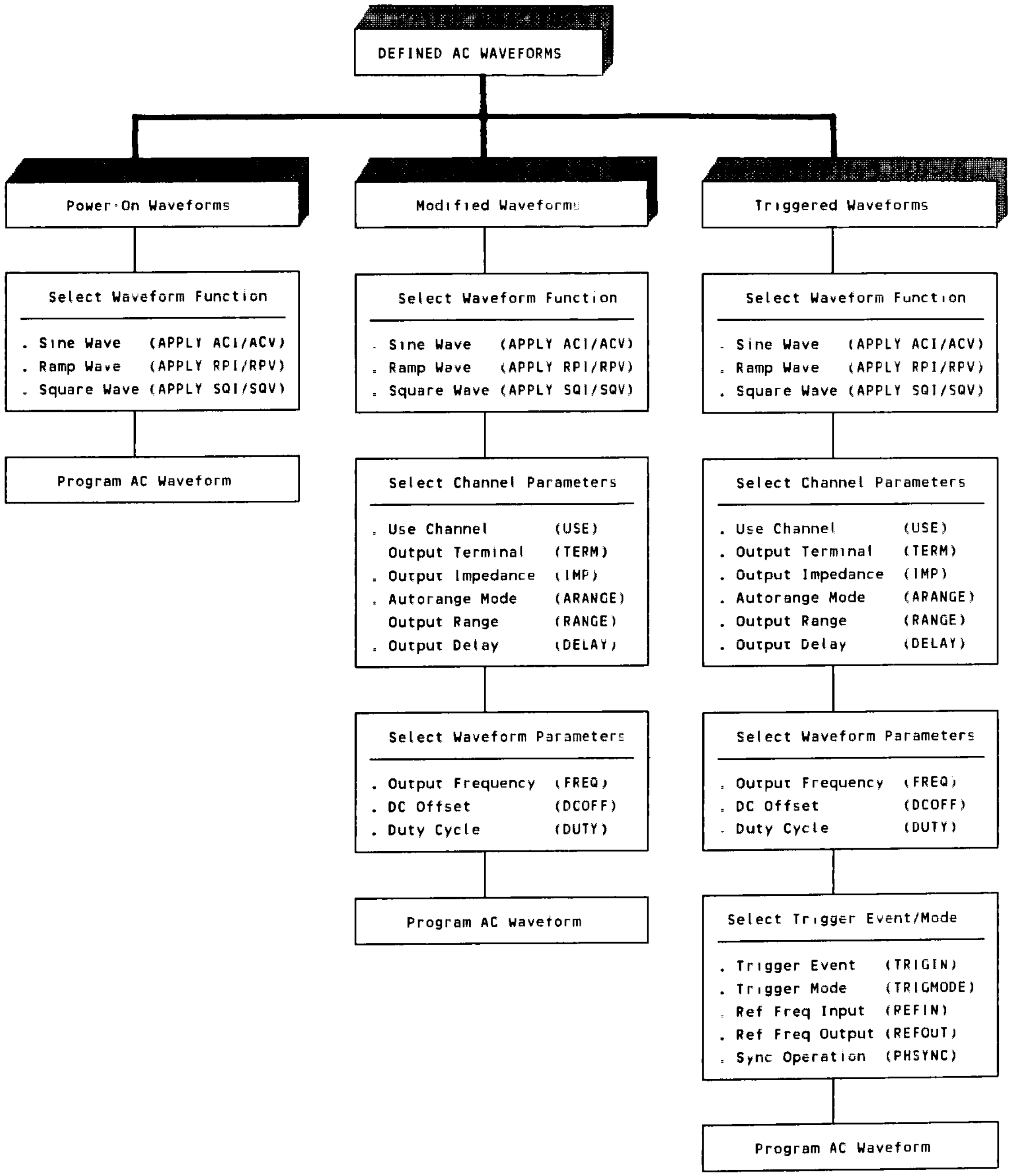


Figure 11-1. Defined AC Waveforms - Overview

Table 11-1. AC Waveforms - Power-On State

Chan/Waveform Para	Command	Power-On
Output Function	APPLY	DC Volts
Use Channel	USE	Channel A
Output Terminal	TERM	Front panel
Output Impedance	IMP	0 Ohms
Autorange Mode	ARANGE	Enabled
Output Range	RANGE	1V (auto)
Output Delay	DELAY	0.04 sec
Waveform Frequency	FREQ	1000 Hz
DC Offset	DCOFF	0 Volts
Duty Cycle	DUTY	50%

Programming Power-On AC Waveforms

This section show how to generate defined (sine, ramp, or square wave) voltage or current waveforms when power-on conditions are set. As shown in Figure 11-2, for power-on AC waveforms you only need to select the waveform function (sine, ramp, or square), required and then program the HP 3245A for the desired waveform with an **APPLY** command.

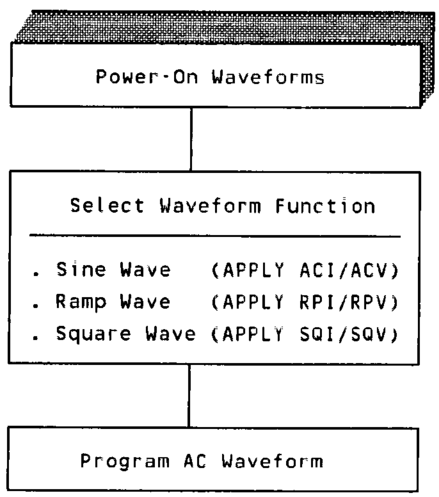


Figure 11-2. Power-On AC Waveforms - Selection Steps

Command Summary

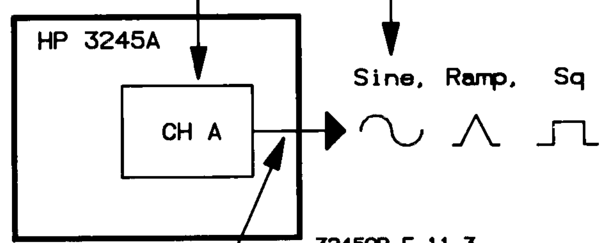
Figure 11-3 summarizes channel operation and related commands for power-on defined AC waveforms. The waveform function is set with **APPLY ACV/ACI** (sine wave), **APPLY RPV/RPI** (ramp wave), or **APPLY SQV/SQI** (square wave). With power-on conditions, the selected waveform is output from the channel A front panel Output terminal. Channel A is set for 0 Ω output impedance, high-resolution mode, and autorange. The delay time between commands is defaulted to the values shown in Table 11-3. The waveform frequency is 1000 Hz, with 0 volts DC offset. For ramp and square waves, the duty cycle is 50%.

Power-On Channel Parameters

USE 0	Channel A is USE channel
TERM FRONT	Front panel output
IMP 0	0 Ω output impedance.
ARANGE ON	Autorange function enabled.
RANGE AUTO	Autorange set.
DELAY	Default delays (Table 11-3).

Power-On Waveform Functions

APPLY ACV volts	Voltage sine waveform.
APPLY RPV volts	Voltage ramp waveform.
APPLY SQV volts	Voltage square waveform.
APPLY ACI amps	Current sine waveform.
APPLY RPI amps	Current ramp waveform.
APPLY SQI amps	Current square waveform



32450P F.11.3

Power-On Waveform Parameters

FREQ 1000	1000 Hz waveform frequency.
DCOFF 0	0 volts DC offset
DUTY 50	50% duty cycle (ramp/sq waves).

Figure 11-3. Power-On AC Waveforms - Command Summary

Selecting Waveform Function

As noted, the only step required to generate defined AC waveforms is to select the waveform function with an **APPLY** command. The amplitude range for AC voltage waveforms is from 0.03125 Vac PP to 20 Vac PP with 0 Ω output impedance or from 0.015625 Vac PP to 10 Vac PP with 50 Ω output impedance. The range for AC current waveforms is from 0.00001 Amps AC PP to 0.2 A AC PP.

When fixed range is set with **RANGE**, voltage range is from 10% to 100% of **RANGE** setting and current range is from 5% to 100% of **RANGE** setting. You can use the function query command (**APPLY?**) to return the waveform function string (such as **ACV**, etc.) or you can use the **OUTPUT?** command to return the output value. AC voltage values are returned in Vac PP. AC current values in Amps AC PP.

Sine Waveforms (APPLY ACV/ACI)

You can generate sine waveforms with an **APPLY ACV** or an **APPLY ACI** command. **APPLY ACV** *pp_amplitude* outputs sine voltage waveforms with peak-to-peak amplitude (in Vac) equal to the *pp_amplitude* parameter value. **APPLY ACI** *pp_amplitude* outputs sine current waveforms with peak-to-peak amplitude (in Amps) equal to the *pp_amplitude* parameter value.

Ramp Waveforms (APPLY RPV/RPI)

You can generate ramp waveforms with an **APPLY RPV** or an **APPLY RPI** command. **APPLY RPV** *pp_amplitude* outputs ramp voltage waveforms with peak-to-peak amplitude (in Vac) equal to the *pp_amplitude* parameter value. **APPLY RPI** *pp_amplitude* outputs ramp current waveforms with peak-to-peak amplitude (in Amps) equal to the *pp_amplitude* parameter value.

With power-on/reset conditions, ramp waveforms are generated with 0 VDC offset, 50% duty cycle, and output frequency of 1000 Hz. For ramp and square waves, the duty cycle can be varied from 5% to 95% with the **DUTY** command. Refer to "Duty Cycle (DUTY)" for details on the duty cycle for ramp and square waves.

Square Waveforms (APPLY SQV/SQI)

You can generate square waveforms with an **APPLY SQV** or an **APPLY SQI** command. **APPLY SQV** *pp_amplitude* outputs square voltage waveforms with peak-to-peak amplitude (in Vac) equal to the *pp_amplitude* parameter value. **APPLY SQI** *pp_amplitude* outputs square current waveforms with peak-to-peak amplitude (in Amps) equal to the *pp_amplitude* parameter value.

With power-on/reset conditions, square waveforms are generated with 0 VDC offset, 50% duty cycle, and output frequency of 1000 Hz. For ramp and square waves, the duty cycle can be varied from 5% to 95% with the **DUTY** command. Refer to "Duty Cycle (DUTY)" for details on the duty cycle for ramp and square waves.

Programming Examples

When you have selected the waveform function and amplitude, program the HP 3245A for the desired waveform. Figure 11-4 shows the suggested programming sequence. Example programs follow to generate sine and square wave voltage AC waveforms.

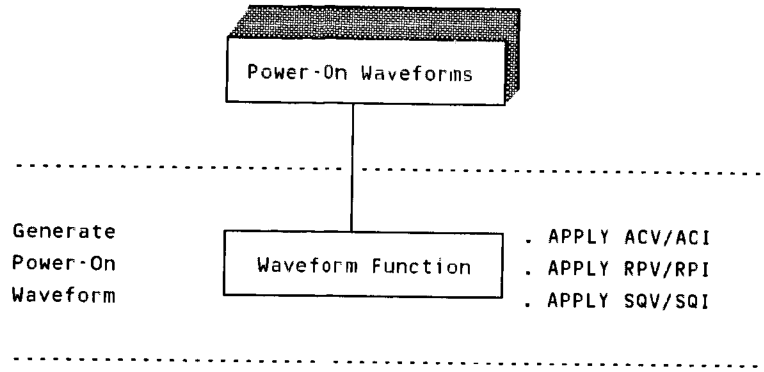


Figure 11-4. Power-On AC Waveforms - Programming Sequence

Example 11-1: Power-On Sine Waveform (SINEP11)

This program outputs a voltage sine wave @ 3.0 Vac PP and 1000 Hz from channel A. The waveform amplitude is returned to the controller CRT. The HP 3245A is set for power-on/default conditions.

```
10   !file SINEP11
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   OUTPUT 709;"APPLY ACV 3.0"      !Gen sine wave @3.0 Vac PP
70   OUTPUT 709;"OUTPUT?"            !Query sine wave PP value
80   ENTER 709;A                     !Enter PP value
90   PRINT "Output =";A;"Vac PP"     !Display PP value
100  END
```

A typical controller return follows. Figure 11-5 shows a typical oscilloscope display of the waveform. If you measure the RMS value on an HP 3458A DMM, the result will be 1.0608 Vac RMS, since $V_{RMS} = V_{PP}/2.828$ for a voltage sine wave.

```
Output = 3 Vac PP
```

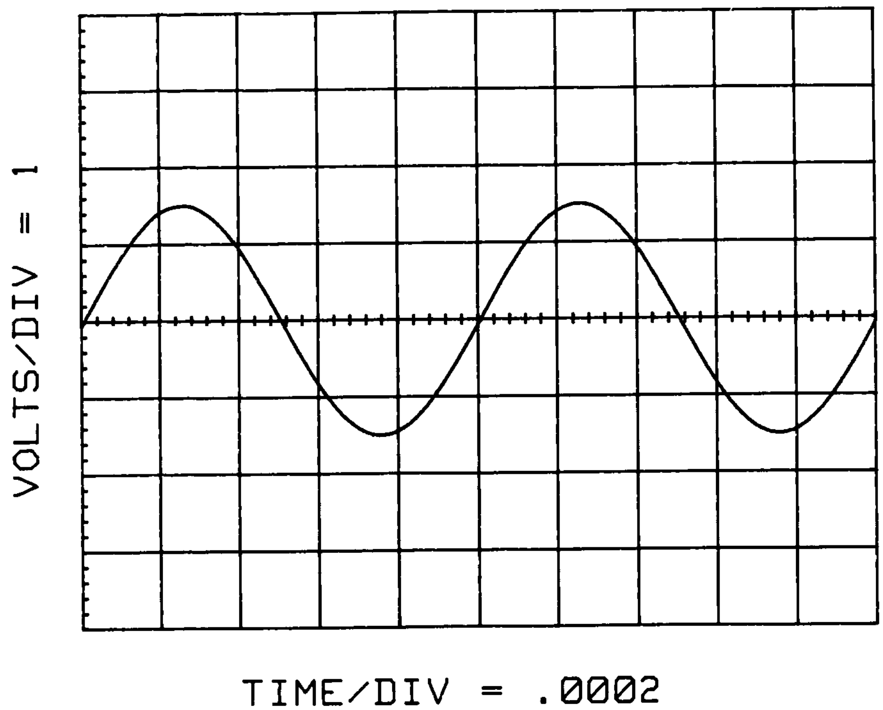


Figure 11-5. Example: Power-On Sine Waveform

Example 11-2: Power-On Square Waveform (SQREP11)

This program generates a 50% duty cycle square waveform @ 3.0 Vac PP and 1000 Hz from channel A. The waveform value is returned to the controller CRT. The HP 3245A is set for power-on/default conditions. (To generate a 3.0 Vac PP ramp waveform @ 1000 Hz and 50% duty cycle, use **APPLY RPV 3.0** in line 60.)

```
10    !file SQREP11
20    !
30    CLEAR 709                        !Clear HP 3245A
40    OUTPUT 709;"RST"                 !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"             !Clear HP 3245A memory
60    OUTPUT 709;"APPLY SQV 3.0"       !Gen square wave @3.0 Vac PP
70    OUTPUT 709;"OUTPUT?"             !Query square wave PP value
80    ENTER 709;A                      !Enter PP value
90    PRINT "Output =";A;"Vac PP"      !Display PP value
100   END
```

A typical controller return follows. Figure 11-6 show a typical oscilloscope display of the waveform. If you measure the RMS waveform value on an HP 3458A DMM, the result will be 1.5000 Vac RMS, since $V_{RMS} = V_{PP}/2.00$ for a 50% duty cycle square wave. (For a 50% duty cycle ramp waveform, the result is 3.000/3.464 = 0.8667 Vac RMS.)

```
Output = 3 Vac PP
```

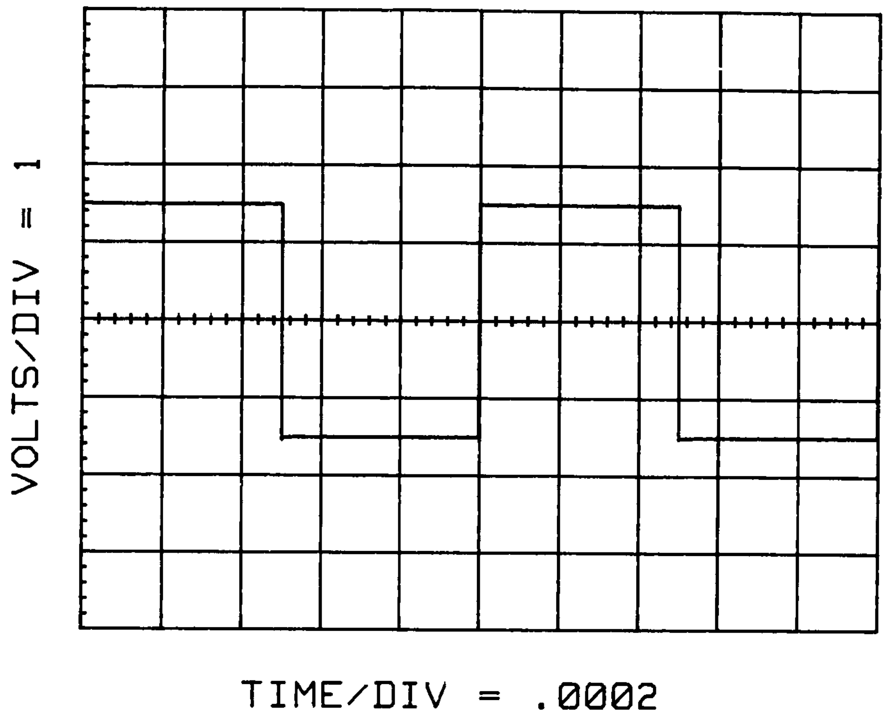


Figure 11-6. Example: Power-On Square Waveform

Programming Modified AC Waveforms

This section shows how to generate defined AC waveforms with parameter(s) modified from their power-on settings. As shown in Figure 11-7, there are three steps to select functions and parameters for modified AC waveforms First, select the waveform function, next select the channel parameters, and then select the waveform parameters required. (Refer to Chapter 12 - Programming Arbitrary Waveforms for methods to output sine and ramp waves faster using precomputing.)

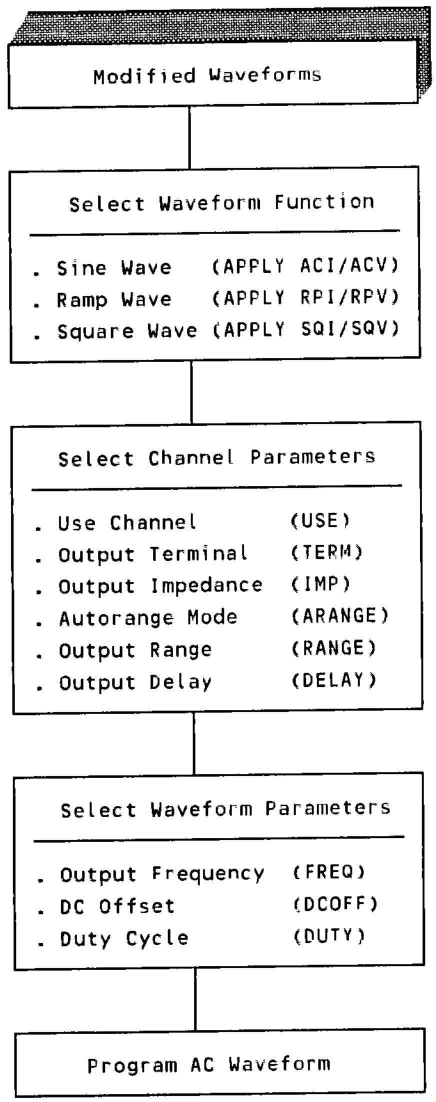


Figure 11-7. Modified AC Waveforms - Selection Steps

Command Summary

Figure 11-8 summarizes channel operation and related commands for modified AC waveforms. For this discussion, parameters are grouped into two categories: channel parameters and waveform parameters.

Channel Parameter Commands

There are five channel parameters which can be selected: the use channel, output terminal, output impedance, range, and output delay. **USE** sets the output channel (A or B) and **TERM** sets the output terminal (front or rear). **IMP** sets 0 Ω or 50 Ω output impedance, and **RANGE** or **ARANGE** sets a fixed range or autorange. **DELAY** sets the delay time (from 0 to 9.99 seconds) between output commands.

Waveform Parameter Commands

Three waveform parameters can be set for modified AC waveforms: waveform frequency, DC offset, and duty cycle. **FREQ** sets the waveform frequency (from 0 Hz to 1 MHz), **DCOFF** sets the DC offset, and **DUTY** sets the duty cycle (from 5% to 95%) for (ramp and square waveforms only).

Selecting Waveform Function

The first step to program modified AC waveforms is to select the output function (sine, ramp, or square wave) and the output amplitude (in Volts or Amps). Refer to "Programming Power-On AC Waveforms" for a discussion of AC waveform functions.

Selecting Channel Parameters

When you have selected the waveform function, select the channel parameters required. As shown in Figure 11-8, there are six commands which can be used to set channel operating parameters: **USE**, **TERM**, **IMP**, **ARANGE**, **RANGE**, and **DELAY**.

Use Channel (USE)

At power-on, the **USE** channel (the channel to receive output commands) is set to channel A. If you have a two-channel version of an HP 3245A, you can use **USE 0** or **USE CHANA** to select channel A or use **USE 100** or **USE CHANB** to select channel B. **USE?** returns the current **USE** channel (channel A or B). **USE?** returns "0" if channel A is the **USE** channel or returns "100" if channel B is the **USE** channel.

NOTE

You can simultaneously have two **USE** *channels - one for commands entered from the front panel and the other for commands entered over HP-IB (e.g.,* **USE 0** *over HP-IB and* **USE 100** *entered from the front panel).*

Channel Parameters

Use Channel (USE)

Use ch sets use (output) channel
USE 0 or USE CHANA sets channel A
USE 100 or USE CHANB sets channel B

Output Terminal (TERM)

TERM [mode] sets the output terminal
TERM FRONT sets front panel output
TERM REAR sets rear panel output and
TERM OFF OPEN disconnects outputs

Output Impedance (IMP)

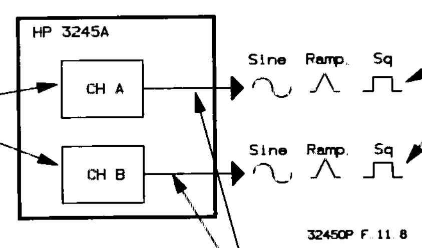
IMP mode sets the output impedance on the use channel. IMP 0 sets 0 Ω impedance. IMP 50 sets 50 Ω impedance

AC Range (RANGE/ARANGE)

ARANGE [mode] sets autorange mode and RANGE [max_output] sets fixed output range. AC voltage output resolution depends on the output impedance (set by IMP). RANGE AUTO sets autorange

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Delay range is from 0 to 9.99 seconds with 0.01 seconds resolution



Waveform Functions

Voltage Waveforms (APPLY volts)

APPLY ACV volts sets sine wave voltage waveform. APPLY RPV volts sets ramp wave voltage waveform and APPLY SQV volts sets square wave voltage waveform

Current Waveforms (APPLY amps)

APPLY ACI amps sets sine wave current waveform. APPLY RPI amps sets ramp wave current waveform and APPLY SQI amps sets square wave current waveform

Waveform Parameters

Waveform Frequency (FREQ)

FREQ freq1 [,freq2] sets waveform frequency (in Hz). Range is 0 to 1 MHz (ramp performance not specified above 100 kHz). Power-on frequency = 1000 Hz (freq2 applies to dual-freq mode only)

DC Offset (DCOFF)

DCOFF volts sets DC offset on use channel. Peak amplitude of waveform plus magnitude of DC offset must be ≤50% of selected AC range

Duty Cycle (DUTY)

DUTY %duty sets the duty cycle for AC ramp and square waves. Duty cycle range is 5% to 95% with power-on duty = 50%. For square waves 50% duty cycle only for frequencies above 100 kHz

Figure 11-8. Modified AC Waveforms - Command Summary

Output Terminal (TERM)

Select the output terminal for the **USE** channel with the **TERM** [*mode*] command. **TERM FRONT** sets the front panel Output terminal and **TERM REAR** sets the rear panel Output terminal for the **USE** channel. **TERM OFF** or **TERM OPEN** disconnects all outputs. The power-on/reset setting is **TERM FRONT**. **TERM?** returns the current output terminal setting for the **USE** channel. The output must be from the front panel or from the rear panel. The signal cannot be output simultaneously from both output ports.

NOTE

EXCESSIVE TERMINAL VOLTAGE ERROR. If an external signal is applied to the selected Output terminal and is large enough to damage the instrument (>± 15 V peak), the HP 3245A automatically opens the appropriate output relay. When an output relay opens, the "EXCESSIVE TERMINAL VOLTAGE" error is generated. To continue operation, remove the external signal and reset the HP 3245A.

Output Impedance (IMP)

For AC voltage waveforms, **IMP** *mode* sets the output impedance on the use channel (**IMP** does not apply to current outputs). **IMP 0** selects the 0 Ω mode, while **IMP 50** selects the 50 Ω mode. The power-on/reset setting is **IMP 0**. The **IMP?** command returns the impedance setting on the **USE** channel.

Actual vs. Programmed Outputs

In the 0 Ω (**IMP 0**) mode, the voltage output from the use channel equals the programmed output voltage with or without load termination. For example, if you program channel A to output 2 Vac in the 0 Ω mode, the actual output voltage will be 2 Vac with or without load termination.

However, in the 50 Ω mode (**IMP 50**), the actual output voltage equals the programmed output voltage ONLY when the channel is terminated with a 50 Ω load. For example, if you program channel A to output 2 Vac in the 50 Ω mode, the actual output voltage will be 2 Vac only when channel A is terminated with a 50 Ω load. If the load is removed and replaced with an open circuit, the actual output voltage will double to 4 Vac.

OUT OF RANGE Error

If the present DC or AC output voltage is generated in the 0 Ω mode and is greater than the maximum allowable output in the 50 Ω mode, selecting **IMP 50** will generate an "OUT OF RANGE" error. For example, if you previously set channel A to output 16 Vac in the 0 Ω mode, selecting **IMP 50** will generate an error. Similarly, if the present AC output voltage is generated in the 50 Ω mode and is less than 0.03125 Vac PP (the smallest peak-to-peak voltage which can be generated in the 0 Ω mode), selecting **IMP 0** will generate an "OUT OF RANGE" error.

NOTE

When the output impedance or function is changed, the autorange mode is automatically enabled.

High-Frequency AC Outputs

AC waveforms output in the 0 Ω mode may not drive 50 Ω cables properly at frequencies above 100 kHz. Therefore, select the 50 Ω mode when using 50 Ω cables and outputs above 100 kHz. Also, generating high-frequency waveforms with amplitudes greater than 10 Vac PP in the Ω mode may cause the HP 3245A to output unpredictable waveforms.

Autorange Mode (ARANGE)

The autorange mode is enabled or disabled with the **ARANGE** [*mode*] command. Also, autorange mode is enabled at power-on and when commands which change the DC resolution mode (**DCRES** command for DCV/DCI), output impedance (**IMP** command), or output function (**APPLY** commands) are executed. **RANGE?** returns the present setting on the **USE** channel.

Autorange Enabled (ARANGE ON)

With autorange mode enabled (**ARANGE ON**), the HP 3245A automatically selects the lowest voltage or current range which will provide maximum accuracy for the desired output level. An error is generated if you attempt to set the output level greater than 100% of the highest peak-to-peak voltage or current range available. An error is also generated if you attempt to set the peak-to-peak amplitude less than 10% of the lowest peak-to-peak voltage range (less than 5% of the lowest peak-to-peak current range)

Autorange Disabled (ARANGE OFF)

With autorange mode disabled (**ARANGE OFF**), the HP 3245A retains its present range regardless of the programmed output value. If the programmed value attempts to exceed 100% of the present peak-to-peak range, an error is generated. As with **ARANGE ON**, an error is also generated if you attempt to set the peak-to-peak amplitude less than 10% of the lowest peak-to-peak voltage range (less than 5% of the lowest peak-to-peak current range).

AC Range (RANGE)

For many AC waveform applications, using autorange will be sufficient. However, if you want to set a fixed range, you can do so with the **RANGE** command. **RANGE** [*max_output*] selects a specific output range or enables the autorange function on the use channel. **RANGE?** returns the present range setting on the **USE** channel.

Voltage and Current Ranges

Table 11-2 shows AC voltage and current ranges and maximum programmed output values. The values shown assume that the channel is terminated with a 50 Ω load and is set to the 50 Ω mode Refer to "Output Impedance (IMP)" for details on output impedance.

Table 11-2. AC Voltage/Current Ranges (RANGE)

AC Current Outputs		
Range (Amps AC)	Maximum Programmed Output	Resolution
0.0002A	0.0002 A	50 nA
0.002A	0.002 A	500 nA
0.02A	0.02 A	5 µA
0.2A	0.2 A	50 µA

AC Voltage Outputs (Sine, Ramp, Square, Arb)			
Output Impedance (Ohms)	Range (Vac PP)	Maximum Programmed Output (Vac PP)	Resolution
0	0.3125V	0.3125 V	156 µV
0	0.625V	0.625 V	312 µV
0	1.25V	1.25 V	625 µV
0	2.5V	2.5 V	1.25 mV
0	5V	5 V	2.5 mV
0	10V	10 V	5 mV
0	20V	20 V	10 mV
50	0.15625V	0.15625 V	78 µV
50	0.3125V	0.3125 V	156 µV
50	0.625V	0.625 V	312 µV
50	1.25V	1.25 V	625 µV
50	2.5V	2.5 V	1.25 mV
50	5V	5 V	2.5 mV
50	10V	10 V	5 mV

Selecting a Range

To select a voltage or current range, specify *max_output* as the maximum expected output voltage (in Vac PP) or current (in Amps AC PP). The channel then selects the correct range, based on the previous mode (DCI, DCV, ACI, or ACV) of the use channel. To enable the autorange function, use the word "AUTO" for *max_output* or default the parameter.

For voltage outputs, the range is from 10% to 100% of the peak-to-peak range selected. For current outputs, the range is from 5% to 100% of the peak-to-peak range selected. When selecting the output ranges with **RANGE**, make the range selection AFTER the resolution mode, output impedance, and output function have been chosen. To avoid the "OUT OF RANGE" error, do not execute **RANGE** until you have ensured that the desired output level can be generated on the desired range.

Output Delay (DELAY)

As required, you can change the delay time between output commands with the **DELAY** command. **DELAY** [*time*] specifies a time interval during which outputs from the **USE** channel are allowed to settle before the next command is executed. The **DELAY?** command returns the present delay time (in seconds). If no delay time has been programmed with the **DELAY** command, **DELAY?** returns the default delay time based on the values shown in Table 11-3.

Setting Delay Time

The **DELAY** [*time*] parameter specifies the settling time (range = 0 to 9.99 seconds) with 0.01 second resolution. At power-on/reset, the default *time* is automatically determined by the function, range, DC resolution, and output impedance set on the **USE** channel.

Default Delays

If a delay time is not specified, the HP 3245A determines a default delay based on present function, range, DC resolution, and output impedance. If the output function, range, resolution, or output impedance are changed without specifying a new delay, the default delay time is updated based on the new selection. Table 11-3 shows default delay times for all output functions.

Table 11-3. AC Waveforms - Default Delay Times

Change In:	Default Delay
Output Function:	
High-res DCV/DCI (from other function)	1 sec
All other function changes	30 msec
Output Value:	
High-resolution mode	40 msec
Low-resolution mode (same range)	0
Triggered DCV/DCI (first value)	30 msec
Triggered DCV/DCI (after first value)	0
All other modes	30 msec
Range:	
High-resolution mode	40 msec
All other modes	30 msec
Output Impedance (Voltage Only):	
High-resolution mode	40 msec
All other modes	30 msec

Changing Delay Times

Once a delay time is set with **DELAY**, the value does not change until another value is specified or the **USE** channel is changed. A delay time can be set which is shorter than the default value, but the resultant settling time may be too short to allow the channel to meet accuracy specifications for that output function.

Commands Affected by DELAY

The **DELAY** command affects commands which control the waveform function (**APPLY** commands); the output range (**ARANGE** and **RANGE**); and output impedance (**IMP**).

Selecting Waveform Parameters

When you have selected the channel parameters required, next select the AC waveform parameters required. For modified AC waveforms, use **FREQ** to set the waveform frequency, **DCOFF** to set the DC offset, and **DUTY** to set the waveform duty cycle (for ramp and square waveforms only).

Waveform Frequency (FREQ)

The **FREQ** *freq_1* [*freq_2*] command sets the waveform frequency for sine, ramp, square, and arbitrary waveforms on the **USE** channel. (The [*freq_2*] parameter applies only to dual-frequency mode operation. Refer to "Dual-Frequency Mode (TRIGMODE DUALFR)" for details.) **FREQ?** returns the frequency settings for the **USE** channel.

The output frequency range for sine, square, and arbitrary waveforms is 0 to 1 MHz with 0.001 Hz resolution. However, ramp waveform performance is not specified above 100 kHz and will degrade substantially above this frequency Power-on/reset frequency is 1000 Hz, 1000 Hz.

DC Offset (DCOFF)

Use **DCOFF** *volts* to set the DC offset voltage for sine, ramp, square, and arbitrary waveform voltage outputs. Or, use **DCOFF** *amps* to set the DC offset voltage for sine, ramp, square, and arbitrary waveform current outputs. **DCOFF?** returns the DC offset value on the **USE** channel (Vac PP or Amps AC PP).

The *volts* or *amps* Parameter

The *volts* or *amps* parameter sets the offset value in DC volts. *volts* or *amps* must be a positive or negative number such that the peak AC value of the output waveform, plus the magnitude of the offset, does not exceed 50% of the selected peak-to-peak range. If the autorange function is enabled (**ARANGE ON** or **RANGE AUTO**), the HP 3245A automatically changes range as required to accomodate the offset voltage. Power-on/reset *volts* or *amps* = 0.

NOTE

Selecting a DC offset which is larger than the waveform peak-to-peak amplitude may result in a range selection which is too high to produce the desired amplitude. The amplitude must be set to at least 10% of the selected peak-to-peak voltage range or at lease 5% of the selected peak-to-peak current range.

Using DCOFF with Fixed Range (RANGE)

When a fixed range is selected with the **RANGE** command, the magnitude of the voltage peak value plus the magnitude of the DC offset must be ≤ 50% of the fixed range value. For example, if the 10 VPP range is selected, any combination of waveform peak value and DC offset magnitude must be ≤ 5 Volts. Thus, a waveform of 5.0 Vac PP (2.5 Vac peak) and a DC offset of -2.5 Volts (2.5 magnitude) is acceptable, but a waveform of 5.0 Vac and a DC offset >±2.5 Volts will return an "OUT OF RANGE" error.

Using DCOFF in Autorange Mode (ARANGE)

If the autorange mode is enabled (**ARANGE ON** or **RANGE AUTO**) when the **DCOFF** command is executed, the HP 3245A automatically changes range as required to accommodate the offset voltage.

Using DCOFF with Output Impedance Modes (IMP)

With the 0 Ω mode (**IMP 0**), the actual DC offset voltage equals the programmed value with or without load termination. In the 50 Ω mode, the actual offset voltage equals the programmed value ONLY when the channel is terminated with a 50 Ω load.

Duty Cycle (DUTY)

The **DUTY** *%_duty* command sets the duty cycle for ramp and square wave output waveforms. **DUTY?** returns the duty cycle on the use channel. For frequencies up to 100 kHz, you can set the duty cycle between 5% and 95%. The power-on *%_duty* value = 50%. For square waveforms above 100 kHz, setting the duty cycle to a value other than 50% generates an error. Repeated specification of the same duty cycle (e.g., **DUTY 15;DUTY 15**) will cause momentary (approximately 120 msec) irregularities in the waveform. Figure 11-9 shows square and ramp waveforms with 5% and 95% duty cycles.

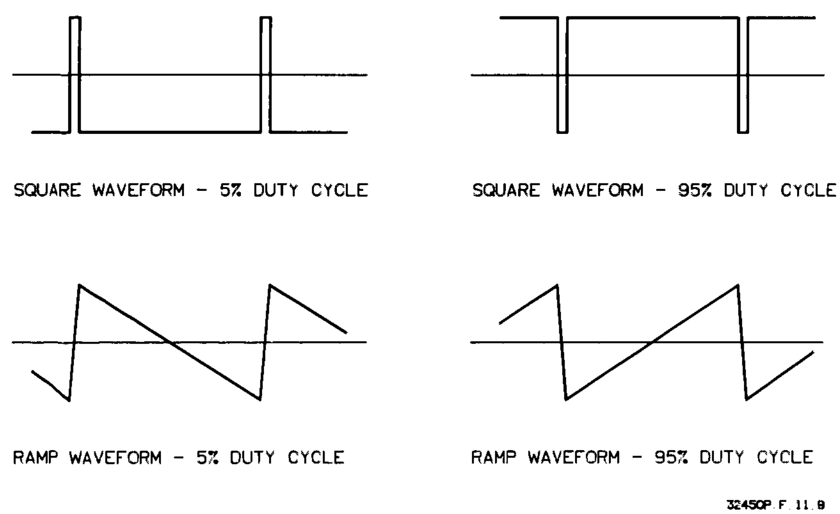


Figure 11-9. Square/Ramp Waveforms - Duty Cycles

Programming Examples

When you have selected the waveform function, channel parameters, and waveform parameters required, program the HP 3245A for the desired waveform. Figure 11-10 shows a suggested sequence of commands to program the **USE** channel for modified AC waveforms

Note that the programming sequence for commands is NOT the same as the sequence of steps to select the waveform function, channel parameters, and waveform parameters shown in Figure 11-7. Example programs follow to generate modified AC waveforms.

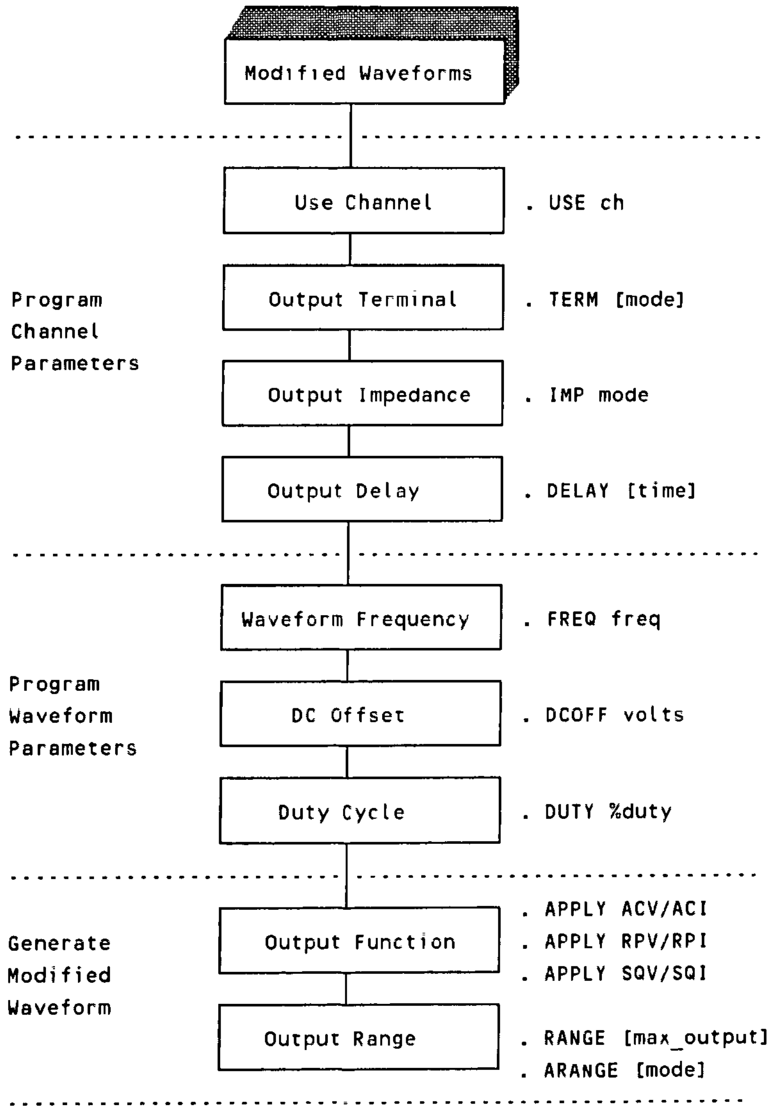


Figure 11-10. Modified AC Waveforms - Programming Sequence

Example 11-3: Modified Sine Waveform (SINEM11)

This program outputs a 4.0 Vac peak-to-peak PP sine waveform from channel A and returns the PP amplitude of the waveform in Volts. The the waveform frequency is set to 2 kHz, the DC offset to -2.0 VDC, and the range is set to the 10 Vac range. The waveform amplitude value is returned to the controller CRT. With these settings, the DC offset amplitude (2.0) plus the waveform amplitude peak value (2.0) is less than 50% of the 10 Vac PP voltage range. Also, the range is set after the **APPLY ACV** command so that the range setting will be for AC voltage, rather than for the DC voltage function set by **RST**

```
10   !file SINEM11
20   !
30   CLEAR 709                         !Clear HP 3245A
40   OUTPUT 709;"RST"                  !Reset HP 3245A
50   OUTPUT 709;"CLEAR"                !Clear HP 3245A memory
60   OUTPUT 709;"FREQ 2E3"             !2 kHz waveform frequency
70   OUTPUT 709;"DCOFF -2.0"           !-2.0 VDC offset voltage
80   OUTPUT 709;"APPLY ACV 4.0"        !Gen 4.0 Vac PP sine wave
90   OUTPUT 709;"OUTPUT?"              !Query PP value of waveform
100  OUTPUT 709;"RANGE 10"             !Set 10 VPP range
110  ENTER 709;A                       !Enter PP value
120  PRINT "Output =";A;"Vac PP"       !Display PP value
130  END
```

A typical CRT display follows. Figure 11-11 shows a typical oscilloscope display of the waveform (set to the DC display mode). If an HP 3458A DMM (or equivalent) is connected to the channel, the value measured is 1.414 Vac RMS, since $V_{RMS} = V_{PP}/2.828$ for sine waveforms

```
Output = 4 Vac PP
```

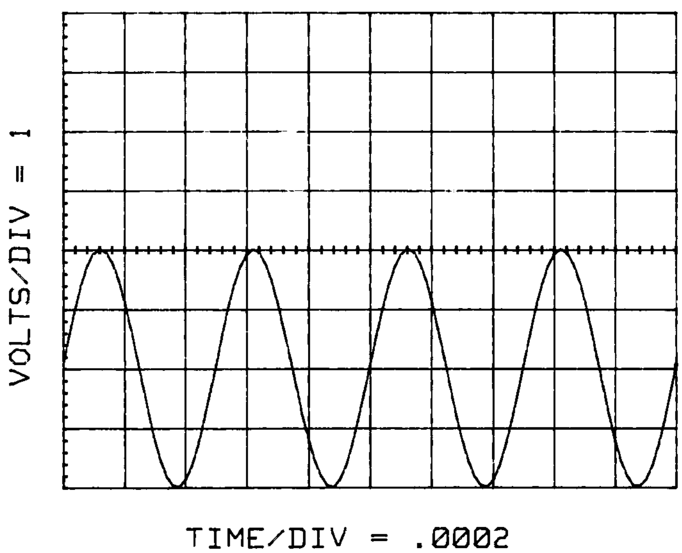


Figure 11-11. Example: Modified AC Sine Waveform

Example 11-4: Modified Ramp Waveform (RAMPM11)

This program outputs a voltage ramp waveform @ 3.0 Vac PP and 2000 Hz from channel A. The duty cycle is set for 30% and the DC offset to 1.0 Volts. The waveform value is returned to the controller CRT.

```
10   !file RAMPM11
20   !
30   CLEAR 709                        !Clear HP 3245A
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Clear HP 3245A memory
60   OUTPUT 709;"FREQ 2000"           !Set output freq to 2000 Hz
70   OUTPUT 709;"DCOFF 1.0"           !Set DC offset of 1.0V
80   OUTPUT 709;"DUTY 30"             !Set 30% duty cycle
90   OUTPUT 709;"APPLY RPV 3.0"       !Output ramp wave @3.0 Vac PP
100  OUTPUT 709;"OUTPUT?"             !Query ramp wave PP value
110  ENTER 709;A                      !Enter PP value
120  PRINT "Output =";A;"Vac PP"      !Display PP value
130  END
```

A typical controller CRT display (in Vac PP) follows. Figure 11-12 shows a typical oscilloscope display (set for DC mode). If you measure the output on an HP 3458A DMM, the result will be 0.866 Vac RMS, since $V_{RMS} = V_{PP}/3.464$ for a ramp waveform.

```
Output = 3.0 Vac PP
```

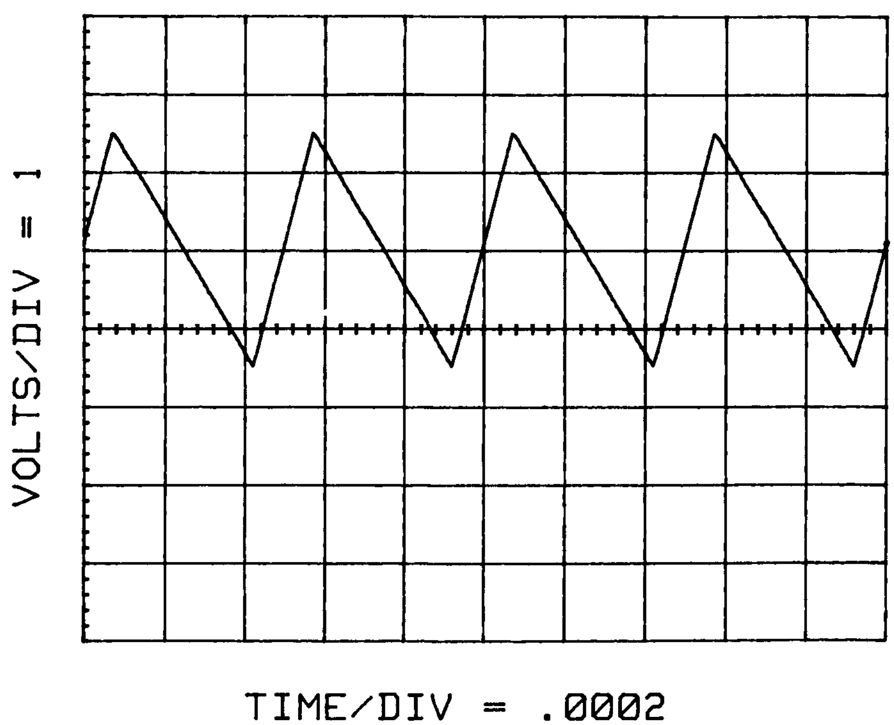


Figure 11-12. Example: Modified AC Ramp Waveform

Example 11-5: Modified Square Waveform (SQREM11)

This program outputs a 0.005 A (5 mA) current square waveform @ 1000 Hz and 50% duty cycle from channel A. For this program, the output impedance is changed from 0 Ω to 50 Ω. The output value is returned to the controller CRT. Since this is a current waveform, changing the output impedance does not affect the output level even if the channel is not terminated in a 50 Ω load.

```
10   !file SQREM11
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   OUTPUT 709;"IMP 50"             !Set 50 ohm output imped
70   OUTPUT 709;"APPLY SQI 5E-3"     !Set output to 5 mA PP
80   OUTPUT 709;"OUTPUT?"            !Query AC output value
90   ENTER 709;A                     !Enter output value
100  PRINT "Output =";A;"Amps"       !Display output value
110  END
```

A typical controller CRT display (in Amps AC) follows. If the output is measured with an HP 3458 DMM, the result = .005 A/2.000 = 0.0025 A (2.5 mA), since this is a 50% duty cycle square waveform.

```
Output = .005 Amps
```

Example 11-6: Frequency-Sweep Sine Waveform (SWEEP11)

This program performs a 100 Hz to 100 kHz frequency-stepped sweep. The program outputs a 5 Vac PP sine waveform from channel A and runs through the frequency range in 100 Hz steps. The step rate is about 600 Hz. Refer to Chapter 14 - Advanced Programming Topics for additional sweep examples.

```
10   !file SWEEP11
20   !
30   CLEAR 709                                 !Clear HP 3245A
40   OUTPUT 709;"RST"                          !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                      !Clear HP 3245A memory
60   OUTPUT 709;"SUB SWEEP"                    !Define subroutine
70   OUTPUT 709;"  APPLY ACV 5"                !5 Vac PP sine wave
80   OUTPUT 709;"  FOR I = 100 to 100000 STEP 100"  !Sweep through range
90   OUTPUT 709;"    FREQ I"                   !Change frequency
100  OUTPUT 709;"  NEXT I"                     !Increment count
110  OUTPUT 709;"SUBEND"                       !End subroutine
120  OUTPUT 709;"CALL SWEEP"                   !Execute subroutine
130  END
```

Programming Triggered AC Waveforms

This section shows how to program the HP 3245A for triggered AC waveforms. As shown in Figure 11-13, there are four steps to select triggered AC waveform parameters:

- Select waveform function
- Select channel parameters.
- Select waveform parameters.
- Select trigger event/mode.

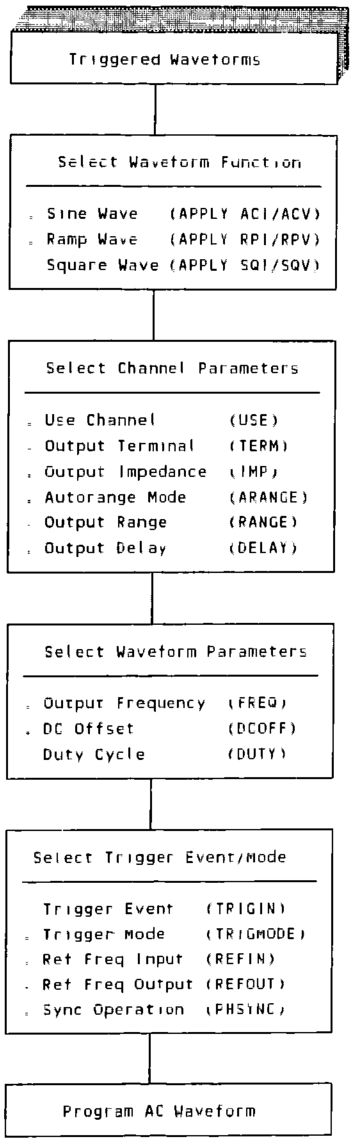


Figure 11-13. Triggered AC Waveforms - Selection Steps

Command Summary

Figure 11-14 summarizes channel operation and associated commands for triggered AC waveforms. The waveform function is selecting with an **APPLY** command.

Channel/Waveform Parameters

USE sets the **USE** channel (A or B) and **TERM** sets the output terminal (front or rear). **IMP** sets 0 Ω or 50 Ω output impedance, and **ARANGE** or **RANGE** sets a fixed range or autorange. **DELAY** sets the delay time (from 0 to 9.99 seconds) between output commands. The waveform frequency (from 0 Hz to 1 MHz) is set with **FREQ**, the DC offset is set with **DCOFF**, and (for ramp and square waves) **DUTY** sets the duty cycle (from 5% to 95%).

Trigger Event/Mode

Select one of three triggered modes using **TRIGMODE**: synchronized, gated, or dual-frequency. With synchronized mode, (**TRIGMODE ARMWF**) the waveform is not output until a trigger from the source set by **TRIGIN** occurs. With gated mode (**TRIGMODE GATE**), the waveform is output ONLY when the input trigger level is LOW (0V). With dual-frequency mode (**TRIGMODE DUALFR**), the waveform frequency can be varied depending on the input trigger level.

In synchronized mode (**TRIGMODE ARMWF**), **REFOUT** sets the reference frequency output destination and **REFIN** sets the reference frequency input source. The **PHSYNC** command can be used to set a specific reference frequency input/output and trigger source as well as to trigger the **USE** channel.

Selecting Waveform Function

As with modified AC waveforms, the first step to program triggered AC waveforms is to select the waveform function (sine, square, or arbitrary) required. Refer to "Programming Power-On Waveforms" for a discussion of AC waveform functions.

Selecting Channel/ Waveform Parameters

When you have selected the waveform function, select the channel parameter(s) (**USE** channel, output terminal, output impedance, autorange mode, output range, and/or output delay) to be modified. Then, select the waveform parameters (output frequency, DC offset, and duty cycle) required. Refer to "Programming Modified Waveforms" for details on selecting channel and waveform parameters.

Selecting Trigger Event/Mode

For triggered AC waveforms, use **TRIGIN** to select the trigger input event (source) and **TRIGMODE** to select the trigger mode. **TRIGMODE** selects one of three trigger modes: synchronized mode (**TRIGMODE ARMWF**); gated mode (**TRIGMODE GATE**); or dual-frequency mode (**TRIGMODE DUALFR**). Power-on/reset setting is **TRIGMODE OFF** and **TRIGMODE** has no effect on DC outputs.

With **TRIGMODE OFF**, waveforms are output immediately when an **APPLY** command is executed. When **TRIGMODE ARMWF** or **GATE** is set, the waveform is not generated until a trigger event specified by **TRIGIN** occurs and **TRIGMODE DUALFR** enables modulation between two frequencies.

Channel Parameters

Use Channel (USE)

USE ch sets use (output) channel
USE 0 or USE CHANA sets channel A
USE 100 or USE CHANB sets channel B

Output Terminal (TERM)

TERM [mode] sets the output terminal
TERM FRONT sets front panel output
TERM REAR sets rear panel output and
TERM OFF OPEN disconnects outputs

Output Impedance (IMP)

IMP mode sets the output impedance on the use channel. IMP 0 sets 0 Ω impedance. IMP 50 sets 50 Ω impedance

AC Range (RANGE/ARANGE)

ARANGE [mode] sets autorange mode and RANGE [max_output] sets fixed output range. AC voltage output resolution depends on the output impedance (set by IMP). RANGE AUTO sets autorange

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Delay range is from 0 to 9.99 seconds with 0.01 seconds resolution

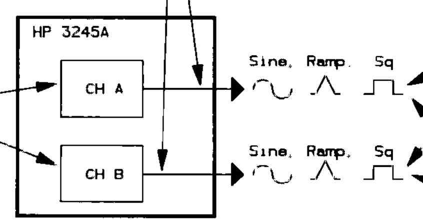
Trigger Event/Mode

Trigger Event (TRIGIN)

When TRIGMODE ARMWF or GATE is set TRIGIN [event] sets the trigger event

Trigger Mode (TRIGMODE)

The TRIGMODE command sets the trigger mode. TRIGMODE ARMWF sets synchronized mode. TRIGMODE GATE sets gated mode and TRIGMODE DUALFR sets dual-frequency mode



32450P F 11 14

Waveform Functions

Voltage Waveforms (APPLY volts)

APPLY ACV volts sets sine wave voltage waveform, APPLY RPV volts sets ramp wave voltage waveform and APPLY SQV volts sets square wave voltage waveform

Current Waveforms (APPLY amps)

APPLY ACI amps sets sine wave current waveform. APPLY RPI amps sets ramp wave current waveform and APPLY SQI amps sets square wave current waveform

Waveform Parameters

Waveform Frequency (FREQ)

FREQ freq1 [, freq2] sets waveform frequency (in Hz). Range is 0 to 1 MHz (ramp performance not specified above 100 kHz). Power-on frequency = 1000 Hz (freq2 applies to dual-freq mode only)

DC Offset (DCOFF)

DCOFF volts sets DC offset on use channel. Peak amplitude of waveform plus magnitude of DC offset must be ≤50% of selected AC range

Duty Cycle (DUTY)

DUTY %duty sets the duty cycle for AC ramp and square waves. Duty cycle range is 5% to 95% with power-on duty = 50%. For square waves 50% duty cycle only for frequencies above 100 kHz

Figure 11-14. Triggered AC Waveforms - Command Summary

NOTE

Refer to Chapter 13 - Triggering Outputs/Waveforms for a discussion of HP 3245A triggering techniques.

Trigger Event (TRIGIN)

Select the event to trigger the AC waveform with **TRIGIN** [*event*]. When **TRIGMODE OFF** is not set, AC waveform generation is controlled by the event specified by **TRIGIN** [*event*]. Table 11-4 shows **TRIGIN** [*event*] parameters. Power-on *event* = **HIGH**, default *event* = **SGL**.

Table 11-4. Trigger Event (TRIGIN) Parameters

event	Definition
TB0	Trigger Bus 0. Trigger using the signal on the TB0 trigger bus.
TB1	Trigger Bus 1. Trigger using the signal on the TB1 trigger bus.
EXT	Trigger Connector. Trigger using the input to the front panel Trigger connector.
EXTBAR	Inverse of EXT. Trigger using the inverse of input to front panel Trigger connector.
LOW	Low Signal - Software Control. Used with HIGH parameter to internally trigger.
HIGH	High Signal - Software Control. Used with LOW parameter to internally trigger.
HOLD	Same as HIGH parameter.
SGL	Single Trigger. Assure a HIGH then change to LOW and then to HIGH to internally trigger.

Synchronized Mode (TRIGMODE ARMWF)

Synchronized mode, set with **TRIGMODE ARMWF**, is convenient for generating multiphase AC waveforms and precise harmonics. As shown in Figure 11-15, in synchronized mode one channel is designated as a "master" and one or more other channels are designated as "slaves". Each channel (master or slave) generates a reference frequency (1073741.824 Hz) based on its internal reference oscillator.

In synchronized mode, each slave's oscillator locks on to the reference frequency generated by the master. When the master and the slaves are triggered (all channels must be triggered simultaneously), each channel begins outputting its waveform at the selected frequency and phase angle. At power-on/reset, channel B is slaved to channel A.

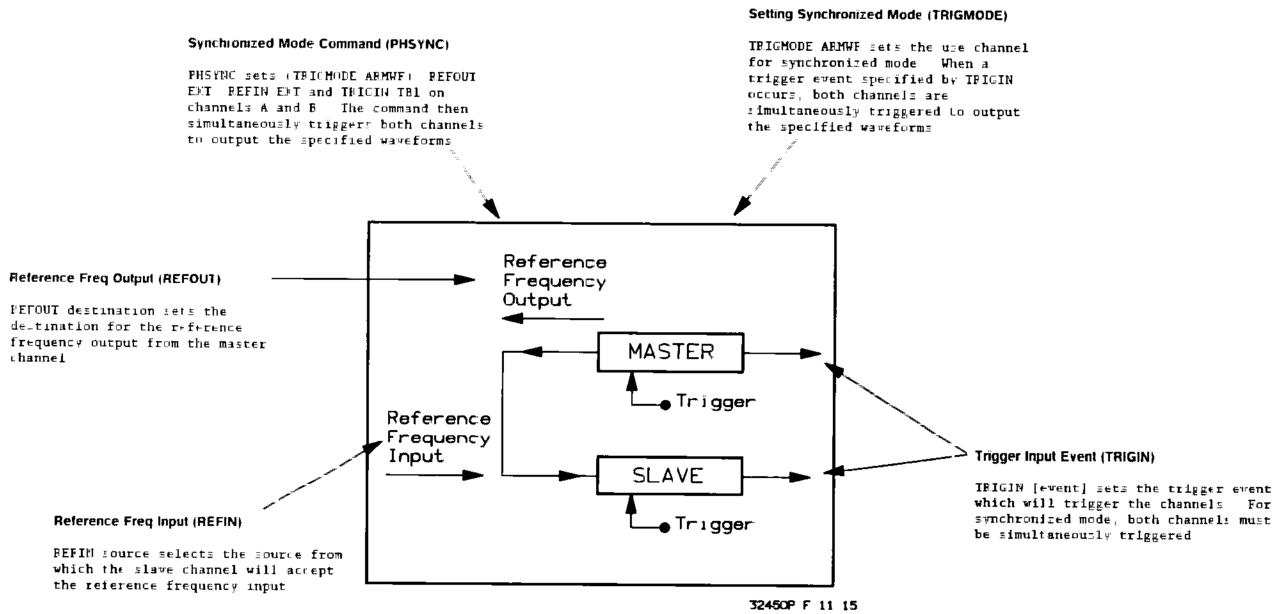


Figure 11-15. Triggered AC Waveforms - Synchronized Mode

Setting Reference Frequency Output Destination (REFOUT)

The reference frequency generated by the master channel can be routed to the slave channel via the master channel's Freq Ref connector (on the rear panel) or via the TB0 or TB1 trigger buses. The **REFOUT** *destination* command sets the output destination for the master channel. At power-on/reset, channel A is set to **REFOUT EXT; REFIN INT** and channel B is set to **REFOUT OFF; REFIN EXT**. Table 11-5 lists the **REFOUT** *destination* parameters. Use **REFOUT?** to query the **USE** channel reference output destination.

Table 11-5. Reference Frequency Output (REFOUT) Parameters

dest	Definition
OFF	Disable reference frequency output from master channel. Places FREQ REF connector in high-impedance state.
EXT	FREQ REF Connector. Output reference frequency to the FREQ REF connector.
TB0	Trigger Bus 0. Output reference frequency from master channel to trigger bus 0.
TB1	Trigger Bus 1. Output reference frequency from master channel to trigger bus 1.

NOTE

Since a channel cannot be a master and a slave at the same time, when **REFOUT** *is executed with a parameter other than* **OFF**, *that channel's reference frequency input source is automatically set to internal (***REFIN INT***). Or, when* **REFIN** *is executed with a parameter other than* **INT**, *that channel's reference frequency output destination is automatically disabled (***REFOUT OFF***).*

Setting Reference Frequency Input Source (REFIN)

As noted, **REFOUT** sets the destination for the reference frequency output by the master channel. **REFIN** *source* sets the source from which the slave channel(s) will accept the reference frequency from the master channel. At power-on/reset, channel A is set for **REFOUT EXT; REFIN INT**, while channel B is set for **REFOUT OFF; REFIN EXT**. With these settings, no external connection is required since the reference frequency is sensed internally by the slave channel. Table 11-6 shows the reference frequency input parameters.

Table 11-6. Reference Frequency Input (REFIN) Parameters

Command	Description
DRIVETB1 HIGH	Set TB1 HIGH (+5 V)
USE 0	Use channel A
TRIGMODE ARMWF	Set synchronized mode on ch A
REFOUT EXT	Ref freq dest is Freq Ref connector
TRIGIN TB1	Ch A input trigger source is TB1
USE 100	Use channel B
TRIGMODE ARMWF	Set synchronized mode on ch A
REFIN EXT	Ref freq input is Freq Ref connector
TRIGIN TB1	Ch B input trigger source is TB1
DRIVETB1 SGL	Trigger both channels simultaneously
Note: USE is returned to the channel specified when PHSYNC was executed.	

Changing Master/Slave Relationship (REFIN/REFOUT)

At power-on/reset, channel A is set to **REFOUT EXT; REFIN INT** and channel B is set to **REFOUT OFF;REFIN EXT**. This setting "slaves" channel B reference to channel A reference, making channel A the "master". However, when **REFOUT** is changed to **EXT** on a channel, it forces the other channel (if **REFOUT EXT** is set) to **REFOUT OFF;REFIN EXT**. That is, the master/slave relationship is reversed.

NOTE

When two HP 3245As have their Freq Ref terminals connected, at power-on they both drive the connector since **REFOUT EXT** *is selected at power-on. For proper operation, both channels on one of the HP 3245As should be set to* **REFIN EXT**.

Using Synchronized Mode Command (PHSYNC)

A convenient way to set input trigger source, reference frequency output destination, reference frequency input source, and to trigger the waveforms all with a single command is to use the **PHSYNC** command. With **PHSYNC**, channels A and B are set for synchronous operation (with **TRIGMODE ARMWF**), the reference frequency is routed from the master channel to the slave channel via the Freq Ref connector, and triggering is via the TB1 trigger bus.

Using **PHSYNC** is equivalent to setting channel A as the master channel, setting channel B as the slave channel, setting a defined set of parameters, and executing a trigger to output the waveforms synchronously. Note that channel and waveform parameters, such as amplitude, frequency, DC offset, etc. must still be set with the appropriate command. Table 11-7 lists commands set by executing **PHSYNC**.

NOTE

Be careful using **PHSYNC** *if you use two HP 3245As connected by their Freq Ref connectors, as there may be two masters attempting to drive the same Freq Ref connector.*

Table 11-7. Commands Set by PHSYNC

Command	Description
DRIVETB1 HIGH	Set TB1 level high (+5 V).
USE 0	Use channel A (master channel).
TRIGIN TB1	Ch A input trigger source is TB1.
TRIGMODE ARMWF	Set synchronized mode on ch A.
REFOUT EXT	Ref freq dest is FREQ REF connector.
USE 100	Use channel B (slave channel).
TRIGIN TB1	Ch B input trigger source is TB1.
TRIGMODE ARMWF	Set synchronized mode on ch B.
REFIN EXT	Ref freq input is FREQ REF connector.
DRIVETB1 SGL	Trigger both channels simultaneously. Returns to the USE channel in effect when executed.

Setting Phase Angle (PANG)

By using the **PANG** *degrees* command, you can set the phase angle between the master channel waveform and the slave channel(s) waveform. At power-on/reset, the phase angle is 0 degrees. In the **PANG** command, *degrees* is the phase angle between -360 degrees and +360 degrees with 0.001 degrees resolution.

Zero degrees is defined as the positive-going, zero-crossing point on the sine, ramp, or square waveform (relative to the DC offset). (For arbitrary waveforms, zero degrees is defined as the first of the 2048 points required to define the waveform. One complete waveform equals 360 degrees.)

To guarantee that **PANG** will produce the desired waveform value prior to triggering, set the phase angle to at least 10 degrees from the zero-crossing phase for sine and square waveforms. For example, **PANG 0** on a sine waveform may produce a low or a high Sync signal. However, **PANG -10** always produces a low Sync signal while **PANG 10** always produces a high Sync signal. Waveforms with lower zero-crossing slew rates will require larger phase correction

Enabling Synchronized Mode (TRIGMODE/APPLY)

When you have selected the master channel reference frequency output destination (**REFOUT** command), the slave channel reference frequency input source (**REFIN** command), and the phase angle (**PANG** command), you must enable the synchronized mode with the **TRIGMODE ARMWF** command.

When synchronized mode has been enabled, use an **APPLY** command to set the waveform function (sine, ramp, square, or arbitrary) on each channel and its assigned frequency (**FREQ**) and phase angle (**PANG**). When the master and slaves are simultaneously triggered by a trigger from the **TRIGIN** source, each channel begins outputting its defined waveform. Note that all channels MUST be triggered at the same time.

In synchronized mode, the **APPLY** commands, **ARANGE ON**, **DCOFF**, **DUTY**, **FREQ**, **IMP**, **PANG**, **RANGE**, and **TRIGMODE ARMWF** reset the waveforms from the master and the slaves to their starting phase. After one of these commands is executed, the waveform returns to its assigned phase angle and waits for another trigger before resuming. (Sending an **ARANGE**, **IMP**, or **RANGE** command to the same range will not reset the waveforms to their starting phase.)

Figure 11-16 shows an example of synchronized mode operation. Channel A (the master channel) outputs a 5 Vac PP sine waveform at phase angle 0°, while channel B outputs a 5 Vac PP ramp waveform at phase angle 180°. Both waveforms have output frequency of 5000 Hz.

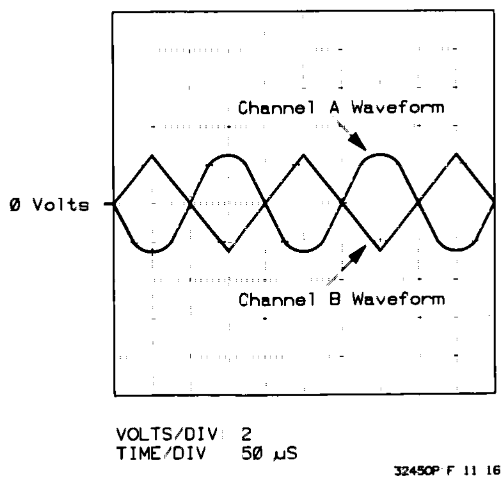


Figure 11-16. Synchronized Mode - Typical Waveform

Gated Mode (TRIGMODE GATE)

AC waveforms can be gated by using the **TRIGMODE GATE** command. Gated mode is convenient for generating waveforms which must start and stop at precisely the zero phase point. With gated mode, when the signal on the trigger event selected by **TRIGIN** is low (0 V), the channel generates the waveform specified by **APPLY**. When the trigger signal goes high (+5 V), the waveform output continues until it reaches its zero phase point and then stops.

Zero phase is defined as the positive-going, zero-crossing point of a sine, ramp, or square waveform (relative to the DC offset). For arbitrary waveforms, zero degrees is defined as the first of the 2048 points required to define the waveform. At power-on, gated mode is disabled (**TRIGMODE OFF**) and the waveform is continuously generated (no trigger is required).

Figure 11-17 shows an example of a ramp waveform gated by a trigger signal. When the signal is high (+5 V), there is no output. When the trigger signal goes low (0 V), the ramp waveform is output. When the trigger signal returns high, the ramp waveform continues until it reaches its zero phase point and then stops. Refer to "Programming Triggered Waveforms" for example gated waveform programs.

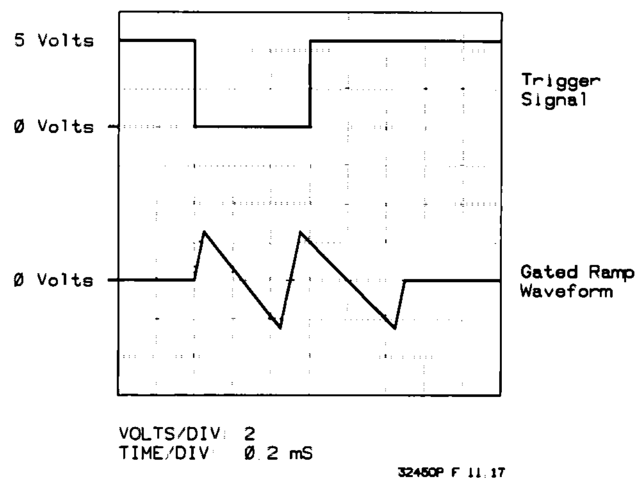


Figure 11-17. Gated Mode - Typical Waveform

Dual-Frequency Mode (TRIGMODE DUALFR)

You can use **TRIGMODE DUALFR** to enable modulation of AC waveforms between two frequencies. This mode is convenient to generate precise frequency changes or to hold a waveform at a specified frequency determined by the logic level of the selected trigger event. The dual-frequency mode is enabled with **TRIGMODE DUALFR**.

The two waveform frequencies are set by the **FREQ** *freq_1* [,*freq_2*] command, where *freq_1* is the output frequency when the event selected by **TRIGIN** is high (+5 V) and *freq_2* is the output frequency when the trigger event is low (0 V). Refer to Chapter 14 - Advanced Programming Topics for frequency hopping (modulation of more than two frequencies) techniques.

For sine, square, and arbitrary waveforms, both frequencies must be in the range 0 to 1 MHz. For ramp waveforms, both frequencies must be in the range 0 to 100 kHz. At power-on, both frequencies are set to 1000 Hz. Figure 11-18 shows an example of dual-frequency mode operation. When the trigger level is high, the output is 50 Hz and when the trigger level is low, the output is 500 Hz.

NOTE

Executing **APPLY** *commands*, **ARANGE ON, DCOFF, DUTY, FREQ, IMP, RANGE,** *or* **TRIGMODE** *may cause the output waveform to momentarily change to the other of the two frequencies.*

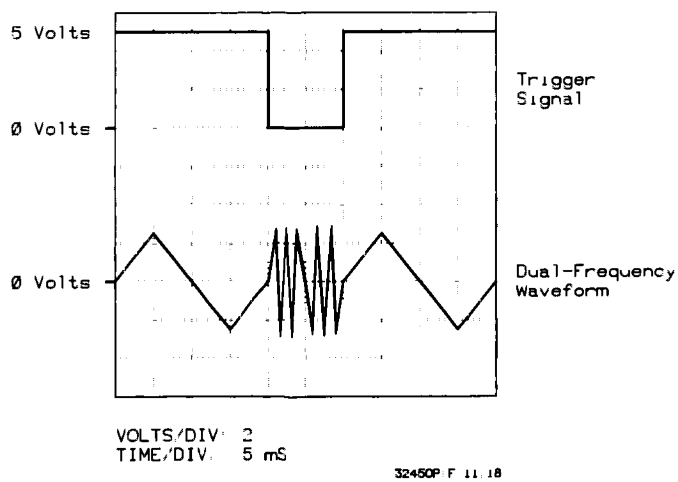


Figure 11-18. Dual-Frequency Mode - Typical Waveform

Programming Examples

When you have selected the waveform function, channel/waveform parameters, and trigger event/mode required, program the HP 3245A for the desired waveform. Figure 11-19 shows a suggested sequence of commands to program triggered AC waveforms. Note that the programming sequence shown in Figure 11-19 is NOT the same as the steps to select the waveform function, channel/waveform parameters, and trigger event/mode shown in Figure 11-13. Example programs follow to generate triggered AC waveforms.

Example 11-7: Triggered Ramp Waveform (RAMPT11)

This program outputs a voltage ramp waveform @ 3.0 Vac PP and 2000 Hz from channel A. The duty cycle is set for 30% and the DC offset for 1.0 Volts. The waveform is not output until an external (high-to-low) trigger is input to the channel A Trigger port.

This program requires a BNC connector to the channel A Trigger connector and a TTL-compatible input trigger. When a trigger is input, the ramp waveform is generated. Note that **TRIGMODE ARMWF** sets channel A so that the output is not generated until the trigger is received even though **APPLY RPV** is executed.

```
10   !file RAMPT11
20   !
30   CLEAR 709                     !Clear HP 3245A
40   OUTPUT 709;"RST"              !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"          !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE ARMWF"   !Set trigger armed mode
70   OUTPUT 709;"FREQ 2000"        !Set output freq to 2000 Hz
80   OUTPUT 709;"DUTY 30"          !Set 30% duty cycle
90   OUTPUT 709;"DCOFF 1.0"        !Set DC offset of 1.0V
100  OUTPUT 709;"APPLY RPV 3.0"    !Output ramp wave @3.0 Vac PP
110  OUTPUT 709;"TRIGIN EXT"       !External trigger channel A
120  END
```

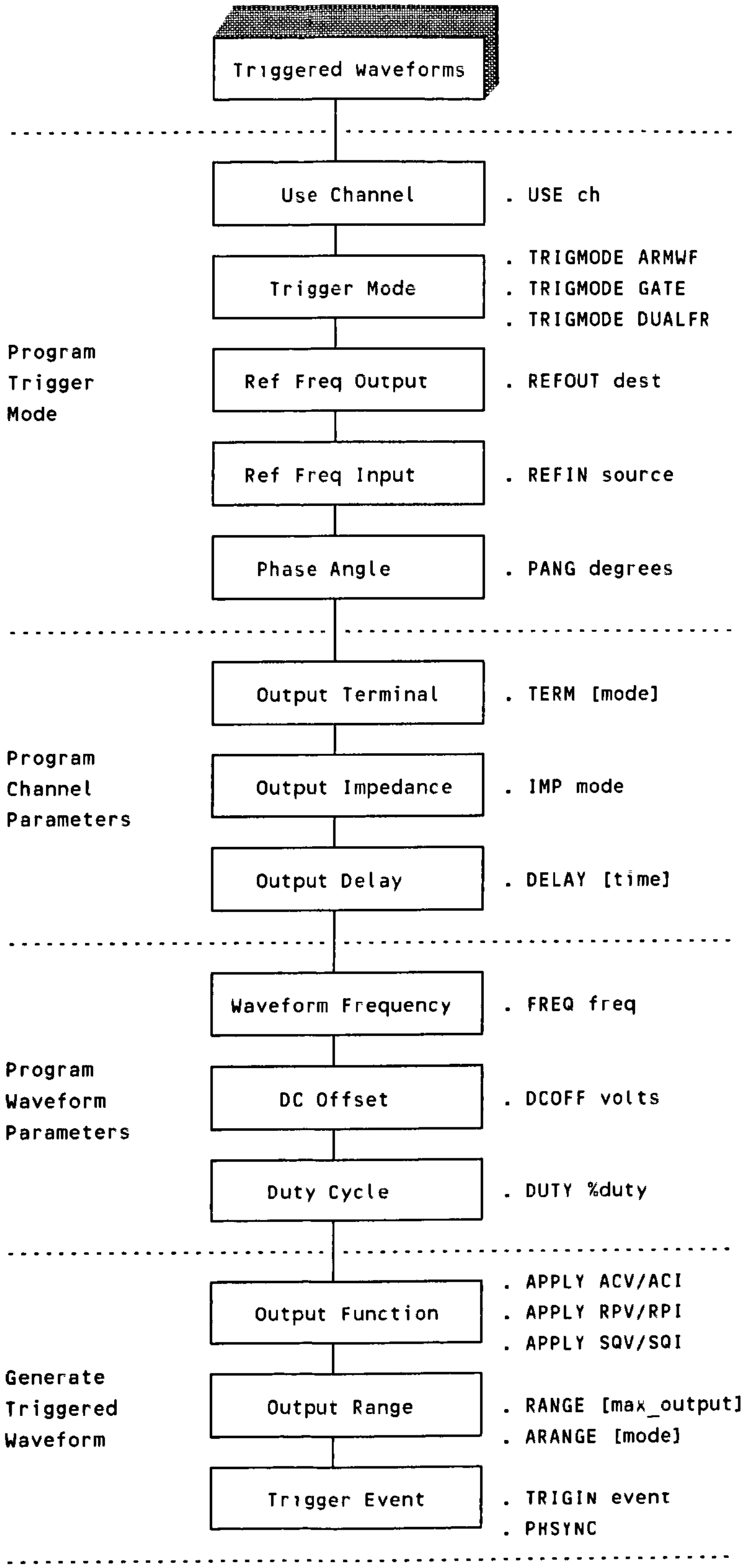


Figure 11-19. Triggered AC Waveforms - Programming Sequence

Example 11-8: Sync Waveform - Using PHSYNC (SYNCP11)

This program outputs a 5 kHz sine waveform (phase = 0°) from the channel A (the master channel) and a 5 kHz ramp waveform (phase = 180°) from channel B (the slave channel). This example is similar to the previous example except that **PHSYNC** is used to set the input trigger source, reference frequency destination/source, and to trigger the waveforms.

PHSYNC sets synchronized waveform mode, sets TB1 as the channel input trigger sources, sets the Freq Ref connector as the output/input path for the reference frequency, and simultaneously triggers both channels (via TB1) to output their waveforms (refer to Table 11-7). See Figure 11-16 for a typical waveform display.

```
10   !file SYNCP11
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   !
70   !Configure channel A
80   !
90   OUTPUT 709;"USE 0"              !Use channel A
100  OUTPUT 709;"FREQ 5000"          !Output freq is 5 kHz
110  OUTPUT 709;"APPLY ACV 5"        !Output sine wave @ 5 Vac PP
120  !
130  !Configure channel B
140  !
150  OUTPUT 709;"USE 100"            !Use channel B
160  OUTPUT 709;"FREQ 10E3"          !Output freq is 10 kHz
170  OUTPUT 709;"PANG 180"           !Phase angle = 180 degrees
180  OUTPUT 709;"APPLY RPV 5"        !Output ramp wave @ 5 Vac PP
190  !
200  !Generate synchronized outputs
210  !
220  OUTPUT 709;"PHSYNC"             !Generate sync outputs
230  END
```

Example 11-9: Sync Waveform - Using REFIN (SYNCR11)

This program outputs a 5 kHz sine waveform (phase = 0°) from the channel A (master channel) and a 5 kHz ramp waveform (phase = 180°) from channel B (slave channel). The master channel is triggered using the **TRIGIN** command HIGH and LOW parameters. Since **TRIGOUT EXT** is set, Channel A routes its trigger signal to the channel A Trigger (I/O) port.

To trigger channel B, connect a BNC connector between the channel A and channel B Trigger (I/O) ports. Then, channel A and channel B are simultaneously triggered when **TRIGIN LOW** is executed. Since **REFOUT EXT** and **REFIN EXT** are set, the reference frequency from channel A is automatically sensed by channel B and no external connections are required for the reference frequency. Figure 11-16 shows a typical oscilloscope display of the output.

```
10   !file SYNCR11
20   !
30   CLEAR 709                     !Clear HP 3245A
40   OUTPUT 709;"RST"              !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"          !Clear HP 3245A memory
60   !
70   !Configure channel A (master)
80   !
90   OUTPUT 709;"USE 0"            !Use channel A
100  OUTPUT 709;"TRIGIN HIGH"      !Set trigger level high (+5V)
110  OUTPUT 709;"REFOUT EXT"       !REF FREQ connector is dest
120  OUTPUT 709;"TRIGMODE ARMWF"   !Set sync mode on channel A
130  OUTPUT 709;"TRIGOUT EXT"      !Enable ch A Trigger output
140  OUTPUT 709;"FREQ 5000"        !Output freq is 5 kHz
150  OUTPUT 709;"PANG 0"           !Phase angle = 0 degrees
160  OUTPUT 709;"APPLY ACV 5"      !Output sine wave @ 5 Vac PP
170  !
180  !Configure channel B (slave)
190  !
200  OUTPUT 709;"USE 100"          !Use channel B
210  OUTPUT 709;"REFIN EXT"        !REF FREQ connector is source
220  OUTPUT 709;"TRIGMODE ARMWF"   !Set sync mode on channel B
230  OUTPUT 709;"TRIGIN EXT"       !Trigger conn is trigger source
240  OUTPUT 709;"FREQ 5E3"         !Output freq is 5 kHz
250  OUTPUT 709;"PANG 180"         !Phase angle = 180 degrees
260  OUTPUT 709;"APPLY RPV 5"      !Output ramp wave @ 5 Vac PP
270  !
280  !Generate synchronized outputs
290  !
300  OUTPUT 709;"USE 0"            !Use channel A
310  OUTPUT 709;"TRIGIN LOW"       !High-to-low trigger
320  END
```

Example 11-10: Four-Channel Sync Waveforms (SYNCF11)

This program sets four 3245A channels so that they can be triggered simultaneously. One channel is configured as a "master" channel and three channels are configured as slave channels. The reference frequency of the master channel is routed to each slave. Routing is from the Freq Ref BNC connector on the "master" HP 3245A (address 709) to the Freq Ref BNC connector on the "slave" HP 3245A (address 717).

Channels are triggered using trigger bus 0 (TB0) on each HP 3245A. By connecting TB0 on the slave channel to TB0 on the master channel then driving a signal to the bus, the signal is routed to all channels, thus triggering them simultaneously.

```
10   !file SYNCF11
20   !
30   CLEAR 717                    !Clear HP 3245A at address 17
40   OUTPUT 717;"RST"             !Reset HP 3245A at address 17
50   OUTPUT 717;"SCRATCH"         !Clear HP 3245A memory at address 17
60   !
70   CLEAR 709                    !Clear HP 3245A at address 9
80   OUTPUT 709;"RST"             !Reset HP 3245A at address 9
90   OUTPUT 709;"SCRATCH"         !Clear HP 3245A memory at address 9
100  !
110  OUTPUT 717;"USE 0"           !Use "slave" channel A (address 17)
120  OUTPUT 717;"  DRIVETB0 OFF"  !Set TB0 as input source
130  OUTPUT 717;"  TRIGMODE ARMWF" !Arm channel A  (address 17)
140  OUTPUT 717;"  TRIGIN TB0"    !Set channel A trigger source to TB0
150  OUTPUT 717;"  REFIN EXT"     !Set freq ref to master's reference
160  OUTPUT 717;"  APPLY RPV 2.5" !Apply 2.5 Vac PP ramp when triggered
170  !
180  OUTPUT 717;"USE 100"         !Use "slave" channel B (address 17)
190  OUTPUT 717;"  TRIGMODE ARMWF" !Arm channel B (address 17)
200  OUTPUT 717;"  TRIGIN TB0"    !Set channel B trigger source to TB0
210  OUTPUT 717;"  REFIN EXT"     !Set freq ref to master's reference
220  OUTPUT 717;"  APPLY RPV 2.5" !Apply 2.5 Vac PP ramp when triggered
230  !
240  OUTPUT 709;"USE 100"         !Use "slave" channel B (address 9)
250  OUTPUT 709;"  TRIGMODE ARMWF" !Arm channel B (address 9)
260  OUTPUT 709;"  TRIGIN TB0"    !Set channel trigger source to TB0
270  OUTPUT 709;"  REFIN EXT"     !Set freq ref to master's reference
280  OUTPUT 709;"  APPLY RPV 2.5" !Apply 2.5 Vac PP ramp when triggered
290  !
300  OUTPUT 709;"USE 0"           !Use "master" channel A (address 9)
310  OUTPUT 709;"  TRIGMODE ARMWF" !Arm channel A (address 9)
320  OUTPUT 709;"  TRIGIN TB0"    !Set channel trigger source to TB0
330  OUTPUT 709;"  REFOUT EXT"    !Route master freq ref to slaves
340  OUTPUT 709;"  APPLY RPV 2.5" !Apply 2.5 Vac PP ramp when triggered
350  OUTPUT 709;"  DRIVETB0 SGL"  !Trigger all channels simultaneously
360  END
```

Example 11-11: Gated Ramp Waveform (RAMPG11)

This program enables gated waveform mode on the channel A. When the trigger signal to the channel A Trigger (I/O) connector is low (0 V), the channel outputs a 2 kHz ramp waveform with a 20% duty cycle. When the trigger input is HIGH (+5 V), the ramp waveform continues until it reaches its zero phase point and then stops. For this program, connect an external trigger source to the channel A front panel Trigger (I/O) connector. See Figure 11-17 for a typical output display.

```
10   !file RAMPG11
20   !
30   CLEAR 709                        !Clear HP 3245A
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Clear HP 3245A memory
60   OUTPUT 709;"TRIGMODE GATE"       !Set gated waveform mode
70   OUTPUT 709;"TRIGIN EXT"          !Trigger conn is trig source
80   OUTPUT 709;"FREQ 2000"           !Output frequency is 2 kHz
90   OUTPUT 709;"DUTY 20"             !20% duty cycle
100  OUTPUT 709;"APPLY RPV 5"         !Output ramp waveform @ 5 Vac PP
110  END
```

Example 11-12: Dual-Freq Wave - Using Ext Trig (DUALE11)

This program enables channel A to output a dual-frequency ramp waveform with an amplitude of 5 Vac PP. The output frequency is 50 Hz when the input to the channel A Trigger (I/O) connector is high (+5V) and 500 Hz when the signal is low (0V). For this program, connect an external trigger source to the channel A Trigger (I/O) connector. See Figure 11-18 for a typical display.

```
10   !file DUALE11
20   !
30   CLEAR 709
40   OUTPUT 709;"RST"                 !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"             !Delete vars, arrays, subs
60   OUTPUT 709;"TRIGMODE DUALFR"     !Set dual frequency mode
70   OUTPUT 709;"FREQ 50,500"         !Output 50 Hz or 500 Hz
80   OUTPUT 709;"TRIGIN EXT"          !Trigger conn is trig source
90   OUTPUT 709;"APPLY RPV 5"         !Output ramp waveform @ 5 Vac PP
100  END
```

Example 11-13: Dual-Freq Wave - Using Sync Trig (DUALS11)

This program outputs a sine wave on channel B at two different frequencies depending on the level of the trigger signal. The trigger signal is the Sync signal from channel A routed along trigger bus TB0. The Sync signal is a logic "1" (+5V) while the signal output from channel A is positive relative to its DC offset.

The signal is a logic "0" (0V) when the waveform output from channel A is negative relative to its DC offset. Thus, the frequency of the sine wave output on channel B is 5 kHz while the Sync signal is a logic "1" and 50 kHz when the Sync signal is a logic "0". See Figure 11-18 for a typical output

```
10    !file DUALS11
20    !
30    CLEAR 709                    !Clear HP 3245A
40    OUTPUT 709;"RST"             !Reset HP 3245A
50    OUTPUT 709;"SCRATCH"         !Clear HP 3245A memory
60    OUTPUT 709;"USE 0"           !Use channel A as source channel
70    OUTPUT 709;"  SYNCOUT TB0"   !Sync trig via TB0 to ch B
80    OUTPUT 709;"  APPLY ACV 1"   !Apply 1 Vac PP sine wave
90    OUTPUT 709;"USE 100"         !Use channel B as output channel
100   OUTPUT 709;"  TRIGMODE DUALFR" !Set dual-freq trigger mode
110   OUTPUT 709;"  FREQ 5E3, 50E3"  !Set 5 kHz, 50 kHz
120   OUTPUT 709;"  APPLY ACV 1"   !Output 1 Vac PP sine wave @ two freq
130   OUTPUT 709;"  TRIGIN TB0"    !Set TB0 as trigger source
140   END
```

Contents

Chapter 12
Programming Arbitrary Waveforms

Chapter 12

Programming Arbitrary Waveforms

Introduction

This chapter shows how to program the HP 3245A for arbitrary (user-defined) AC voltage and current waveforms. Refer to the following chapters to generate DC outputs or defined (sine, ramp, and square wave) AC waveforms.

- Chapter 10 - Generating DC Outputs
- Chapter 11 - Programming Defined Waveforms
- Chapter 13 - Triggering Outputs/Waveforms
- Chapter 14 - Advanced Programming Topics

Chapter Contents

The chapter contents are:

- **Programming User-Defined Waveforms** shows how to program the HP 3245A for arbitrary (user-defined) AC voltage and current waveforms when precomputing and triggering are not used.

- **Programming Precomputed Waveforms** shows how to program the HP 3245A for arbitrary waveforms when precomputing is used to reduce the time required to output the waveform.

- **Programming Triggered Waveforms** shows how to program the HP 3245A for arbitrary waveforms when triggering is used

---

NOTE

All programs in this chapter are stored on the discs titled Example Programs, Chapters 2 - 13. Refer to Chapter 1 - Using This Manual for details on using the disc. Refer to the HP 3245A Command Reference Manual for command details. If you are using the front panel keyboard to generate arbitrary waveforms, refer to Chapter 6 - Front Panel Operation.

Arbitrary Waveforms Overview

Figure 12-1 shows the three types of arbitrary waveforms discussed in this chapter (user-defined, precomputed, and triggered) shows a suggested set of steps to select the waveform function, define array values, select channel and waveform parameters, and select the trigger mode/event.

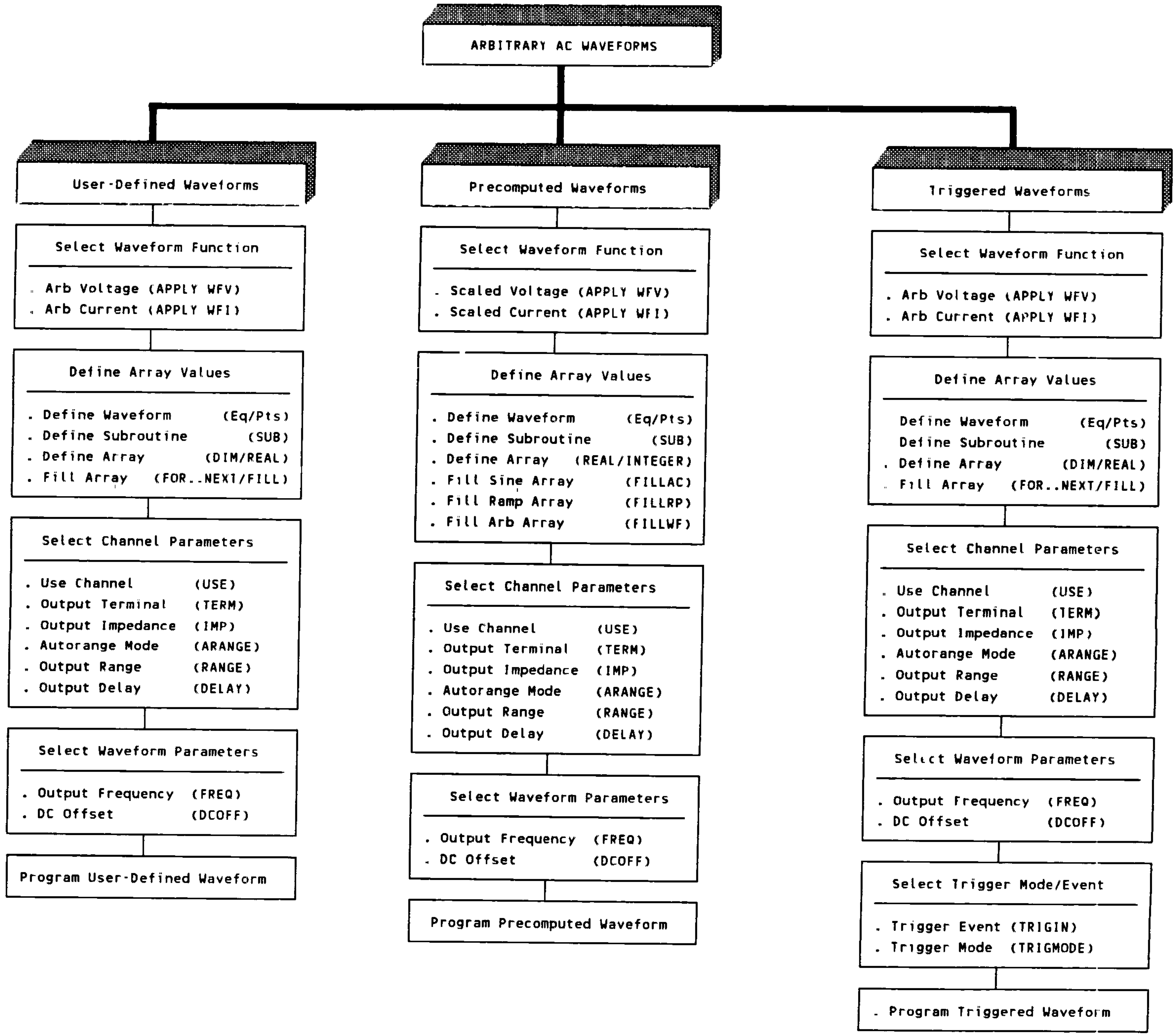


Figure 12-1. Arbitrary Waveforms - Overview

Programming User-Defined Waveforms

This section show how to program user-defined (arbitrary) voltage or current AC waveforms when precomputing and triggering are not used. As shown in Figure 12-2, there are four steps to program the HP 3245A for user-defined AC waveforms:

- Select waveform function.
- Define array values.
- Select channel parameters.
- Select waveform parameters

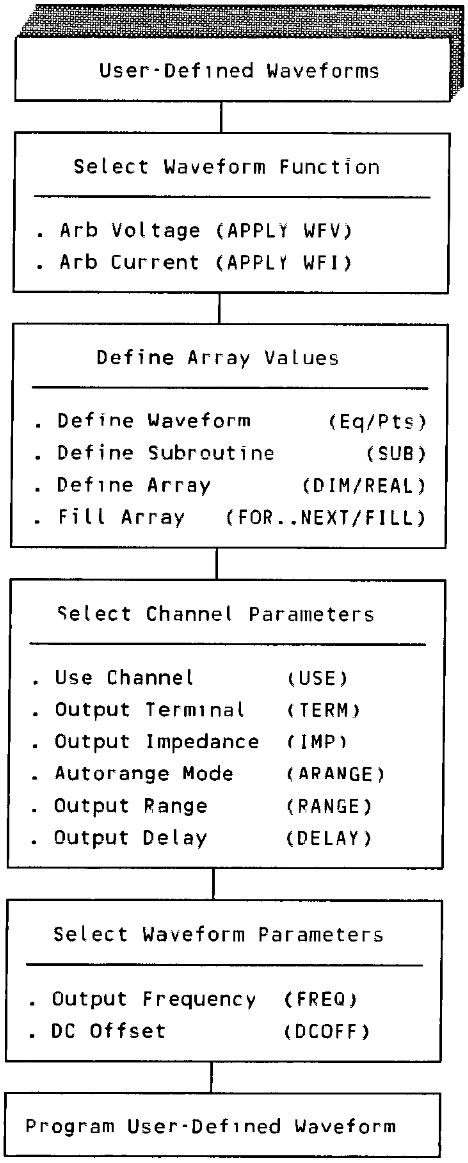


Figure 12-2. User-Defined Waveforms - Selection Steps

Command Summary

Figure 12-3 summarizes commands for user-defined arbitrary waveforms. **APPLY WFV** sets voltage outputs, while **APPLY WFI** sets current outputs

Array Values

To generate arbitrary waveforms, the (user-defined) waveform points must first be stored in an array. Then, when **APPLY WFV** or **APPLY WFI** is executed, the array values are used to generate the output levels. The storage array is defined with a **REAL**, **DIM**, or an **INTEGER** command. Each REAL array must have 2048 elements.

For user-defined waveforms, you can fill the array by using a subroutine (defined with **SUB**) or by using the **FILL** command. If you use a mathematical function [such as f(t) = sin(x)/x], use a subroutine to generate the array values If the waveform cannot be described mathematically, use the **FILL** command to specify each element value Each element value must be between -1 and +1, inclusive.

NOTE

The front panel **DIM Array** *and* **Edit Array** *keys can also be used to define and fill arrays A0 through A9. Refer to Chapter 6 - Front Panel Operation for details. If your waveform has less than 2048 points, refer to Chapter 14 - Advanced Programming Topics for methods of filling defined arrays.*

Channel/Waveform Parameters

There are five channel parameters which can be selected: use channel, output terminal, output impedance, range, and output delay. **USE** sets the output channel (A or B) and **TERM** sets the output terminal (front or rear). **IMP** sets 0 Ω or 50 Ω output impedance, and **RANGE** or **ARANGE** sets a fixed range or autorange. **DELAY** sets the delay time (from 0 to 9.99 seconds) between output commands Two waveform parameters can be set for arbitrary waveforms: waveform frequency and DC offset. The waveform frequency (from 0 Hz to 1 MHz) is set with **FREQ** and the DC offset is set with **DCOFF**.

Selecting Waveform Function

The **APPLY WFV** *pp_amplitude* [,*array_name*] command generates an arbitrary voltage waveform. The *pp_amplitude* parameter is the peak-to-peak amplitude of the output between 0.03125 Vac PP and 20 Vac PP in the 0 Ω output mode or between 0.015625 Vac PP and 10 Vac PP in the 50 Ω mode (when terminated with a 50 Ω load). Output voltage limits are 10% to 100% of the selected range.

APPLY WFI *pp_amplitude* [,*array_name*] generates an arbitrary current waveform. The *pp_amplitude* parameter is the peak-to-peak amplitude of the output between 0.00001 Amps AC PP and 0.2 Amps AC PP. Output current limits are 5% to 100% of the selected range.

Array Values

Define Array (REAL/INTEGER)

DIM or REAL defines a REAL array (must be 2048 elements) to store waveform values. INTEGER defines an INTEGER array (must be 2048 elements) to store values. REAL array values must be in range -1 to +1.

Fill Array (SUB/FILL)

Use SUB (with FOR ... NEXT loop) to fill a defined array for a mathematical function. Use FILL array_name, list to fill a defined array for a non-mathematical (user-specified) set of values. Can also fill array using front panel Edit Array key.

Waveform Function (APPLY WFV/WFI)

APPLY WFV or APPLY WFI pp_amplitude [,array_name] generates an arbitrary voltage or current waveform with peak-to-peak amplitude set by pp_amplitude, using values stored in array specified by [array_name].

Waveform values are determined by multiplying each value in the array with the peak value of the peak-to-peak amplitude.

Channel Parameters

Use Channel (USE)

USE ch sets use (output) channel. USE 0 or USE CHANA sets channel A; USE 100 or USE CHANB sets channel B.

Output Terminal (TERM)

TERM [mode] sets the output terminal. TERM FRONT sets front panel output, TERM REAR sets rear panel output, TERM OFF/OPEN disconnects outputs.

Output Impedance (IMP)

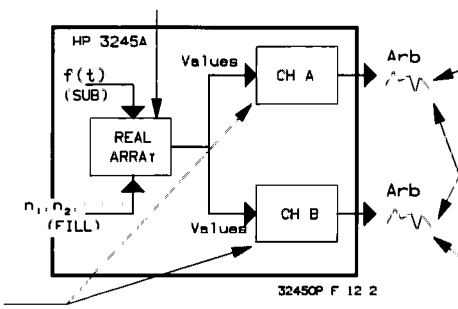
IMP mode sets the output impedance on the USE channel. IMP 0 sets 0Ω impedance. IMP 50 sets 50Ω impedance.

AC Range (RANGE or ARANGE)

RANGE [max_output] sets fixed output range. AC voltage output resolution depends on the output impedance (set by IMP). RANGE AUTO or ARANGE sets autorange.

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Power-on delay time = 0.04 sec. Delay range is from 0 to 9.99 seconds with 0.01 second resolution.



Waveform Parameters

Waveform Frequency (FREQ)

FREQ freq1 [, freq2] sets waveform frequency (in Hz). Range is 0 to 1 MHz (ramp performance not specified above 100 kHz). Power-on frequency = 1000 Hz. (freq2 applies to dual-freq mode only.)

DC Offset (DCOFF)

DCOFF volts sets DC offset on use channel. Peak amplitude of waveform plus magnitude of DC offset must be ≤50% of selected AC range.

Figure 12-3. User-Defined Waveforms - Command Summary

For both **APPLY WFV** and **APPLY WFI**, the *array_name* parameter specifies the name of a REAL or INTEGER array containing 2048 points which define the waveform. Each REAL array element must have a value between -1 and +1, inclusive. The INTEGER array will be discussed in "Programming Precomputed Waveforms".

The output waveform amplitude is determined by multiplying the peak value of the peak-to-peak amplitude set by **APPLY WFV** or **APPLY WFI** by the value in each array element. For example, if the element values of array A are -0.05, 0.9, 0.2, 0,...,0.5 **APPLY WFV 4,A** generates a voltage waveform with point values (PP) of -0.1 Vac, 1.8 Vac, 0.4 Vac, 0 Vac, ..., 1.0 Vac.

Since the **APPLY** command *array_name* is optional, you can change the waveform peak-to-peak amplitude without reloading the 2048 points. For example, to change the peak value of the output amplitude in the previous example to 1.5 Vac PP, use **APPLY WFV 3.0** and use the existing values in array A.

Defining Array Values

As noted, the **APPLY WFV** or **APPLY DCI** command generates arbitrary waveforms using 2048 points which were previously defined and stored into an array. Each point is sampled at 4294967.296 Hz and then low-pass filtered at 1.25 MHz. This technique does not guarantee that each point is held the same length of time, but does enable instantaneous change of the waveform frequency using the **FREQ** command. There are two steps to define array values: first, define the array and then fill the array.

Defining the Array (DIM/REAL/INTEGER)

Arrays are defined with an **DIM**, **REAL**, or **INTEGER** command. **DIM** *array_name (max_index)* [,*array_name (max_index)*, ...] or **REAL** *name* [(*max_index*)] [,*name* [(*max_index*)], ...] define REAL arrays, while **INTEGER** *name* [(*max_index*)] [,*name* [(*max_index*)], ...] defines an INTEGER array.

Each REAL or INTEGER array must have 2048 elements and each REAL array element must have a value between -1 and +1, inclusive. For example, REAL A(2047) defines REAL array A of 2048 elements, while INTEGER IA(2047) defines an INTEGER array IA of 2048 elements.

NOTE

If array_name is defined as an INTEGER array, each element must first be converted to a hardware integer format (with a **FILLAC**, **FILLRP**, *or* **FILLWF** *command) before executing an* **APPLY WFV** *or* **APPLY WFI** *command. Refer to "Programming Precomputed Waveforms" for details.*

Filling the Array (SUB/FILL)

For user-defined waveforms, you can fill the defined array in one of two ways: use a subroutine or use the **FILL** command. For a mathematically-defined function, such as f(t) = sin(x)/x you can use the equation in a subroutine (in a **FOR..NEXT** loop) to fill each array element.

For a function which cannot be described mathematically, use the **FILL** *array_name, list* command to store the (user-defined) values in an array. For either method of entry, the array values must be between -1 and +1, inclusive. For example, if A is defined as a 2048-element REAL array, **FILL A,-0.05,0.9,0.2** places -0.05 in the first element, 0.9 in the second, and 0.2 in the third element (elements four through 2048 are filled with zeroes).

Selecting Channel/ Waveform Parameters

As with defined AC waveforms, you can select five channel parameters: use channel (**USE**), output terminal (**TERM**), output impedance (**IMP**), AC range (**RANGE** or **ARANGE**), and output delay (**DELAY**). In addition, you can set the waveform frequency (**FREQ**) and the DC offset (**DCOFF**). Refer to Chapter 11 - Programming Defined Waveforms for details.

Programming Examples

When you have selected the waveform function, defined the array values, and selected the channel and waveform parameters required, program the HP 3245A for the desired waveform. Figure 12-4 shows a suggested sequence of commands to program the **USE** channel for user-defined waveforms.

Note that the programming sequence shown in Figure 12-4 is NOT the same as the sequence of steps shown in Figure 12-2 to select the parameters required. Two example programs to generate an arbitrary waveform follow. Refer to Chapter 5 - Arbitrary Waveform Programs for additional programming examples.

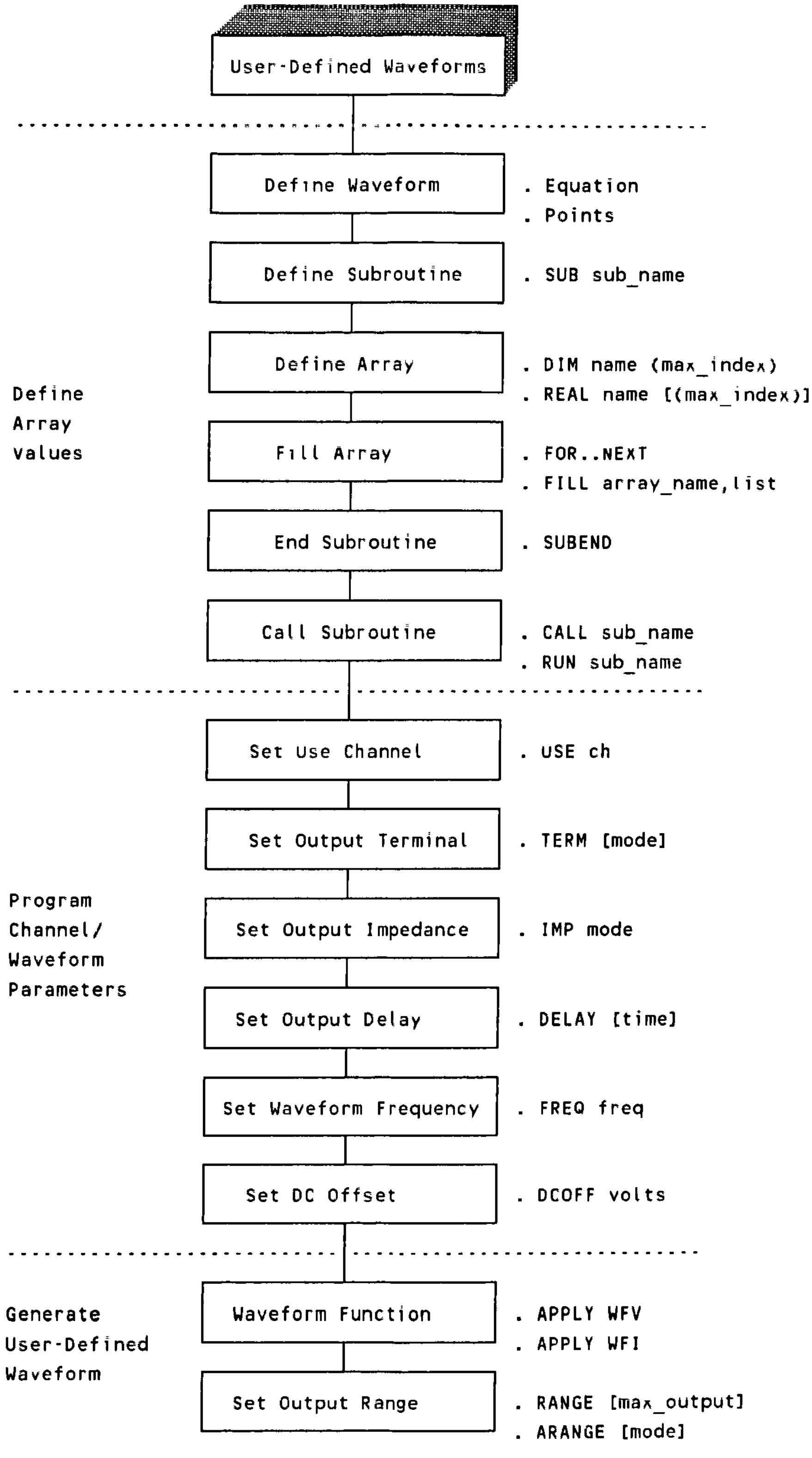


Figure 12-4. User-Defined Waveforms - Programming Sequence

Example 12-1: Arbitrary Waveform Using Sub (COSWF12)

This program defines and outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A.

```
10   !file COSWF12
20   !
30   CLEAR 709                                       !Clear HP 3245A
40   OUTPUT 709;"RST"                                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                            !Clear HP 3245A memory
60   OUTPUT 709;"DIM CARRAY(2047)"                   !Dim REAL array
70   OUTPUT 709;"REAL PI"                            !Define REAL var
80   OUTPUT 709;"PI = 3.1415925"                     !Assign value to PI
90   OUTPUT 709;"SUB DEF_ARRAY"                      !Begin subroutine
100  OUTPUT 709;"  FOR I = 0 TO 2047"                !Begin counting loop
110  OUTPUT 709;"    CARRAY(I) = COS(2*PI*I/2048)"   !Compute values
120  OUTPUT 709;"  NEXT I"                           !Increment loop
130  OUTPUT 709;"SUBEND"                             !End subroutine
140  OUTPUT 709;"CALL DEF_ARRAY"                     !Call subroutine
150  OUTPUT 709;"FREQ 2000"                          !Change freq to 2 kHz
160  OUTPUT 709;"APPLY WFV 2.3,CARRAY"               !Output cosine waveform
170  END
```

Example 12-2: Arbitrary Waveform Using Loop (COSWFM12)

This program also defines and outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A. However, in contrast to Example 12-1, the array points are first computed and stored in a controller array and then downloaded.

```
10   !file COSWFM12
20   !
30   DIM Carray(0:2047)                 !Dim controller array
40   FOR I=0 to 2047                    !Begin counting loop
50     Carray(I)=COS(2*PI*I/2048)       !Compute values
60   NEXT I                             !Increment loop
70   CLEAR 709                          !Clear HP 3245A
80   OUTPUT 709;"RST"                   !Reset HP 3245A
90   OUTPUT 709;"SCRATCH"               !Clear HP 3245A memory
100  OUTPUT 709;"DIM CARRAY(2047)"      !Dim REAL array
110  OUTPUT 709;"FILL CARRAY,";         !Fill array
120  OUTPUT 709;Carray(*)               !Fill with Carray points
130  OUTPUT 709;"FREQ 2000"             !Change freq to 2 kHz
140  OUTPUT 709;"APPLY WFV 2.3,CARRAY"  !Output cosine waveform
150  END
```

Programming Precomputed Waveforms

This section show how to program precomputed voltage or current waveforms when triggering is not used. The primary purpose of using precomputation is to reduce the time required to output the waveform after an **APPLY WFV** or **APPLY WFI** command is executed. As shown in Figure 12-5, there are four steps to program the HP 3245A for precomputed waveforms:

- Select waveform function.
- Define array values.
- Select channel parameters.
- Select waveform parameters.

```
Precomputed Waveforms

Select Waveform Function
. Scaled Voltage (APPLY WFV)
. Scaled Current (APPLY WFI)

Define Array Values
. Define Waveform       (Eq/Pts)
. Define Subroutine       (SUB)
. Define Array   (REAL/INTEGER)
. Fill Sine Array      (FILLAC)
. Fill Ramp Array      (FILLRP)
. Fill Arb Array       (FILLWF)

Select Channel Parameters
. Use Channel          (USE)
. Output Terminal      (TERM)
. Output Impedance     (IMP)
. Autorange Mode       (ARANGE)
. Output Range         (RANGE)
. Output Delay         (DELAY)

Select Waveform Parameters
. Output Frequency     (FREQ)
. DC Offset            (DCOFF)

Program Precomputed Waveform
```

Figure 12-5. Precomputed Waveforms - Selection Steps

Command Summary

Figure 12-6 summarizes commands for precomputed waveforms. As with user-defined waveforms, **APPLY WFV** sets the channel for voltage outputs and **APPLY WFI** sets the channel for current outputs.

Speed Enhancements with Precomputing

When an **APPLY** command is used to generate a sine, square, or arbitrary waveform, the HP 3245A must download 2048 points to the **USE** channel. To arithmetically scale (precompute) these points into a hardware integer format which the channel can use requires about 20 msec for sine waves, 45 msec for ramp waves, and 500 msec for arbitrary waveforms (real format). In addition, about 70 msec is required to download the points to the **USE** channel.

However, by using the **FILLAC**, **FILLRP**, or **FILLWF** commands, you can reduce the time required to output a sine, ramp, or arbitrary waveform to about 70 msec after an **APPLY WFV** or **APPLY DCI** command is executed. For example, the time required to output a ramp waveform is about 115 msec without precomputation, while the time required to output the waveform is about 70 msec with precomputation. (Precomputation is not available for square waveforms).

Array Values

As shown in Figure 12-6, the **FILLAC** and **FILLRP** commands directly precompute the sine or ramp waveform data points and store them in an INTEGER array which the channel can use. For arbitrary waveforms, the **FILLWF** command takes previously stored values in a REAL array, precomputes the values, and stores them in the INTEGER array. Then, when **APPLY WFV** is executed, the specified waveform (sine, ramp, or arbitrary) is output about 70 msec after the command is executed

---

NOTE

The array values can also be defined and stored using a subroutine or using the front panel **DIM Array** *and* **Edit Array** *keys.*

Selecting Waveform Function

The **APPLY WFV** *pp_amplitude* [,*array_name*] command generates an arbitrary voltage waveform The *pp_amplitude* parameter is the peak-to-peak amplitude of the output between 0.03125 Vac PP and 20 Vac PP in the 0 Ω output mode or between 0.015625 Vac PP and 10 Vac PP in the 50 Ω mode (when terminated with a 50 Ω load). Output voltage limits are 10% to 100% of the selected range.

APPLY WFI *pp_amplitude* [,*array_name*] generates an arbitrary current waveform. The *pp_amplitude* parameter is the peak-to-peak amplitude of the output between 0.00001 Amps AC PP and 0.2 Amps AC PP. Output current limits are 5% to 100% of the selected range.

For both **APPLY WFV** and **APPLY WFI**, the *array_name* parameter specifies the name of a REAL or INTEGER array containing 2048 points which defines the waveform. (REAL arrays were discussed in "Programming User-Defined Waveforms".)

Array Values

Define Array (REAL/INTEGER)

DIM or REAL defines a REAL array (must be 2048 elements) to store waveform values. INTEGER defines an INTEGER array (must be 2048 elements) to store values. REAL array values must be in range -1 to +1

Fill defined arrays with subroutine operation (mathematical functions) or with FILL array_name, list (non-mathematical) functions. Can also fill arrays A0 through A9 using front panel Edit Array key

Precompute Values (FILLAC/RP/WF)

FILLAC integer_array precomputes and stores a sine waveform into an INTEGER array specified by integer_array. FILLRP integer_array [,%_duty] precomputes and stores a ramp wave with duty cycle defined by %_duty into an INTEGER array specified by integer_array

FILLWF real_array, integer_array precomputes and stores the values from the REAL array defined by real_array into the INTEGER array defined by integer_array

Waveform Function

APPLY WFV or APPLY WFI pp_amplitude [,array_name] generates an arbitrary voltage or current waveform with peak-to-peak amplitude set by pp_amplitude, pp_amplitude, using values stored in array specified by [,array_name].

For precomputed waveforms, the waveform values are determined by multiplying each value in the INTEGER array specified by array_name by the peak value of the peak-to-peak amplitude specified with pp_amplitude

Channel Parameters

Use Channel (USE)

USE ch sets use (output) channel. USE 0 or USE CHANA sets channel A, USE 100 or USE CHANB sets channel B.

Output Terminal (TERM)

TERM [mode] sets the output terminal. TERM FRONT sets front panel output, TERM REAR sets rear panel output, TERM OFF/OPEN disconnects outputs

Output Impedance (IMP)

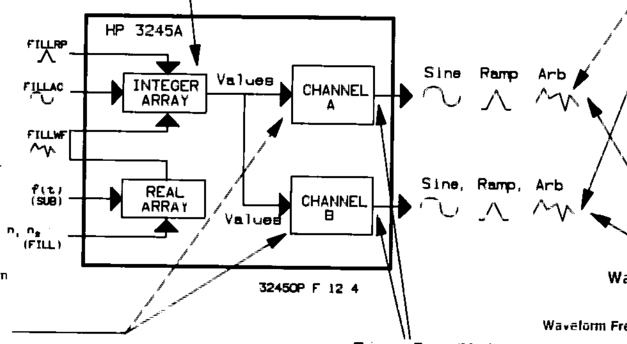
IMP mode sets the output impedance on the use channel. IMP 0 sets 0Ω impedance. IMP 50 sets 50Ω impedance

AC Range (RANGE or ARANGE)

RANGE [max_output] sets fixed output range. AC voltage output resolution depends on the output impedance (set by IMP). RANGE AUTO or ARANGE sets autorange

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Power-on delay time = 0.04 sec. Delay range is from 0 to 9.99 seconds with 0.01 seconds resolution



Trigger Event/Mode

Trigger Event (TRIGIN)

When TRIGMODE OFF is not set, TRIGIN [event] sets the input trigger event

Trigger Mode (TRIGMODE)

The TRIGMODE command sets the trigger mode. TRIGMODE ARMWF sets armed (synchronized) mode, TRIGMODE GATE sets gated mode, and TRIGMODE DUALFR sets dual-frequency mode.

Waveform Parameters

Waveform Frequency (FREQ)

FREQ freq1 [,freq2] sets waveform frequency (in Hz). Range is 0 to 1 MHz (ramp performance not specified above 100 kHz). Power-on frequency = 1000 Hz. (freq2 applies to dual freq mode only.)

DC Offset (DCOFF)

DCOFF volts sets DC offset on use channel. Peak amplitude of waveform plus magnitude of DC offset must be ±50% of selected AC range

Figure 12-6. Precomputed Waveforms - Command Summary

The output waveform amplitude for user-defined waveforms is determined by multiplying the peak value of the peak-to-peak amplitude set by **APPLY WFV** or **APPLY WFI** by the value of each REAL array element which was previously stored by **FILLWF** into the INTEGER array. For example, if the element values of a REAL array A are -0.05, 0.9, 0.2, 0,...,0.5 then **INTEGER IA(2047);FILLWF A,IA;APPLY WFV 4,IA** generates a voltage waveform with point values (PP) of -0.1 Vac, 1.8 Vac, 0.4 Vac, 0 Vac, .., 1.0 Vac.

The output amplitude for precomputed sine and ramp waveforms (previously stored by **FILLAC** or **FILLRP**) is the peak-to-peak amplitude set by **APPLY WFV** or **APPLY WFI**. Since the **APPLY** command *array_name* is optional, you can change the waveform peak-to-peak amplitude without reloading the 2048 points For example, to change the peak value of the output amplitude in the previous example to 1.5 Vac PP, use **APPLY WFV 3.0** and the values previously loaded from array IA will be used.

Defining Array Values

For precomputed waveforms, the **USE** channel uses the 2048 precomputed points stored in an INTEGER array. The INTEGER array is defined by an **INTEGER** command and is always filled by a **FILLAC**, **FILLRP**, or **FILLWF** command. For arbitrary waveforms, the waveform points to be precomputed by **FILLWF** must have been previously stored in a REAL array defined by **DIM** or **REAL**.

Defining the Array (DIM/REAL/ INTEGER)

Arrays are defined with an **DIM**, **REAL**, or **INTEGER** command **DIM** *array_name (max_index)* [,*array_name (max_index)*, ..] or **REAL** *name* [(*max_index*)] [,*name* [(*max_index*)], ...] define REAL arrays, while **INTEGER** *name* [(*max_index*)] [,*name* [(*max_index*)], ...] defines an INTEGER array.

Each REAL array must have 2048 elements and each array element (REAL or INTEGER) must have a value between -1 and +1, inclusive. For example, REAL A(2047) defines REAL array A of 2048 elements, while INTEGER IA(2047) defines an INTEGER array IA of 2048 elements.

Filling the Array (FILLAC/FILLRP/ FILLWF)

To fill the INTEGER array with sine waveform values, use **FILLAC** *integer_array*, where *integer_array* is the name of the array specified by an **INTEGER** command. To fill the INTEGER array with ramp waveform values, use **FILLRP** *integer_array* [,*%_duty*] where *integer_array* is the name of the array specified by an **INTEGER** command and *%_duty* is the waveform duty cycle between 5% and 95% (default = 50%).

For arbitrary waveforms, first define a REAL array with **DIM** or **REAL** and fill the array using a subroutine or the **FILL** command (refer to "Programming User-Defined Waveforms" for details). Then, use **FILLWF** *real_array,integer_array* to scale the REAL array points and store them in the INTEGER array. In the **FILLWF** command, *real_array* specifies the name of the REAL array and *integer_array* specifies the name of the INTEGER array.

Selecting Channel/ Waveform Parameters

As with defined AC waveforms, you can select five channel parameters: use channel (**USE**), output terminal (**TERM**), output impedance (**IMP**), AC range (**RANGE** or **ARANGE**), and output delay (**DELAY**). In addition, you can set the waveform frequency (**FREQ**) and the DC offset (**DCOFF**). Refer to Chapter 11 - Programming Defined Waveforms for details.

Programming Examples

When you have selected the waveform function, defined the array values, and selected channel/waveform parameters, program the HP 3245A for the desired output. Figure 12-7 shows a suggested programming sequence. Example programs follow to generate precomputed sine, ramp, and arbitrary waveforms.

Example 12-3: Precomputed Sine Waveform (SINPC12)

This program precomputes and then outputs a 10 kHz sine waveform with an amplitude of 5 Vac PP from channel A. Since precomputing is used, the waveform is output about 70 msec after **APPLY WFV** is executed. Without precomputing, the waveform would be output about 90 msec after **APPLY WFV** was executed.

```
10   !file SINPC12
20   !
30   CLEAR 709                              !Clear HP 3245A
40   OUTPUT 709;"RST"                       !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                   !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER SCALE(2047)"       !Dim INTEGER array
70   OUTPUT 709;"FILLAC SCALE"              !Fill INTEGER array
80   OUTPUT 709;"FREQ 10E+3"                !10 kHz frequency
90   OUTPUT 709;"APPLY WFV 5,SCALE"         !Generate sine waveform
100  END
```

Example 12-4: Precomputed Ramp Waveform (RMPPC12)

This program precomputes and then outputs a 3 kHz ramp waveform with an amplitude of 3.25 Vac PP from channel A. The duty cycle is set to 33%. Since precomputing is used, the waveform is output about 70 msec after **APPLY WFV** is executed. Without precomputing, the waveform would be output about 115 msec after **APPLY WFV** was executed.

```
10   !file RMPPC12
20   !
30   CLEAR 709                         !Clear HP 3245A
40   OUTPUT 709;"RST"                  !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"              !Clear HP 3245A memory
60   OUTPUT 709;"INTEGER J(2047)"      !Dim INTEGER array
70   OUTPUT 709;"FILLRP J,33"          !Fill array, set 33% duty
80   OUTPUT 709;"FREQ 3000"            !3 kHz frequency
90   OUTPUT 709;"APPLY WFV 3.25,J"     !Generate ramp waveform
100  END
```

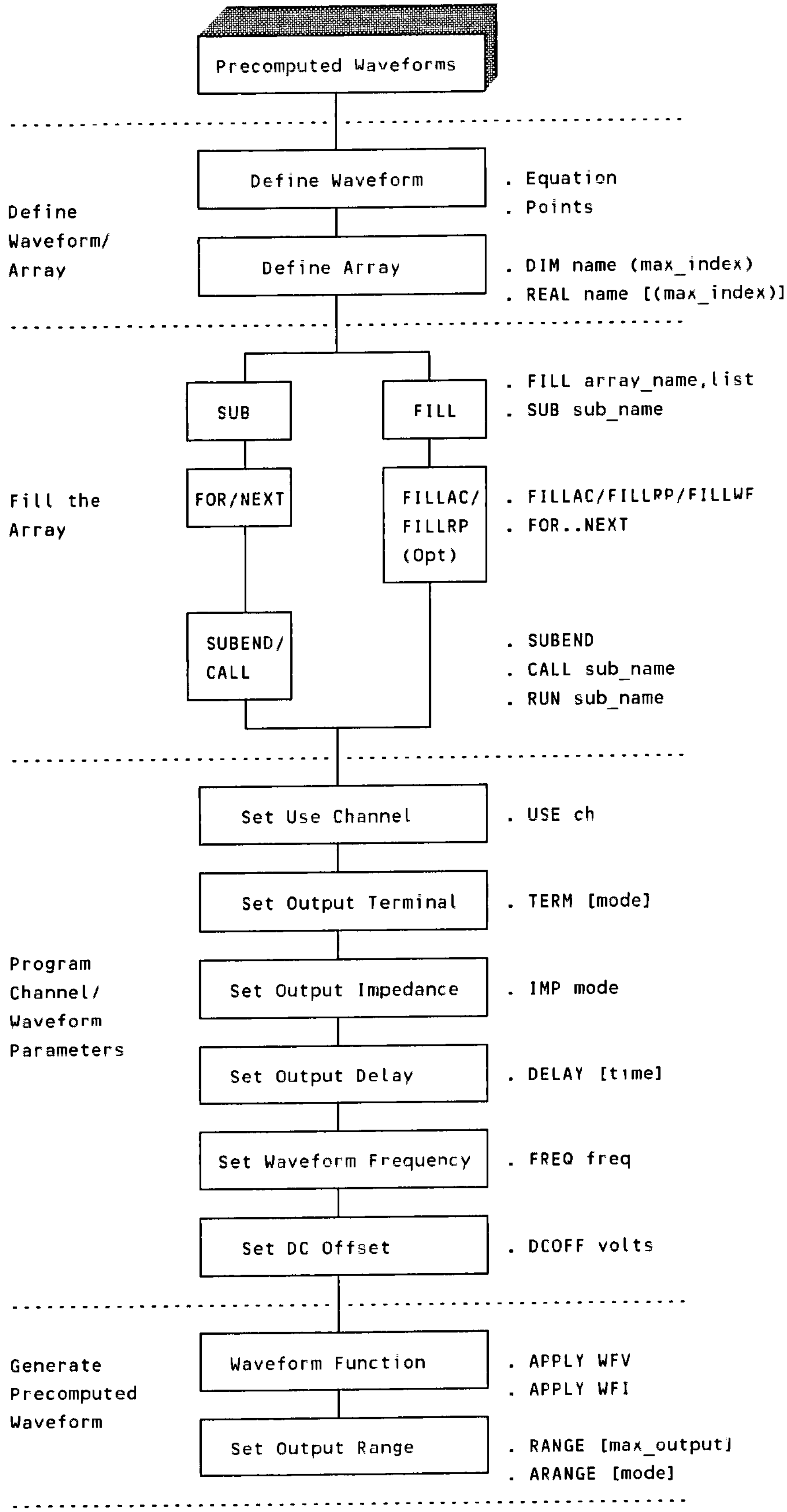


Figure 12-7. Precomputed Waveforms - Programming Sequence

Example 12-5: Precomputed Arbitrary Waveform (ARBPC12)

This program defines, precomputes, and then outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A. Since precomputing is used, the waveform is output about 70 msec after **APPLY WFV** is executed. Without precomputing, the waveform would be output about 570 msec after **APPLY WFV** was executed.

```
10   !file ARBPC12
20   !
30   CLEAR 709                                  !Clear HP 3245A
40   OUTPUT 709;"RST"                           !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                       !Clear HP 3245A
60   OUTPUT 709;"DIM CARRAY(2047)"              !Dim REAL array
70   OUTPUT 709;"REAL PI"                       !Define REAL var
80   OUTPUT 709;"PI = 3.1415925"                !Assign value to PI
90   OUTPUT 709;"SUB DEF_ARRAY"                 !Begin subroutine
100  OUTPUT 709;"  FOR I = 0 TO 2047"           !Begin counting loop
110  OUTPUT 709;"    CARRAY(I) = COS(2*PI*I/2048)"  !Compute values
120  OUTPUT 709;"  NEXT I"                      !Increment loop
130  OUTPUT 709;"SUBEND"                        !End subroutine
140  OUTPUT 709;"CALL DEF_ARRAY"                !Call subroutine
150  OUTPUT 709;"INTEGER J(2047)"               !Define INTEGER array
160  OUTPUT 709;"FILLWF CARRAY,J"               !Scale, store result in J
170  OUTPUT 709;"FREQ 2000"                     !2 kHz waveform frequency
180  OUTPUT 709;"APPLY WFV 2.3,J"               !Output precomputed waveform
190  END
```

Programming Triggered Waveforms

This section show how to program triggered (arbitrary) voltage or current AC waveforms when precomputing is not used. As shown in Figure 12-8, there are five steps to program the HP 3245A for triggered arbitrary waveforms:

- Select waveform function.
- Define array values.
- Select channel parameters.
- Select waveform parameters
- Select trigger mode/event.

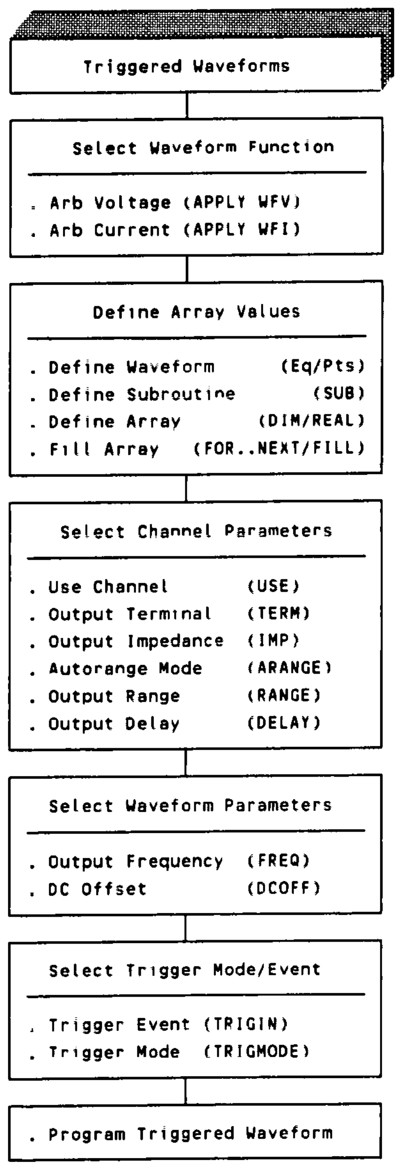


Figure 12-8. Triggered Waveforms - Selection Steps

Command Summary

Except for trigger event/mode, programming triggered arbitrary waveforms is identical to that for user-defined arbitrary waveforms. Refer to "Programming User-Defined Waveforms" for details on selecting waveform function, defining array values, and selecting channel and waveform parameters. Refer to Chapter 13 - Triggering Outputs/Waveforms for details on selecting the trigger mode/event. Figure 12-9 summarizes commands for triggered arbitrary waveforms.

Programming Example

When you have selected the waveform function, defined the array values, and selected the channel/waveform parameters and trigger mode/event, program the HP 3245A for the desired output. Figure 12-10 shows a suggested programming sequence An example program follows to generate triggered arbitrary waveforms

---

Example 12-6: Generating Triggered Waveform (TRGWF12)

This program defines and outputs a 2 kHz cosine waveform with a peak-to-peak amplitude of 2.3 Vac from channel A. The output is generated when an external trigger is input to the channel A Trigger (I/O) port

```
10   !file TRGWF12
20   !
30   CLEAR 709                                          !Clear HP 3245A
40   OUTPUT 709;"RST"                                   !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"                               !Clear HP 3245A memory
60   OUTPUT 709;"DIM CARRAY(2047)"                      !Dim REAL array
70   OUTPUT 709;"REAL PI"                               !Define REAL var
80   OUTPUT 709;"PI = 3.1415925"                        !Assign value to PI
90   OUTPUT 709;"TRIGMODE ARMWF"                        !Set armed mode
100  OUTPUT 709;"SUB DEF_ARRAY"                         !Begin subroutine
110  OUTPUT 709;"  FOR I = 0 TO 2047"                   !Begin counting loop
120  OUTPUT 709;"    CARRAY(I) = COS(2*PI*I/2048)"      !Compute values
130  OUTPUT 709;"  NEXT I"                              !Increment loop
140  OUTPUT 709;"SUBEND"                                !End subroutine
150  OUTPUT 709;"CALL DEF_ARRAY"                        !Call subroutine
160  OUTPUT 709;"FREQ 2000"                             !Change freq to 2 kHz
170  OUTPUT 709;"APPLY WFV 2.3,CARRAY"                  !Output cos waveform
180  OUTPUT 709;"TRIGIN EXT"                            !External trigger
190  END
```

Array Values

Define Array (REAL/INTEGER)

DIM or REAL defines a REAL array (must be 2048 elements) to store waveform values. INTEGER defines an INTEGER array (must be 2048 elements) to store values. REAL array values must be in range -1 to +1

Fill Array (SUB/FILL)

Use SUB (with FOR...NEXT loop) to fill a defined array for a mathematical function. Use FILL array_name,list to fill a defined array for a non-mathematical (user-defined) set of values. You can also fill arrays A0 through A9 using front panel Edit Array key

Waveform Function

APPLY WFV or APPLY WFI pp_amplitude [,array_name] generates an arbitrary voltage or current waveform with peak-to-peak amplitude set by pp_amplitude. pp_amplitude, using values stored in array specified by [,array_name].

Waveform values are determined by multiplying each value in the array with the peak value of the peak-to-peak amplitude

Channel Parameters

Use Channel (USE)

USE ch sets use (output) channel. USE 0 or USE CHANA sets channel A, USE 100 or USE CHANB sets channel B.

Output Terminal (TERM)

TERM [mode] sets the output terminal. TERM FRONT sets front panel output, TERM REAR sets rear panel output, TERM OFF/OPEN disconnects outputs

Output Impedance (IMP)

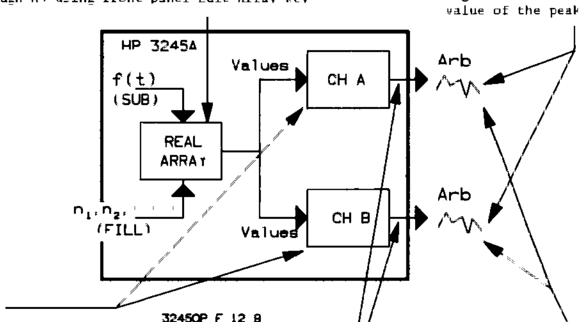
IMP mode sets the output impedance on the use channel. IMP 0 sets 0Ω impedance. IMP 50 sets 50Ω impedance

AC Range (RANGE or ARANGE)

RANGE [max_output] sets fixed output range. AC voltage output resolution depends on the output impedance (set by IMP). RANGE AUTO or ARANGE sets autorange

Output Delay (DELAY)

DELAY [time] sets delay time (in sec) between output commands. Power-on delay time = 0.04 sec. Delay range is from 0 to 9.99 seconds with 0.01 seconds resolution



Trigger Event/Mode

Trigger Event (TRIGIN)

When TRIGMODE OFF is not set, TRIGIN [event] sets the input trigger event

Trigger Mode (TRIGMODE)

The TRIGMODE command sets the trigger mode. TRIGMODE ARMWF sets armed (synchronized) mode, TRIGMODE GATE sets gated mode, and TRIGMODE DUALFR sets dual-frequency mode

Waveform Parameters

Waveform Frequency (FREQ)

FREQ freq1 [,freq2] sets waveform frequency (in Hz). Range is 0 to 1 MHz (ramp performance not specified above 100 kHz). Power-on frequency = 1000 Hz (freq2 applies to dual-freq mode only.)

DC Offset (DCOFF)

DCOFF volts sets DC offset on use channel. Peak amplitude of waveform plus magnitude of DC offset must be ±50% of selected AC range

Figure 12-9. Triggered Waveforms - Command Summary

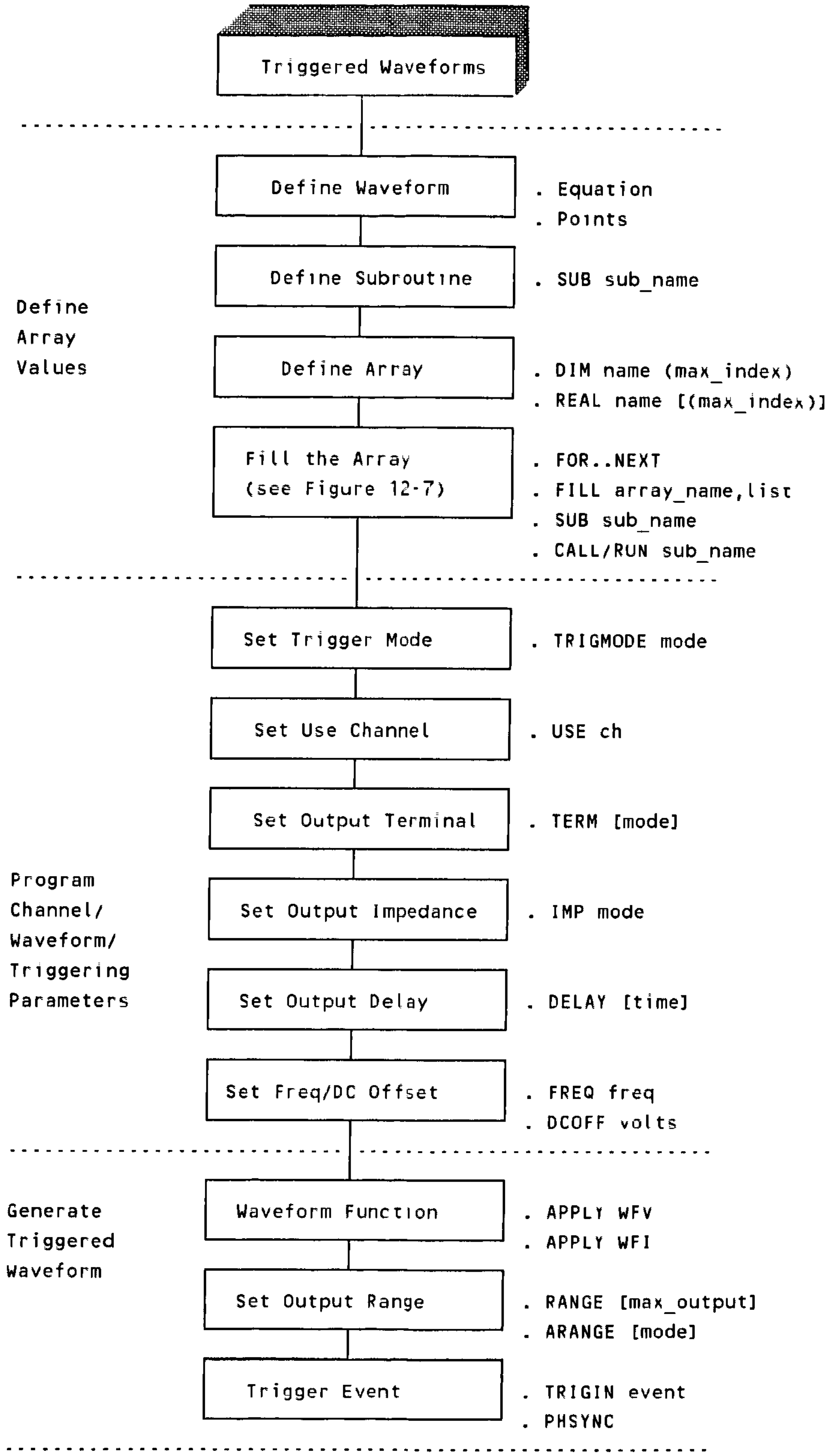


Figure 12-10. Triggered Waveforms - Programming Sequence

Contents

Chapter 13
Triggering Outputs/Waveforms

Chapter 13

Triggering Outputs/Waveforms

Introduction

This chapter describes techniques to trigger DC outputs and AC waveforms for the HP 3245A.

Chapter Contents

The chapter contents are:

- **Selecting Trigger Source** describes front panel, software, and trigger bus triggering methods.

- **Selecting Trigger Modes** summarizes methods to set synchronized, gated, or dual-frequency mode of operation.

- **Selecting Sync Out Destination** shows how to set the Sync Out signal destination.

- **Selecting Reference Frequency Path** shows how to select the reference frequency output destination and the input source.

Triggering Overview

Figure 13-1 shows HP 3245A triggering operation and commands. Table 13-1 summarizes HP 3245A triggering commands.

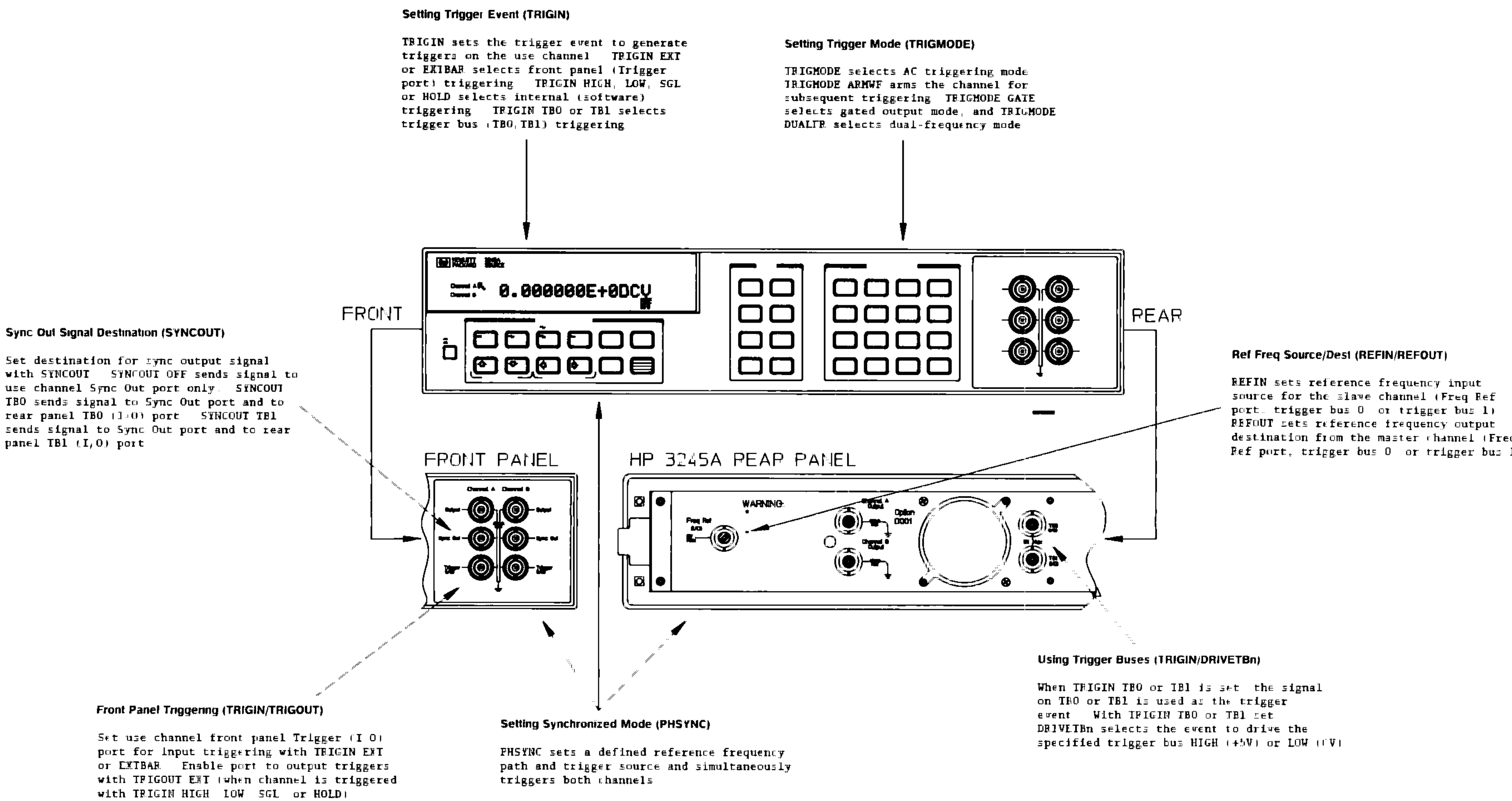


Figure 13-1. HP 3245A Triggering Overview

Table 13-1. HP 3245A Triggering Commands

Command	Description
DRIVETBn/ DRIVETBn?	Set/Read Trigger Bus Event. DRIVETBn sets event to drive specified trigger bus (0 or 1). DRIVETBn? reads trigger bus event.
PHSYNC	Set Synchronized Mode. PHSYNC sets defined ref freq path and trigger source and simultaneously triggers both channels.
REFIN/ REFIN?	Set/Read Ref Freq Input. REFIN sets ref freq input source for slave channel(s). REFIN? reads input source.
REFOUT/ REFOUT?	Set/Read Ref Freq Output. REFOUT sets ref freq output destination from the master channel. REFOUT? reads output destination.
SYNCOUT/ SYNCOUT?	Set/Read Sync Out. SYNCOUT sets destination (front panel and/or TB0 or TB1) of Sync output signal. SYNCOUT? reads destination.
TBn?	Trigger Bus Query. TBn? reads logic level (0 or 1) of the specified trigger bus.
TRG	Pulse Trigger Bus. Pulses trigger bus (TB0 or TB1) specified in DRIVETBn TRG command when a TRG command is received.
TRIGFREQ	Triggered Frequency Change. Allows fast, triggered change of frequency. (Refer to Chapter 14 - Advanced Programming Topics.)
TRIGIN/ TRIGIN?	Set/Read Trigger Source. TRIGIN sets trigger source (front panel, software, or trigger bus) to generate triggers on use channel. TRIGIN? reads trigger source on the use channel.
TRIGMODE/ TRIGMODE?	Set/Read Trigger Mode. TRIGMODE sets AC triggering mode (synchronized, gated, or dual-frequency) on the use channel. TRIGMODE? reads mode on the use channel.
TRIGOUT/ TRIGOUT?	Enable/Read Trigger Output. TRIGOUT enables/disables trigger output from use channel Trigger connector when channel is triggered with TRIGIN HIGH, HOLD, LOW, or SGL. TRIGOUT? reads the enable state.

Selecting Trigger Source

TRIGIN *source* selects the trigger source used to generate triggers on the **USE** channel. **TRIGIN?** returns the source for the **USE** channel. As shown in Figure 13-2, you can select front panel, software, or trigger bus triggering on the **USE** channel with the **TRIGIN** command.

Front panel triggering is selected with **TRIGIN EXT** or **EXTBAR**; software triggering with **TRIGIN HIGH, HOLD, LOW,** or **SGL**; and trigger bus triggering with **TRIGIN TB0** or **TB1**. You can read the current source for the **USE** channel with **TRIGIN?**

Table 13-2 shows the **TRIGIN** *source* parameters. Power-on *source* = HIGH. A description of front panel, software, and trigger bus triggering follows. Refer to "Selecting Reference Frequency Path" for an example program using **TRIGIN**.

Table 13-2. Trigger Input (TRIGIN) Parameters

event	Definition
TB0	Trigger Bus 0. Trigger using the signal on the TB0 trigger bus.
TB1	Trigger Bus 1. Trigger using the signal on the TB1 trigger bus.
EXT	Trigger Connector. Trigger using the input to the front panel Trigger connector.
EXTBAR	Inverse of EXT. Trigger using the inverse of the input to the front panel Trigger connector.
LOW	Low Signal - Software Control. Used with the HIGH parameter to internally trigger.
HIGH	High Signal - Software Control. Used with the LOW parameter to internally trigger.
HOLD	Same as HIGH parameter.
SGL	Single Trigger. Assure a HIGH then change to LOW and back to HIGH to internally trigger.

Front Panel Triggering (TRIGIN)

External triggers can be input to the Channel A and Channel B Trigger (I/O) ports on the front panel. When **TRIGIN EXT** or **EXTBAR** is set, the **USE** channel is triggered with an external (user-supplied) trigger into the Channel A or Channel B Trigger (I/O) port on the front panel (the trigger input must be TTL-compatible). With **TRIGIN EXT**, the **USE** channel is triggered by the Trigger (I/O) port. With **TRIGIN EXTBAR**, the **USE** channel is triggered by the inverse of the Trigger (I/O) port.

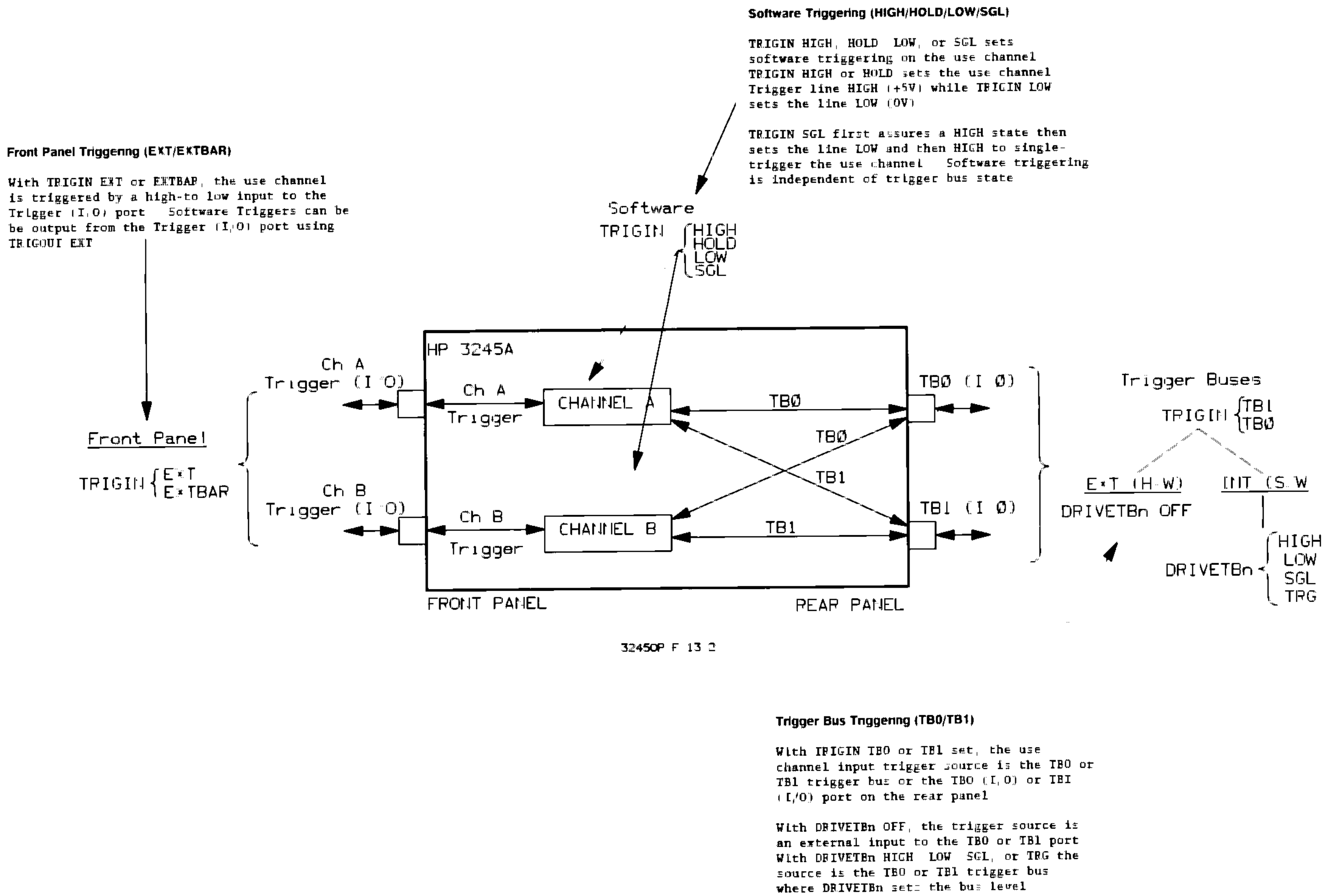


Figure 13-2. Input Trigger Sources

Software Triggering (TRIGIN)

You can internally (software) trigger the **USE** channel with **TRIGIN HIGH, HOLD, LOW,** or **SGL**. For example, **TRIGIN HIGH** or **TRIGIN HOLD** sets the **USE** channel input HIGH (+5V). Triggering in this manner is equivalent to setting **TRIGIN EXT** and inputting a +5V level to the **USE** channel Trigger (I/O) port. **TRIGIN LOW** sets the **USE** channel input LOW (0V). **TRIGIN SGL** first assures that the use channel is set HIGH, then sets the channel LOW and back to HIGH to single-trigger the channel.

Enabling Trigger Outputs (TRIGOUT)

When **TRIGIN HIGH, HOLD, LOW,** or **SGL** is set, the **USE** channel can be enabled to generate output triggers with the **TRIGOUT** *control* command. **TRIGOUT EXT** enables outputs from the **USE** channel Trigger port, while **TRIGOUT OFF** disables outputs from the **USE** channel Trigger port. The power-on setting is **TRIGOUT OFF** (outputs disabled).

The Trigger (I/O) connector is automatically disabled (**TRIGOUT OFF** is set) when **TRIGIN TB0, TB1, EXT,** or **EXTBAR** is selected. Similarly, if **TRIGIN TB0, TB1, EXT,** or **EXTBAR** is selected and **TRIGOUT EXT** is then selected, **TRIGIN HIGH** is automatically set. Refer to "Selecting Reference Frequency Path" for an example program using **TRIGOUT**.

Trigger Bus Triggering (TRIGIN/ DRIVETBn)

TRIGIN TB0 or **TRIGIN TB1** sets trigger bus triggering on the specified **USE** channel. **TRIGIN TB0** sets trigger bus TB0, while **TRIGIN TB1** sets trigger bus TB1 as the bus to be used. The **DRIVETBn** command selects the source to drive the specified trigger bus (TB0 or TB1). Use **DRIVETBn?** to read the source for the specified trigger bus. Figure 13-3 shows simplified trigger bus triggering operation.

With trigger bus triggering (**TRIGIN TB0** or **TRIGIN TB1** set), you can input an external trigger into the TB0 (I/O) or TB1 (I/O) port on the HP 3245A rear panel or you can use internal (software) triggering to trigger the specified **USE** channel. To do this, **DRIVETBn OFF** must be set. This disables the software drivers as well as any channel's drive (with **REFOUT** or **SYNCOUT**) to that trigger bus so that the TB0 or TB1 port can act as an input terminal. Then, you can input a (TTL-compatible) signal into the TB0 or TB1 port to trigger the specified **USE** channel (channel A or B).

Alternatively, you can use **DRIVETBn LOW, HIGH, SGL,** or **TRG** to internally drive the specified trigger bus. As shown in Figure 13-3, **DRIVETBn LOW** drives the specified trigger bus LOW (0V), while **DRIVETBn HIGH** drives the specified trigger bus HIGH (+5V). **DRIVETBn SGL** pulses the specified trigger bus (LOW then HIGH). With **DRIVETBn TRG** set, the specified trigger bus is pulsed (LOW then HIGH) when the HP 3245A receives a **TRG** command or an HP-IB Group Execute Trigger (**GET**) command. Refer to **SYNCOUT** or **REFOUT** commands for more information on driving a trigger bus from a channel.

Reading Trigger Bus Level (TBn?)

Use the **TBn?** command to read the logic level (HIGH or LOW) on the specified trigger bus (trigger buses are normally set HIGH). A "1" returned indicates a HIGH level, while a "0" returned indicates a LOW level on the specified trigger bus.

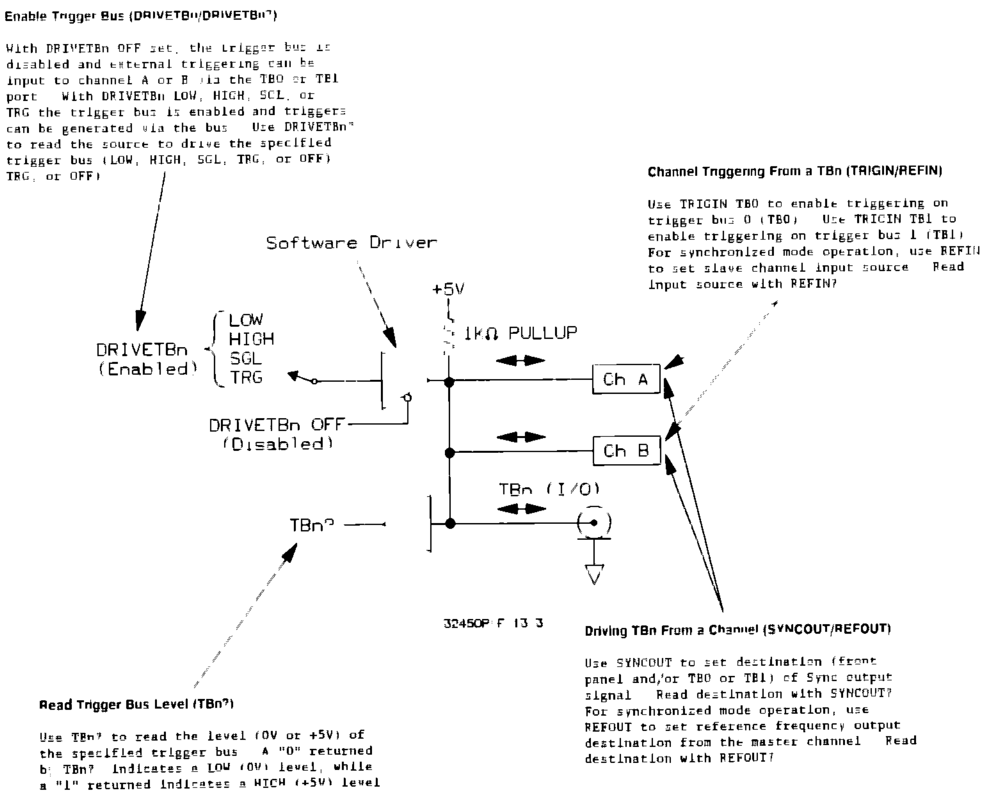


Figure 13-3. Trigger Bus Triggering

Selecting Trigger Modes

TRIGMODE *mode* selects one of three AC triggering modes for the **USE** channel. **TRIGMODE?** reads the current triggering mode on the **USE** channel. **TRIGMODE OFF** disables all triggering modes; **TRIGMODE ARMWF** selects synchronized (armed) mode; **TRIGMODE GATE** selects gated output mode; and **TRIGMODE DUALFR** selects dual-frequency mode. The power-on *mode* is **TRIGMODE OFF**.

The **TRIGMODE?** command returns the current *mode* setting (**OFF**, **ARMWF**, **GATE**, or **DUALFR**) for the use channel. Refer to Chapter 11 - Programming Defined Waveforms for more information and example programs using **TRIGMODE**

Synchronized Mode (TRIGMODE ARMWF)

TRIGMODE ARMWF allows AC waveforms from both channels to be synchronized. In this mode, the waveform begins outputting when a high-to-low edge is received. This allows the synchronization of two waveforms, since when a trigger from the source set by **TRIGIN** is simultaneously received on both channels, AC waveforms begin outputting synchronously from both channels.

Gated Mode (TRIGMODE GATE)

With gated waveform mode, when the trigger level from the source set by **TRIGIN** is LOW (0V), the **USE** channel generates the output waveform specified by **APPLY** When the trigger level goes HIGH (+5 V), the output waveform continues only until it reaches its zero phase point and then stops.

Dual-Frequency Mode (TRIGMODE DUALFR)

With dual-frequency mode, an AC waveform (sine, ramp, square, or arbitrary) can be modulated between two frequencies. An **APPLY** command sets the waveform to one of the two assigned frequencies (**FREQ** command) based on the logic level of the trigger event.

Selecting Sync Out Destination

SYNCOUT *destination* controls the destination of the Sync output signal for AC waveforms **SYNCOUT?** returns the Sync signal output destination (rear panel TB0 or TB1 connector)

The Sync Out Signal

All AC waveforms (sine, ramp, square, and arbitrary) have an associated sync signal which is output from the front panel Sync Out port when the waveform is generated. The sync signal is a logic "1" (+5V) when the waveform output level is positive relative to its DC offset. The sync signal is a logic "0" (0V) when the waveform output level is negative relative to its DC offset. Figure 13-4 shows an example sync output for a ramp waveform with -2.0 VDC offset.

Sync Out Signal Destinations (SYNCOUT)

The Sync Out signal is always routed to the front panel Sync Out connector for the specified **USE** channel, but can also be routed to the TB0 or TB1 ports on the rear panel. **SYNCOUT OFF** disables external routing of sync signal and the signal is routed only to the Sync Out connector **SYNCOUT TB0** or **TB1** automatically disables the software drivers (**DRIVETBn**) and/or the other channel's drive (**SYNCOUT** or **REFOUT**) to the selected trigger bus and routes the present **USE** channel's sync signal to the TB0 or TB1 port.

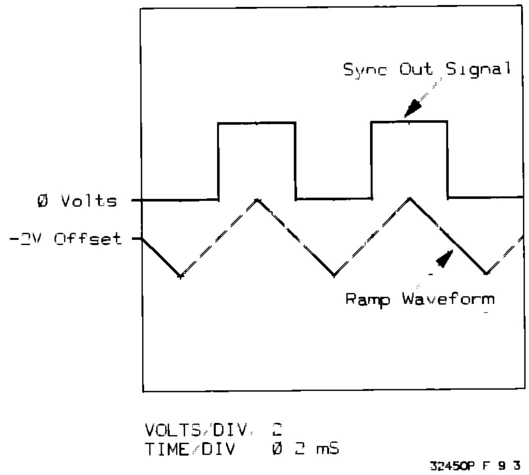


Figure 13-4. Sync Output Signal

Example 13-1. Gating Waveform Using Sync Out (SYNCO13)

This program uses the Sync Out signal to gate the channel B output. Channel A is set for a 1 kHz sine wave @ 1 Vac PP and channel B is set for a 2 kHz sine wave @ 2 Vac PP. **SYNCOUT TB0** and **TRIGIN TB0** allow the sync out signal to be sent from channel A to channel B via the TB0 trigger bus

Since channel B is set for gated waveform mode with **TRIGMODE GATE**, when the Sync Out signal goes from LOW to HIGH, the channel B waveform is output. When the Sync Out signal goes from HIGH to LOW, the channel B waveform is not output. Thus, channel B output occurs only when the channel A waveform is positive with respect to its DC offset (0 V in this case).

```
10   !file SYNCO13
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   !
70   !Configure channel A
80   !
90   OUTPUT 709;"USE 0"              !Use channel A
100  OUTPUT 709;"SYNCOUT TBO"        !Sync out to TBO
110  OUTPUT 709;"APPLY ACV 1"        !Output sq wave @ 1 Vac PP
120  !
130  !Configure channel B
140  !
150  OUTPUT 709;"USE 100"            !Use channel B
160  OUTPUT 709;"TRIGMODE GATE"      !Set gated waveform mode
170  OUTPUT 709;"TRIGIN TBO"         !TBO is trigger source
180  OUTPUT 709;"FREQ 2E3"           !Set frequency to 2 kHz
190  OUTPUT 709;"APPLY ACV 2"        !Output gated sq wave @ 2 Vac PP
200  END
```

Selecting Reference Frequency Path

When synchronized mode is set (**TRIGMODE ARMWF**), the reference frequency output from a "master" channel can be used to synchronize the output between the master channel and a "slave" channel. **REFOUT** and **REFIN** are used to define the path for the reference frequency from the master to the slave channel. **PHSYNC** command can also be used to select a predefined path.

Reference Frequency Paths (REFIN/ REFOUT)

Figure 13-5 shows three typical configurations for synchronized waveform operation with channel A as the master channel and channel B as the slave channel. In synchronized mode (set with **TRIGMODE ARMWF**), one channel is designated as a "master" and the other channel is designated as the "slave" channel. Each channel (master or slave) generates a reference frequency (nominally 1073741.824 Hz) based on its internal reference oscillator. In synchronized mode, the slave channel's oscillator locks on to the reference frequency generated by the master channel.

When the master and slave channels are triggered (both channels must be triggered at the same time), each channel outputs its specified waveform (sine, ramp, square, or arbitrary) at the selected frequency and phase. **REFIN** selects the source from which the slave channel's reference oscillator will accept the reference frequency from the master channel. **REFOUT** selects the master channel reference frequency output destination.

Selecting Output Destination (REFOUT)

REFOUT *destination* selects the reference frequency output destination for the master channel. **REFOUT?** returns the reference frequency output destination for the master channel **REFOUT?** returns the reference frequency output destination (**OFF, EXT, TB0,** or **TB1**) for the master channel. The power-on/reset setting on channel A is **REFOUT EXT** (i.e., channel B's reference is slaved to channel A's).

REFOUT OFF disables reference frequency output from the master channel and places the Freq Ref connector in high-impedance state. **REFOUT EXT** (power-on) outputs the reference frequency to the Freq Ref connector. **REFOUT TB0** or **TB1** automatically disables the software drivers (**DRIVETBn**) and/or the other channel's drive (**REFOUT** or **SYNCOUT**) to the selected trigger bus and outputs the reference frequency from the master channel to TB0 or TB1.

Since a channel cannot be a master and a slave at the same time, whenever **REFOUT** is executed with any parameter other than **OFF**, the channel input source is automatically set to internal (**REFIN INT**). Also, executing **REFOUT EXT** on a channel will change the other channel setting from **REFOUT EXT** to **REFIN EXT**

Selecting Input Source (REFIN)

REFIN selects the reference frequency input source for the slave channel. **REFIN?** returns the reference frequency input source for the slave channel. **REFIN INT** uses the channel's internal 1.07 MHz signal as the reference frequency input source, **REFIN EXT** uses the signal on the Freq Ref connector as the reference frequency input source, and **REFIN TB0** or **TB1** uses the signal on TB0 or TB1 as the reference frequency input source. Power-on/reset setting on channel B is **EXT** (i.e., channel B's reference is slaved to channel A's)

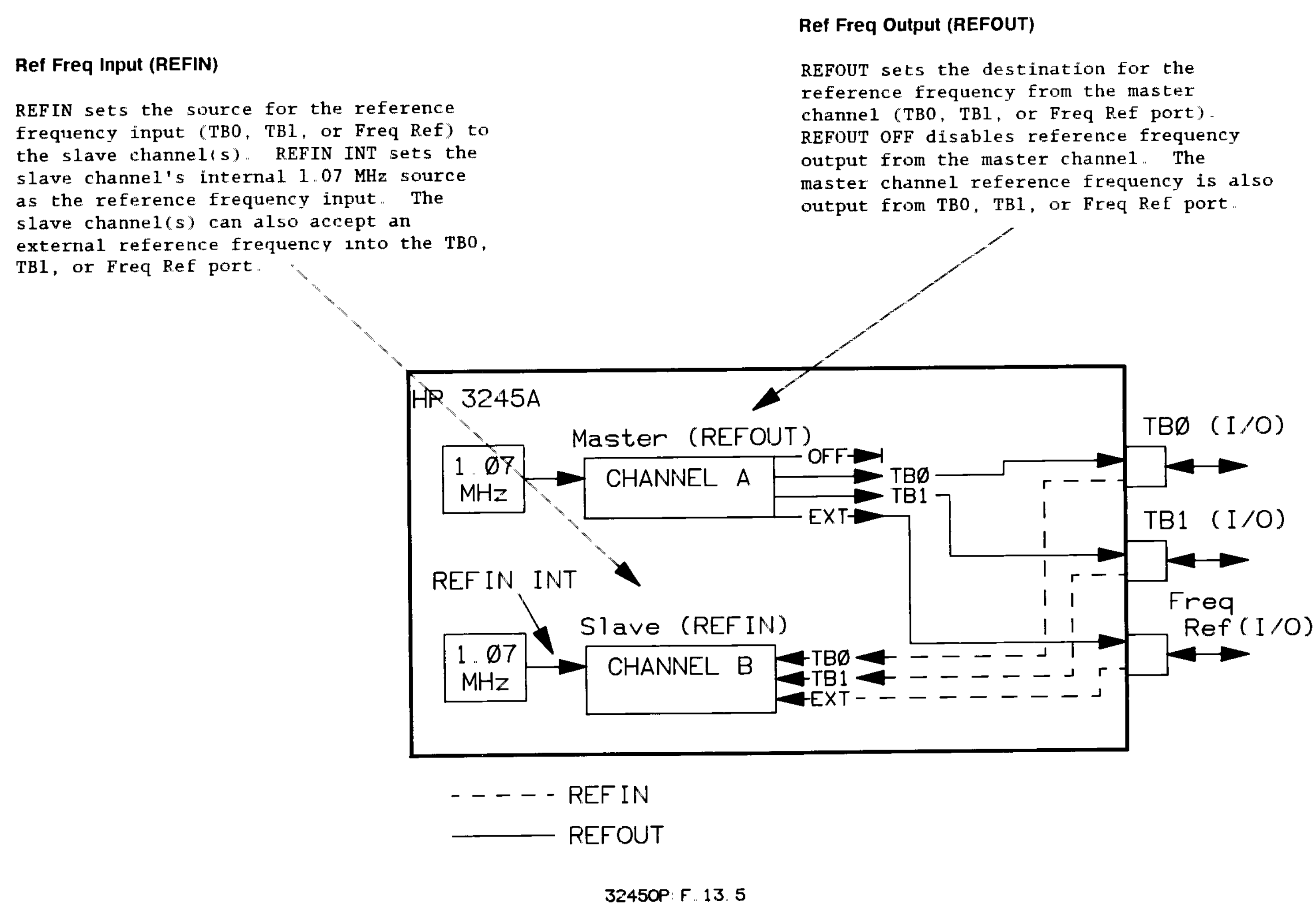


Figure 13-5. Three Reference Frequency Paths

Since a channel cannot be a master and a slave at the same time, whenever **REFIN** is executed with any parameter other than **INT**, the channel output source is automatically disabled (**REFOUT OFF**). Executing **REFOUT EXT** on a channel always changes the other channel setting from **REFOUT EXT** to **REFIN EXT** (and, hence, **REFOUT OFF**).

Example 13-2. Triggering Channels Simultaneously (SYNCH13)

This program demonstrates use of the **TRIGIN**, **REFIN**, **REFOUT**, **TRIGMODE**, and **TRIGOUT** commands to generate synchronized waveforms. To use this program, connect a BNC cable between the Channel A Trigger (I/O) port and the Channel B Trigger (I/O) port on the HP 3245A front panel. When this program executes, a 5 kHz sine waveform is output from channel A and a 5 kHz ramp waveform is output from channel B.

Channels A and B are configured for synchronized mode with **TRIGMODE ARMWF**. The reference frequency used for synchronization between the outputs is routed via the Freq Ref (I/O) port on the rear panel (note that no external connection is required) with **REFOUT EXT** and **REFIN EXT**.

The trigger source for both channels is software triggering as set with **TRIGIN LOW** (**TRIGIN HIGH** is set with **RST**). **TRIGOUT EXT** enables the channel A Trigger (I/O) port to output a trigger, while **TRIGIN EXT** sets the input trigger source for channel B to its Trigger (I/O) port. When **TRIGIN LOW** (line 300) is executed, a high-to-low trigger occurs on channel A which is passed via the BNC cable to channel B, thus simultaneously triggering both channels to output their respective waveforms.

```
10   !file SYNCH13
20   !
30   CLEAR 709                      !Clear HP 3245A
40   OUTPUT 709;"RST"               !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"           !Clear HP 3245A memory
60   !
70   !Configure channel A
80   !
90   OUTPUT 709;"USE 0"             !Use channel A
100  OUTPUT 709;"TRIGMODE ARMWF"    !Set ch A to sync mode
110  OUTPUT 709;"TRIGOUT EXT"       !Enable ch A Trigger output
130  OUTPUT 709;"REFOUT EXT"        !Ref freq to Freq Ref port
140  OUTPUT 709;"FREQ 5000"         !Output freq is 5 kHz
150  OUTPUT 709;"APPLY ACV 5"       !Output sine waveform @ 5 V
160  !
170  !Configure channel B
180  !
190  OUTPUT 709;"USE 100"           !Use channel B
200  OUTPUT 709;"TRIGMODE ARMWF"    !Set ch B to sync mode
210  OUTPUT 709;"TRIGIN EXT"        !Trigger port is trigger source
220  OUTPUT 709;"REFIN EXT"         !Ref freq from Freq Ref port
230  OUTPUT 709;"FREQ 5000"         !Output freq is 5 kHz
240  OUTPUT 709;"PANG 180"          !Phase angle = 180 degrees
```

```
250 OUTPUT 709;"APPLY RPV 5"        !Output ramp waveform @ 5 V
260 !
270 !Generate synchronized outputs
280 !
290 OUTPUT 709;"USE 0"              !Use channel A
300 OUTPUT 709;"TRIGIN LOW"         !Simultaneously trigger ch A,B
310 END
```

Synchronized Mode Using PHSYNC

Although you can use **REFIN** and **REFOUT** to set the reference frequency path and use **TRIGIN** to select a triggering method, the **PHSYNC** command offers a convenient way to set the reference frequency path, set a triggering method, and trigger the channels all with a single command.

The PHSYNC Command

Using the **PHSYNC** command is equivalent to setting channels A and B to a set of parameters and executing a trigger to output the waveforms synchronously Note that waveform parameters, such as amplitude, frequency, DC offset, etc. must still be set with the appropriate command. Table 13-3 lists the commands set by executing **PHSYNC**.

Table 13-3. Commands Set by PHSYNC

Command	Description
DRIVETB1 HIGH	Set TB1 HIGH (+5 V)
USE 0	Use channel A
TRIGMODE ARMWF	Set synchronized mode on ch A
REFOUT EXT	Ref freq dest is Freq Ref connector
TRIGIN TB1	Ch A input trigger source is TB1
USE 100	Use channel B
TRIGMODE ARMWF	Set synchronized mode on ch B
REFIN EXT	Ref freq input is Freq Ref connector
TRIGIN TB1	Ch B input trigger source is TB1
DRIVETB1 SGL	Trigger both channels simultaneously
Note: USE is returned to the channel specified when PHSYNC was executed.	

PHSYNC Operation

As shown in Figure 13-6, with **PHSYNC** channels A and B are set for synchronous operation (with **TRIGMODE ARMWF**), the reference frequency is routed via the Freq Ref connector, and triggering is via the TB1 trigger bus. If these parameters can be used, **PHSYNC** offers a convenient method to enter the reference frequency path and triggering method. (The channel waveform parameters must still be specified, however.)

After **PHSYNC** is executed, both channels remain in synchronized mode (**TRIGMODE ARMWF**). Subsequent execution of **APPLY**, **ARANGE ON**, **DCOFF**, **DUTY**, **FREQ**, **IMP**, **PANG**, **RANGE**, and **TRIGMODE ARMWF** commands will cause that channel's output to be suspended until a trigger is received on a channel (another **PHSYNC** will provide this trigger) or until a **TRIGMODE** other than **ARMWF** is selected. (Sending **ARANGE OFF**, **IMP**, or **RANGE** to the same range will not suspend a channel's output.)

Example 13-3. Generating Synchronized Waveforms (PHSYN13)

This program outputs a 5 kHz sine waveform (phase = 0°) from the channel A (the master channel) and a 5 kHz ramp waveform (phase = 180°) from channel B (the slave channel). **PHSYNC** sets synchronized waveform mode, sets TB1 as the channel input trigger sources, sets the Freq Ref connector as the output/input path for the reference frequency, and simultaneously triggers both channels (via TB1) to output their waveforms.

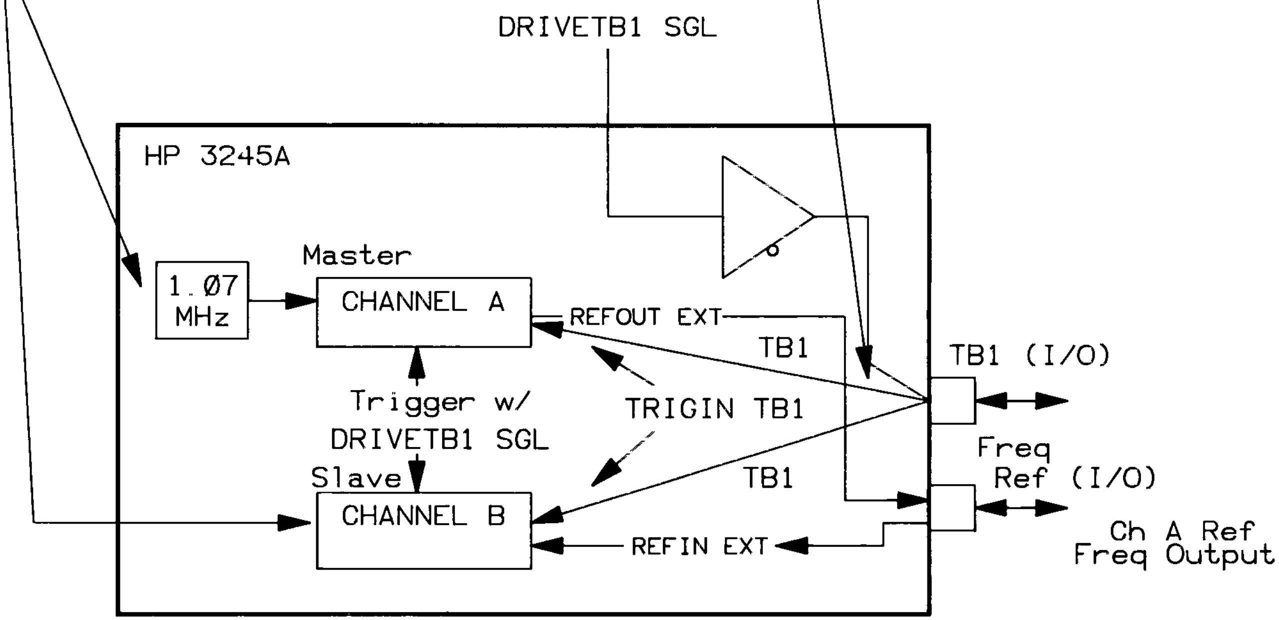
```
10   !file PHSYN13
20   !
30   CLEAR 709                       !Clear HP 3245A
40   OUTPUT 709;"RST"                !Reset HP 3245A
50   OUTPUT 709;"SCRATCH"            !Clear HP 3245A memory
60   !
70   !Configure channel A
80   !
90   OUTPUT 709;"USE 0"              !Use channel A
100  OUTPUT 709;"FREQ 5000"          !Output freq is 5 kHz
110  OUTPUT 709;"APPLY ACV 5"        !Output sine waveform @ 5 V
120  !
130  !Configure channel B
140  !
150  OUTPUT 709;"USE 100"            !Use channel B
160  OUTPUT 709;"FREQ 5000"          !Output freq is 5 kHz
170  OUTPUT 709;"PANG 180"           !Phase angle = 180 degrees
180  OUTPUT 709;"APPLY RPV 5"        !Output ramp waveform @ 5 V
190  !
200  !Generate synchronized outputs
210  !
220  OUTPUT 709;"PHSYNC"             !Generate synced outputs
230  END
```

PHSYNC Configuration

PHSYNC sets channel A as the master channel and channel B as the slave channel and sets the TB1 trigger bus as the input trigger source. The command also sets the Freq Ref port as the channel A output destination and channel B input source (no external connection is required).

PHSYNC Triggering

Since PHSYNC sets DRIVETB1 HIGH, neither channel begins generating its output until PHSYNC is executed. When PHSYNC is executed, both channels are simultaneously triggered (by DRIVETB1 SGL) and begin outputting their respective waveforms using the (master) channel A reference frequency for synchronization.



3245OP:F 13.6

Figure 13-6. PHSYNC Configuration

Contents

Chapter 14
Advanced Programming Topics

Chapter 14
Advanced Programming Topics

Chapter Contents

This chapter shows example programs for some advanced programming topics. The chapter contents are:

- **Enhancing Accuracy/Resolution** includes example programs which use the HP 3458A DMM to increase HP 3245A waveform accuracy and AC amplitude resolution.

- **Advanced Array Operations** includes example programs to recall channel states and to generate arbitrary waveforms of less than 2048 points.

- **Increasing Sweep Speeds** includes example programs to increase the sweep (change) speed of waveform amplitude, DC offset, and frequency.

NOTE

All programs in this chapter are written for HP 9000 Series 200/300 controller BASIC 4.0 or higher. Programs are stored on the Example Programs, Chapter 14 disc. Refer to Chapter 1 - Using This Manual for details on using the disc. Refer to the HP 3245A Command Reference Manual for command details.

Enhancing Accuracy/Resolution

This section gives guidelines to enhance HP 3245A output accuracy or amplitude resolution by using the HP 3458A Digital Multimeter (DMM)

Enhancing Output Accuracy

The HP 3458A Digital Multimeter (DMM) is a high-performance 8 1/2 digit multimeter which can measure voltage and current with very high accuracy. The accuracy of the HP 3245A can be adjusted to nearly the accuracy of the HP 3458A DMM by adjusting the output to produce a desired measurement at the HP 3458A input.

Since the adjustment is made such that the output at the HP 3458A input terminals is very precise, the HP 3458A input terminals also function as "remote-sense" terminals. Remote-sensing is useful for applications where the load will degrade the HP 3245A output accuracy or for controlling an external device such as a voltage-programmable power supply. Figure 14-1 shows the connections for "remote sensing" with the HP 3458A.

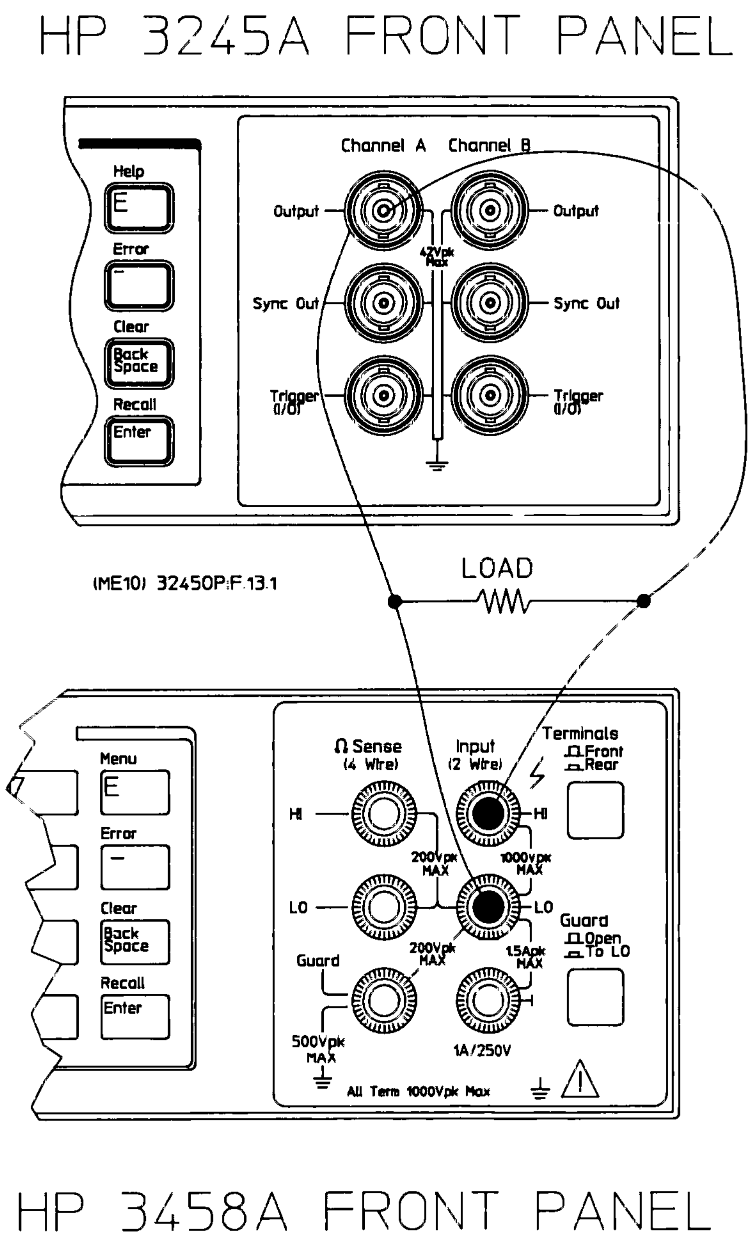


Figure 14-1. Increasing Output Accuracy - Connections

Example 14-1: Enhancing Output Accuracy (ENHAC14)

This program adjusts the DC and AC voltage outputs of the HP 3245A using the HP 3458A DMM. The program can also be used to adjust DC and AC current outputs.

Program Introduction

Prior to running this program, connect the HP 3245A and HP 3458A DMM for the desired function (DCV, ACV, DCI, or ACI) as shown in Figure 14-1. Then, program the instruments for desired speed and accuracy (**DELAY**, **DCRES**, **NPLC**, **AZERO**, **SETACV**, etc.) and frequency (**FREQ**) for AC waveforms.

For the input constants in the program, resetting both instruments before running the program the first time will guarantee execution without problems. Thereafter the program may be run, with or without changing inputs, without the requirement to reset.

This program does not reset the instruments so that front panel setups may be changed and maintained and the program can be run repeatedly to assure precise outputs are maintained with slowly changing loads.

The program can be run repeatedly to assure that precise outputs are maintained with slowly changing loads. Rerunning the program will NOT cause an initial change in the output if an amplifier is not in the feedback loop. If an amplifier is in the feedback loop, Dither (line 370) would be amplified and cause an output change Refer to Example 14-2 for guidelines to eliminate this change

HP 3245A Output Accuracy

The "no-load" accuracy of the output is simply the accuracy of the HP 3458A DMM, derated by **Accuracy**, the specified accuracy of the program. **Accuracy** can be no better than the resolution and short-term stabilty of the HP 3245A.

Typically these two combined produce an additive error of less than 1 ppm of HP 3245A range for DC or 0.05% of HP 3245A range for AC (refer to Example 14-2 for a method to reduce the AC error). Adding a load does not add additonal error unless the load varies faster than the program can be repeatedly executed to assure a precise output

Program Operation

This program has seven inputs: (1) **Chnl** = 0 for Channel A or 100 for Channel B; (2) **Func$** is the HP 3245A output function (DCV, ACV, DCI, or ACI); (3) **Target** is the desired HP 3245A output as measured by the HP 3458A DMM (peak-to-peak for AC); (4) **Accuracy** defines the range (Volts or Amps) within which the HP 3458A must measure to consider the output valid; (5) **Max_n_tries** is the number of attempts performed by the HP 3458A before generating an error, and (6), (7) **Max_src_output** and **Min_src_output** are the HP 3245A output voltage limits.

After setting up the seven inputs and assuring the proper HP 3245A and HP 3458A connection and functions (if the HP 3245A **Func$** changes it is sent **APPLY Func$ Target**), the program measures the present HP 3245A output (**Meas**).

A feedback loop gain of +1 is assumed and the **Source_output** is then changed (by **Dither** times +1) slightly in the direction of **Target**. If **Target** and **Meas** are close, **Dither** is limited to prevent changing the present output which may already be accurate.

Then, another measurement is made. The **New_src_output** is determined by adding to the **Old_src_output** The increment, **Excursion**, is the difference between **Target** and the present measured output, **Meas**, divided by the "relative" gain. This gain is determined by dividing the difference between the present and previous measured outputs, **(Meas-Old_meas)** by the difference between the present and previous programmed outputs, **(New_src_output-Old_src_output)**.

Noise can cause the present and previous measurments to be equal (or very nearly so) which in turn causes **Excursion** to become very large. In this situation **Excursion** is limited to the difference between **Target** and the present measured output, **Meas**. **New_src_output** are generated until the measured output, **Meas**, falls within **Accuracy** of **Target** or more than **Max_n_tries** attempts have been made

---

NOTE

AC is specified in peak-to-peak but measured in RMS (see the subroutine labeled **Measure:**)

```
10   !file ENHAC14
20   !
30   ASSIGN @Multimeter TO 722
40   ASSIGN @Source TO 709
50   DIM Mon_state$[256],Func$[3]
60   INTEGER Chnl,Max_n_tries,N_try
70   !
80   Chnl=0              ! 0 for CHANA, 100 for CHANB
90   Func$="DCV"         ! DCV, ACV, DCI, or ACI
100  Target=10           ! desired volts or amps (pk-pk for AC) measurement
110  Accuracy=.00001     ! in volts or amps
120  Max_n_tries=20
130  Max_src_output=10.25  ! in volts or amps (must be > Min_source_output)
140  Min_src_output=-10.25 ! in volts or amps
150  OUTPUT @Source;"USE ";Chnl
160  !
170  OUTPUT @Source;"MON STATE ";Chnl;";WAIT .5;DISP?"
180  ENTER @Source;Mon_state$                  ! check for changed Func$
190  IF POS(Mon_state$,Func$)=0 THEN OUTPUT @Source;"APPLY ";Func$;Target
200  OUTPUT @Source;"OUTPUT?"
210  ENTER @Source;Source_output               ! present output of source
220  OUTPUT @Multimeter;"FUNC ";Func$
230  !
```

```
240   !  set Multimeter so that ENTER will initiate and ENTER a measurement
250   OUTPUT @Multimeter;"TARM AUTO;TRIG SYN;NRDGS 1,AUTO"
260   !
270   OUTPUT @Source;"RANGE?"
280   ENTER @Source;Range
290   Dither=.5/100*Range                        ! max is .5% of range
300   !
310   N_try=0
320   GOSUB Measure                              ! first measurement
330   N_try=N_try+1             ! change output slightly and measure again
340   IF N_try>=Max_n_tries+1 THEN GOTO Error_numb
350   !
360   !        if Target and Meas are close, limit Dither to prevent glitch
370   Dither=SGN(Target-Meas)*MIN(Dither,ABS(Target-Meas)) ! assume +1 gain
380   New_src_output=Source_output+Dither
390   Old_src_output=Source_output
400   Old_meas=Meas
410   GOSUB Wr_new_output
420   GOSUB Measure
430   !
440   WHILE ABS(Target-Meas)>Accuracy                ! adjust loop
450     N_try=N_try+1
460     IF N_try>=Max_n_tries+1 THEN GOTO Error_numb
470     IF ABS(Meas-Old_meas)>1.E-12 THEN            ! prevent a /0
480
Excursion=(Target-Meas)*(New_src_output-Old_src_output)/(Meas-Old_meas
490   ELSE
500       Excursion=1.E+38                           ! simulate /0
510     END IF
520     Old_src_output=New_src_output
530     IF ABS(Excursion)>ABS(Target-Meas) THEN      ! limit the excursion
540       New_src_output=Old_src_output+ABS(Target-Meas)*SGN(Excursion)
550     ELSE                                         !   to the difference
560       New_src_output=Old_src_output+Excursion    !   between Target
570     END IF                                       !   and Meas in the
580     Old_meas=Meas                                !   direction of Excu
590     GOSUB Wr_new_output
600     GOSUB Measure
610     !
620   END WHILE
630 Done:OUTPUT @Multimeter;"TRIG AUTO"
640   LOCAL @Multimeter
650   STOP
660 Error_numb:BEEP
670   DISP "Can't adjust to ";Accuracy;" in ";Max_n_tries;"tries"
680   STOP
690 Err_minmax:BEEP
700   DISP "Min_src_output/Max_src_output limits violated:";New_src_output
710   STOP
720 Err_overload:BEEP
730   DISP "ERROR:  Overloading voltmeter"
740   STOP
750   !
760 Measure:  ENTER @Multimeter;Meas
```

```
770   IF ABS(Meas)>1200 THEN GOTO Err_overload
780   Meas=Meas*(1+(Func$[1,2]="AC")*(2^1.5-1))  ! convert rms to
790   RETURN                                      ! pk-pk in ACV/I
800 Wr_new_output:  IF New_src_output>Max_src_output THEN GOTO Err_minmax
810   IF New_src_output<Min_src_output THEN GOTO Err_minmax
820   OUTPUT @Source;"APPLY ";Func$;New_src_output
830   RETURN
840   END
```

Enhancing AC Amplitude Resolution

In Example 14-1 (refer to "Enhancing Output Accuracy"), AC voltage accuracy was limited to 0.05% by the HP 3245A amplitude resolution. This resolution can be extended by using channel B as a current-output sinewave which, when connected in parallel with Channel A, fine-tunes or adjusts the amplitude of a fixed 50 Ω voltage-output sinewave from channel A

With this procedure, the channel B output is actually added to the channel A output since the channel B current produces an additive voltage across the 50 Ω output resistance. Using this technique, the accuracy (determined by the combined resolution and short-term stabilty) can be typically extended to better than 5 ppm. Figure 14-2 shows the connections for this technique. An example program follows.

Example 14-2: Enhancing AC Amplitude Res (ENHAM14)

This program uses the same adjustment procedure as Example 14-1 to fine-tune the channel B output. The program inputs are changed and the "check for changed Func$" is replaced with code which assures both Func$ and synchronization of the two channels. Since the adjustment is still made at the HP 3458A input terminals, the terminals still provided the "remote-sense" function. This program is designed to drive >400 Ω loads.

Program Introduction

Before running this program, connect the HP 3245A and HP 3458A as shown in Figure 14-2. Then, program the instruments for appropriate speed and accuracy (**DELAY**, **DCRES**, **NPLC**, **AZERO**, **SETACV**, etc.). Set **DCOFF** and **PHASE** to zero for both channels.

The program can be run repeatedly to assure precise outputs are maintained with slowly changing loads. Rerunning the program will NOT cause an initial change or discontinuity in the output waveform if both channel functions have remained intact and synchronized, the desired final output (**Target**) is the same, and an amplifier is not in the loop.

For the input constants in the program, resetting both instruments before running the program the first time will guarantee execution without problems. Thereafter, the program may be run (with or without changing inputs) without the need to reset. This program does not reset the instruments so that the front panel setups may be changed and maintained and the program can be run repeatedly to assure precise outputs are maintained with slowly changing loads.

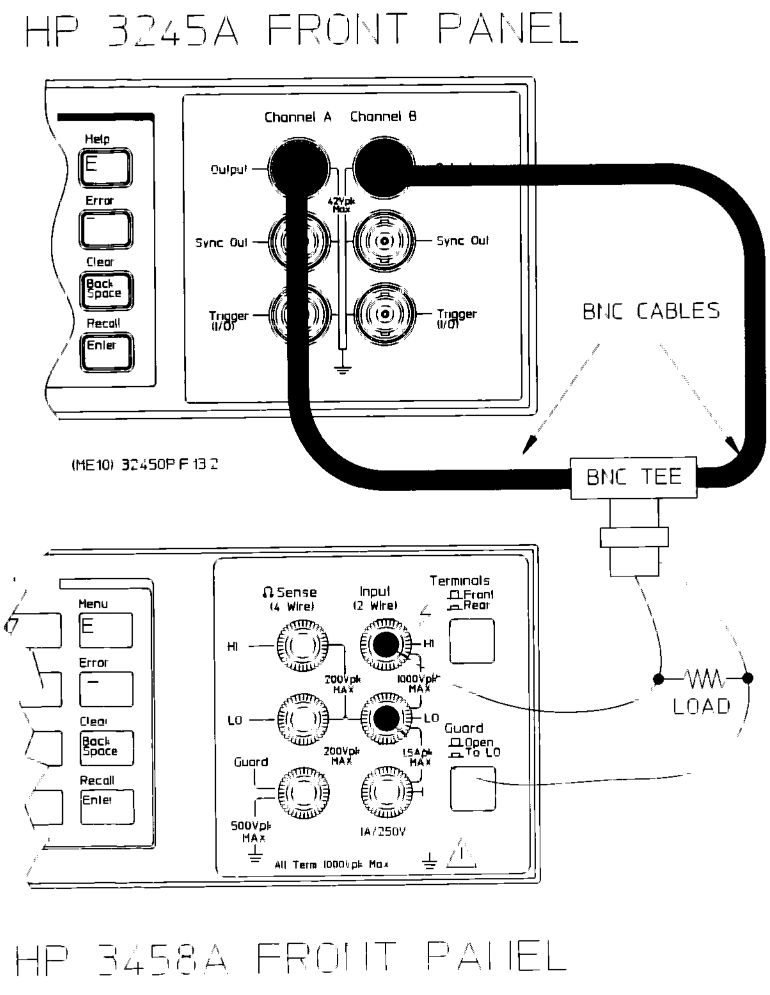


Figure 14-2. Increasing AC Amplitude Accuracy - Connections

Program Operation

The program has five inputs: (1) **Chnl** = 0 for channel A or 100 for channel B (**Tw_chnl** is the "tweak" (fine-tune adjustment) channel), (2) **Target** is the desired HP 3245A output as measured by the HP 3458A (peak-to-peak for AC); (3) **Freq** is the HP 3245A output frequency, (4) **Accuracy** defines the range in peak-to-peak volts within which the HP 3458A must measure to consider the HP 3245A output valid, and (5) **Max_n_tries** is the number of attempts performed before generating an error. **Max_src_output** and **Min_src_output** are fixed at the HP 3245A AC current output limits

After setting up the five inputs and assuring proper HP 3245A and HP 3458A connections, the program sets the fixed source to allow for a nominal 1% additive tweak. That is, the program sets the HP 3245A amplitude to 99% of target (divided by 2 since no load is assumed but **IMP 50** is set) and the frequencies of both channels are set to **Freq**.

If the channel B function changes, the function is not synchronized to **REFIN EXT**, or the channel A output has been changed (a new **Target** was entered), the code following the **IF** statement sets the initial tweak value on channel B to 1% of target (nominally 99% on A plus 1% on B equal **Target**) and synchronizes the two channels (this will cause a momentary discontinuity in the output waveform)

Following the **END IF** statement, both channels are set to **TRIGMODE OFF** to ensure amplitude changes do not cause the waveforms to halt and rearm as would happen if **TRIGMODE ARMWF** was not disabled.

The remainder of the program provides the same adjustment as in Example 14-1 but adjusts channel B current output by measuring the HP 3458A input voltage Since a 1 Amp change in channel B produces a (1 Amp times 50 Ω) change in the measurement, a "gain of 50" has been introduced in the feedback loop.

After the first measurment, only one measurement is known for only one HP 3245A output, which is not enough data to compute a "relative" adjustment gain. If a gain of "+1" is assumed, the **Dither** current output will change 50 times more than necessary and a discontinuity will appear.

Thus, **Dither** has been divided by 50 to account for the "gain of 50" and to eliminate an initial discontinuity. The rest of the algorithm will use two measurements for two outputs, will be able to compute "relative" gains, and therefore needs no modification.

```
10    !file ENHAM14
20    !
30    ASSIGN @Multimeter TO 722
40    ASSIGN @Source TO 709
50    DIM Mon_state$[256],Func$[3]
60    INTEGER Chnl,Max_n_tries,N_try,Tw_chnl
70    !                      (Chnl+100 will be used to "tweak")
80    Chnl=0                 ! fixed chan: 0 for CHANA, 100 for CHANB
90    Tw_chnl=Chnl+100       ! tweak channel
100   Func$="ACI"            ! tweak is ACI
110   Target=2.8284271       ! desired pk-pk volts >.1 (2.8284271 is 1vrms)
```

```
120   Freq=1000              ! frequency in Hertz
130   Accuracy=.000028       ! in volts (.000028 is 10ppm of Target)
140   Max_n_tries=20
150   Max_src_output=.2      ! in amps (must be > Min_source_output)
160   Min_src_output=.00001 ! in amps (pk-pk)
170   OUTPUT @Source;"USE ";Chnl;";OUTPUT?"
180   ENTER @Source;Source_output      ! enter Channel A's present output value
190   !
200   !  allow for a nominal 1% tweak on fixed source (/2 for IMP 50 & no load)
210   OUTPUT @Source;"PHASE 0;FREQ ";Freq;";IMP 50;APPLY ACV ";.99*Target/2
220   OUTPUT @Source;"USE ";Tw_chnl;";PHASE 0;FREQ ";Freq! same Freq on Tw_chnl
230   OUTPUT @Source;"MON STATE ";Tw_chnl;";WAIT .5;DISP?"
240   ENTER @Source;Mon_state$
250   N_try=(POS(Mon_state$,Func$)=0) OR (POS(Mon_state$,"REFIN EXT")=0)
260   N_try=N_try OR (ABS(Source_output-.99*Target/2)>1.E-12)
270   IF N_try>0 THEN        ! if Tw_chnl new Func$ or not sync'd, or new Target
280     OUTPUT @Source;"DRIVETB1 HIGH" ! assure TB1 is inactive & set up sync
290   !
300     OUTPUT @Source;"USE ";Chnl;";TRIGIN TB1;TRIGMODE ARMWF;REFOUT EXT"
310   !
320     OUTPUT @Source;"USE ";Tw_chnl;";TRIGIN TB1;TRIGMODE ARMWF;REFIN EXT"
330     OUTPUT @Source;"RANGE AUTO"
340     New_src_output=.01*Target/50           ! set to nominal 1% (I=V/50ohm)
350     GOSUB Wr_new_output                    ! check limits & apply
360     OUTPUT @Source;"RANGE ";.02*Target/50   ! fix range & allow a 2% tweak
370   !
380     OUTPUT @Source;"DRIVETB1 LOW;DRIVETB1 HIGH"! this starts both channels
390   END IF
400   OUTPUT @Source;"USE ";Chnl;";TRIGMODE OFF"! once sync'd turn ARMWF off
410   OUTPUT @Source;"USE ";Tw_chnl;";TRIGMODE OFF"
420   !
430   OUTPUT @Source;"OUTPUT?"
440   ENTER @Source;Source_output
450   OUTPUT @Multimeter;"FUNC ACV"              ! assure the ACV function
460   !
470   !  set Multimeter so that ENTER will initiate and ENTER a measurement
480   OUTPUT @Multimeter;"TARM AUTO;TRIG SYN;NRDGS 1,AUTO"
490   !
500   OUTPUT @Source;"RANGE?"
510   ENTER @Source;Range
520   Dither=.5/100*Range                        ! max is .5% of range
530   !
540   N_try=0
550   GOSUB Measure                              ! first measurement
560   N_try=N_try+1              ! change output slightly and measure again
570   IF N_try>=Max_n_tries+1 THEN GOTO Error_numb
580   !
590   !         if Target and Meas are close, limit Dither to prevent glitch
600   Dither=SGN(Target-Meas)*MIN(Dither,ABS(Target-Meas)) ! assume +1 gain
610   New_src_output=Source_output+Dither/50      ! correct for "gain" of 50
620   Old_src_output=Source_output               !(Meas=Source_output*50ohm)
630   Old_meas=Meas
640   GOSUB Wr_new_output
650   GOSUB Measure
```

```
660   !
670   WHILE ABS(Target-Meas)>Accuracy                 ! adjust loop
680     N_try=N_try+1
690     IF N_try>=Max_n_tries+1 THEN GOTO Error_numb
700     IF ABS(Meas-Old_meas)>1.E-12 THEN             ! prevent a /0
710       Excursion=(Target-Meas)*(New_src_output-Old_src_output)/(Meas-Old_meas
720     ELSE
730       Excursion=1.E+38                            ! simulate /0
740     END IF
750     Old_src_output=New_src_output
760     IF ABS(Excursion)>ABS(Target-Meas) THEN       ! limit the excursion
770       New_src_output=Old_src_output+ABS(Target-Meas)*SGN(Excursion)
780     ELSE                                          !   to the difference
790       New_src_output=Old_src_output+Excursion     !   between Target
800     END IF                                        !   and Meas in the
810     Old_meas=Meas                                 !   direction of Excu
820     GOSUB Wr_new_output
830     GOSUB Measure
840     !
850   END WHILE
860 Done:OUTPUT @Multimeter;"TRIG AUTO"
870   LOCAL @Multimeter
880   STOP
890 Error_numb:BEEP
900   DISP "Can't adjust to ";Accuracy;" in ";Max_n_tries;"tries"
910   STOP
920 Err_minmax:BEEP
930   DISP "Min_src_output/Max_src_output limits violated:";New_src_output
940   STOP
950 Err_overload:BEEP
960   DISP "ERROR:  Overloading voltmeter"
970   STOP
980   !
990 Measure:ENTER @Multimeter;Meas
1000  IF ABS(Meas)>1200 THEN GOTO Err_overload
1010  Meas=Meas*(1+(Func$[1,2]="AC")*(2^1.5-1)) ! convert rms to pk-pk in ACV/I
1020  RETURN
1030 Wr_new_output:IF New_src_output>Max_src_output THEN GOTO Err_minmax
1040  IF New_src_output<Min_src_output THEN GOTO Err_minmax
1050  OUTPUT @Source;"APPLY ";Func$;New_src_output
1060  RETURN
1070  END
```

Advanced Array Operations

This section includes example programs to store/recall channel states and to generate arbitrary waveforms with <2048 points.

Store/ Recall Channel State

For some applications, it may be convenient to extract command strings which define the present setup of an HP 3245A channel directly from the instrument.

Using **MON STATE** *ch* allows the current state of the channel specified by *ch* to be displayed. Executing **DISP?** via HP-IB queries the instrument for the present contents of the display. Since the display consists of a sequence of commands, the display information can be stored in a string and used later to reconfigure the channel to the stored state.

Example 14-3. Recalling Channel State (RECST14)

This program sets the specified channel (A or B) to monitor mode, reads the display into the string variable **Mon_state$** and replaces all the commas with semicolons (semicolons are the HP 3245A command delimiter). Then, it resets HP 3245A, programs a function and amplitude, and "recalls" the remainder of the channel state existing before the rest

```
10    !file RECST14
20    !
30    DIM Mon_state$[256]
40    INTEGER Chnl
50    Chnl=0                       ! 0 = ch A, 100 = ch B
60    OUTPUT 709;"MON STATE ";Chnl
70    WAIT 1                       ! wait at least .5 second for display to fill
80    OUTPUT 709;"DISP?"
90    ENTER 709;Mon_state$
100 !
110 !                                delete header, function, and amplitude
120   Mon_state$=Mon_state$[POS(Mon_state$,"FREQ"),LEN(Mon_state$)]
130 !
140   WHILE POS(Mon_state$,"""")>0! delete trailing quote
150     Mon_state$[POS(Mon_state$,"""");1]=""
160   END WHILE
170   WHILE POS(Mon_state$,",")>0 ! change all commas to semicolons
180     Mon_state$[POS(Mon_state$,",");1]=";"
190   END WHILE
200 !
210   OUTPUT 709;"RESET"          ! set the box to its reset state
220   OUTPUT 709;"USE ";Chnl
230   OUTPUT 709;"APPLY SQV 1.25" ! send function and amplitude
240   OUTPUT 709;Mon_state$       ! send remaining previous settup
250   END
```

Applying Waveforms (<2048 Points)

For some applications, it is necessary to apply a waveform which has fewer than 2048 points specified. However, the HP 3245A arbitrary waveform functions, **APPLY WFV** or **APPLY WFI**, always generate an output from a 2048-point array. When fewer than 2048 points are specified, the remaining elements contain zeros The example programs in this section show ways to generate arbitrary waveforms using <2048 points.

Applying Waveform - Using External Time Base

With the triggered DC functions (**APPLY DCMEMV** or **APPLY DCMEMI**), you can program outputs using 2 up to 2048 points and still use the same analog hardware used for **APPLY WFV** or **APPLY WFI**. For the triggered functions, the **TRIGIN** source be connected to an external timebase or an HP 3245A subroutine can be used to generate the timebase.

When channel B is used as this timebase, the channel can be programmed in sampling frequency or as a sampling time increment. The array elements are now specified in Volts or Amps directly and each element is guaranteed to be held for one sampling clock cycle. An example program follows.

Example 14-4: Waveform <2048 Pts - External Timebase (APPTB14)

This program generates a 10-point sinewave on channel A using channel B as the external timebase. The timebase can be programmed as a frequency (in Hz) or as a time increment (in seconds).

The program has four inputs: (1) **Func$** is set to Volts for voltage outputs or is set to I for current outputs; (2) **Timebase$** is a string containing the timebase frequency (Hz) or time increment (seconds); (3) **N_points** is the number of points of waveform data; and (4) the waveform data.

One cycle of the waveform will be generated in **N_points** time increments. For this program, the waveform data is loaded with a sine wave by the HP 3245A subroutine **DEF_ARRAY**, but the data could have been generated by a digitizer or a BASIC program in the controller.

```
10     !file APPTB14
20     !
30     ASSIGN @Source TO 709
40     DIM Func$[1],Timebase$[20]
50     REAL X
60     INTEGER Chnl,Tb_chnl,N_points
70     !                        (Chnl+100 will be used for the external timebase)
80     Chnl=0               ! output chan: 0 for CHANA, 100 for CHANB
90     Func$="V"            ! V for voltage or I for current (The array must be
100                         !   re-scaled to a maximum range of +-.1 for "I")
110    Tb_chnl=Chnl+100     ! external timebase channel
120    Timebase$="1E-3 S"   ! Timebase:  "frequency H" for Hz, <= 1E6
130                         !            "time_increment S" for seconds, > 1E-6
140    N_points=10          ! number of points
150    !
160    CLEAR @Source
```

```
170   OUTPUT @Source;"INTEGER I;DIM A0(";N_points;")"     ! Set up an array of
180   OUTPUT @Source;"SUB DEF_ARRAY"                       !   N_points & fill it
190   OUTPUT @Source;"  FOR I=0 TO ";N_points-1            !   with a 2Vpp
200   OUTPUT @Source;"    A0(I)=SIN(I*";2*PI/N_points;")" !   sinewave.
210   OUTPUT @Source;"  NEXT I"
220   OUTPUT @Source;"SUBEND;CALL DEF_ARRAY"
230   OUTPUT @Source;"DRIVETB1 HIGH"                     ! assure TB1 is inactive
240   OUTPUT @Source;"USE ";Chnl;";TRIGIN TB1"           ! output at TB1's rate
250   OUTPUT @Source;"APPLY DCMEM"&Func$;N_points;",A0"! set to 1st point of A0
260   OUTPUT @Source;"USE ";Tb_chnl;";SYNCOUT TB1"       ! put ext timebase on TB1
270   ENTER Timebase$;X
280   IF POS(UPC$(Timebase$),"H") THEN
290     OUTPUT @Source;"FREQ ";X                         ! frequency in Hz
300   ELSE
310     OUTPUT @Source;"FREQ ";1/X                       ! time increment in S
320   END IF
330   OUTPUT @Source;"APPLY ACV 1"                       ! generate SYNCOUT
340   END
```

A typical oscilloscope display for channel A follows. Note that the sine wave consists of ten steps, with each step lasting 10 μsec (**Timebase$** input value).

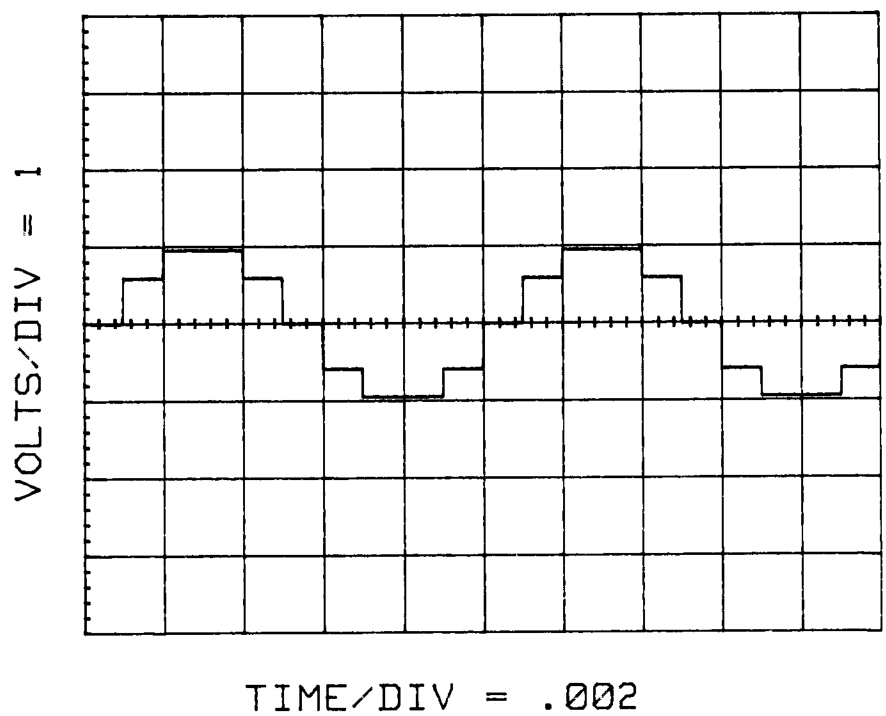


Figure 14-3. Example: Sine Waveform - External Time Base

Applying Waveform - Using Subroutine

You can also use an HP 3245A subroutine to generate the output. The output will be the same as for Example 14-4, but (unfortunately) will include a 0.1 to 0.5 msec pause every 10 msec to update the system clock and generate the time base. The minimum time increment with this method is about 0.5 msec. An example program follows.

Example 14-5: Waveform <2048 Pts - Using Subroutine (APPSB14)

This program also generates a 10-point sine waveform from channel A, but uses an HP 3245A subroutine to generate the input trigger source. The output is the same as that for Example 14-4 (see Figure 14-3), but contains a 0.5 millisecond pause every 10 milliseconds to update the HP 3245A clock. Minimum time increment using this method is about 0.5 msec.

```
10   !file APPSB14
20   !                              an internal sub generates the TRIGINs
30   ASSIGN @Source TO 709
40   DIM Func$[1],Timebase$[20]
50   REAL X,For_sgl_wait0
60   INTEGER Chnl,N_points
70   !
80   Chnl=0                 ! output chan: 0 for CHANA, 100 for CHANB
90   Func$="V"              ! V for voltage or I for current (The array must be
100                         !   re-scaled to a maximum range of +-.1 for "I")
110  Timebase$="1E-3 S"     ! Timebase:  "frequency H" for Hz, <= 1E6
120                         !            "time_increment S" for seconds, > 1E-6
130  N_points=10            ! number of points
140  !
150  CLEAR @Source
160  OUTPUT @Source;"DIM A0(";N_points;")"                ! Set up an array of
170  OUTPUT @Source;"SUB DEF_ARRAY"                       !   N_points & fill it
180  OUTPUT @Source;"  FOR I=0 TO ";N_points-1           !   with a 2Vpp
190  OUTPUT @Source;"    A0(I)=SIN(I*";2*PI/N_points;")" !   sinewave.
200  OUTPUT @Source;"  NEXT I"
210  OUTPUT @Source;"SUBEND;CALL DEF_ARRAY"
220  OUTPUT @Source;"USE ";Chnl
230  !
240  OUTPUT @Source;"INTEGER I;WAIT_TIME=.00001"
250  OUTPUT @Source;"SUB TRIGINS"
260  OUTPUT @Source;"  FOR I=1 TO 1000"
270  OUTPUT @Source;"    TRIGIN SGL"
280  OUTPUT @Source;"    WAIT WAIT_TIME"
290  OUTPUT @Source;"  NEXT I"
300  OUTPUT @Source;"SUBEND"
310  OUTPUT @Source;"MEM T1;TIME;CALL TRIGINS;MEM T2;TIME;FETCH (T2-T1)/1000"
320  ENTER @Source;For_sgl_wait0
330  ENTER Timebase$;X
340  IF POS(UPC$(Timebase$),"H") THEN
350    OUTPUT @Source;"WAIT_TIME=";.00001-For_sgl_wait0+1/X ! freq in Hz
360  ELSE
```

```
370      OUTPUT @Source;"WAIT_TIME=";.00001-For_sgl_wait0+X    ! time incr in S
380    END IF
390    !
400    OUTPUT @Source;"SUB TRIGINS"
410    OUTPUT @Source;"  WHILE 1"
420    OUTPUT @Source;"    FOR I=1 TO 1000"
430    OUTPUT @Source;"      TRIGIN SGL"
440    OUTPUT @Source;"      WAIT WAIT_TIME"
450    OUTPUT @Source;"    NEXT I"
460    OUTPUT @Source;"  END WHILE"
470    OUTPUT @Source;"SUBEND"
480    OUTPUT @Source;"APPLY DCMEM"&Func$;N_points;",A0"! set to 1st point of A0
490    OUTPUT @Source;"RUN TRIGINS"
500    END
```

Applying Waveform - Using End Interpolation

Another way to apply a waveform from an array which has less than 2048 specified elements is to convert the small array into a new array with 2048 elements. First, the small number of specified points are distributed across the (new) 2048-point array, leaving gaps of undefined elements.

Then, the undefined points are defined by interpolating between the defined points. This method requires no external time base and maintains "on-the-fly" frequency, amplitude, and DC offset control, as well as all triggering modes (**TRIGMODE**). An example program follows.

Example 14-6: Waveform <2048 Pts - End Interp (APPIN14)

This program generates a 10-point sine wave on channel A, then expands it across a 2048-element array, interpolates the defined points to defined the remaining elements, and applies the result on channel A using the channel A internal timebase. The internal timebase can be programmed as a frequency (in Hz) or as a time increment (in seconds).

The program has four inputs: (1) **Wave_old** is the array to be processed; (2) **W_old_max** is the number of elements in the array; (3) **Timebase$** is a string containing the timebase frequency (Hz) or time increment (sec); and (4) **Return_to_start**.

The normal state of **Return_to_start** is **Return_to_start=1**, which forces the undefined points at the end of the new array, **Waveform**, to be defined by "end interpolation" between the last point of the array to be processed, **Wave_old**, and the "start" or first point of **Wave_old**. Thus, all points in the array to be processed are given equal time-weighting and the output can be generated continuously without a large discontinuity when transitioning from the final point to the first point.

Setting **Return_to_start=0** will expand and interpolate such that the first and last points of **Wave_old** become the first and last points of **Waveform**.

NOTE

1. In this program, the array to be processed is loaded with a sine wave (by a FOR ... NEXT loop), but the array could have been data generated by a digitizer. One cycle of the waveform will be generated in **W_old_max** *time increments.*

2. The program **EXP_SUB** *contains this same expand and interpolate algorithm in subprogram form.*

```
10    !file APPIN14
20    !
30    ! Return_to_start=0:Expand such that the first and last points of Wave_old
40    !                           become the first and last points of Waveform.
50    ! Return_to_start=1:Expand such that the undefined points at the end of
60    !                Waveform are defined by interpolation between the last
70    !            point of Wave_old and the "start" or first point of Wave_old.
80    OPTION BASE 1
90    REAL E1,E2,E3,Scale_x,Const_x,Delta_x
100   INTEGER W_min,W_max,W_old_min,W_old_max,P1,P2,P3,I,Last_point
110   INTEGER W_o_max,Return_to_start
120   DIM Defined$[1],Undefined$[1],Timebase$[20]
130   Return_to_start=1   ! 1 enables return-to-start on Waveform(W_max+1)
140   W_old_min=1         ! 1st element in Wave_old, the old array (must be 1)
150   W_old_max=10        ! last element in Wave_old (must be >W_old_min)
160   Timebase$="1E-3 S"  ! Timebase: "frequency H" for Hz, "time S" for seconds
170   W_min=1             ! first and last element in Waveform, the output
180   W_max=2048          !   waveform array (must be 1 and 2048 for HP3245A)
190   !
200   W_max=W_max+Return_to_start       ! Add an extra point if Return_to_start.
210   W_o_max=W_old_max+Return_to_start ! Add point (compute it later)
220   ALLOCATE Waveform(W_max),Waveform$[W_max]
230   ALLOCATE Wave_old(W_o_max),Wave_old$[W_o_max]
240   Defined$="@"        ! These are used in Waveform$ and Wave_old$,
250   Undefined$="."      !   to mark whether an element is defined or not.
260   Waveform$=RPT$(Undefined$,W_max)           ! Waveform$ is undefined now.
270   Wave_old$=RPT$(Undefined$,W_o_max)         ! Wave_old$ defined below.
280   !*******************************************
290   FOR I=W_old_min TO W_old_max               ! Define Wave_old (must start
300     Wave_old(I)=SIN(2*PI*(I-1)/W_old_max)    !   at W_old_min=1).
310     Wave_old$[I;1]=Defined$                  ! Mark the points defined.
320   NEXT I                                     ! (This example is a sinewave)
330   !*******************************************
340   IF Return_to_start THEN                    ! If Return_to_start, extra
350     Wave_old(W_o_max)=Wave_old(W_old_min)    !   point is set equal to 1st
360     Wave_old$[W_o_max;1]=Defined$            !   point and marked defined.
370   END IF
380   Scale_x=(W_max-W_min)/(W_o_max-W_old_min)  ! compute x-axis scale factor
390   Const_x=W_min-Scale_x*W_old_min            ! compute x-axis offset
400   IF Scale_x<1 THEN
410     BEEP 200,.5
420     DISP "Data will be compressed and resolution may be reduced."
```

```
430   END IF
440   P2=W_old_min                         ! Set first point to be processed.
450   E2=W_min                             ! Set first point in new array.
460   P3=P2+POS(Wave_old$[P2+1,W_o_max],Defined$) ! next defined point
470   E3=P3*Scale_x+Const_x                !        next location in new array
480   Waveform(W_min)=Wave_old(W_old_min)  ! Stretch the first & last point of
490   Waveform$[W_min;1]=Defined$          !   Wave_old to the first & last
500   Waveform(W_max)=Wave_old(W_o_max)    !   point of the new array Waveform.
510   Waveform$[W_max;1]=Defined$          !   Mark both points defined.
520   IF Scale_x<=1 THEN
530     WHILE P2<W_o_max                   ! If Scale_x<1 then data is
540       P1=P2                            !   actually being compressed
550       E1=E2                            !   from more to fewer points.
560       P2=P1+POS(Wave_old$[P1+1,W_o_max],Defined$)
570       E2=P2*Scale_x+Const_x
580       P3=INT(E2)-INT(E1)
590       IF P3 THEN
600         Waveform(INT(E2))=Wave_old(P2)+(Wave_old(P2)-Wave_old(P1))/(E1-E2)*F
610         Waveform$[INT(E2);1]=Defined$
620       END IF
630     END WHILE                          ! Otherwise data will be expanded.
640   ELSE
650     WHILE P3<W_o_max                   ! P1, P2, and P3 index Wave_old, the
660       P1=P2                            !   original waveform.  P2 is the
670       P2=P3                            !   present point being processed.
680       P3=P2+POS(Wave_old$[P2+1,W_o_max],Defined$) ! E1, E2, and E3 index
690       E1=E2                            !   Waveform, the expanded waveform.
700       E2=E3                            !   E2 is the present point.
710       E3=P3*Scale_x+Const_x
720       Delta_x=FRACT(E2)                ! Delta_x is the roundoff error.
730       IF Delta_x<.5 THEN               ! The slope is between P1 and P2.
740         Waveform(E2)=Wave_old(P2)+Delta_x*(Wave_old(P1)-Wave_old(P2))/(E2-E1
750       ELSE                             ! The slope is between P2 and P3.
760         Waveform(E2)=Wave_old(P2)+(1.-Delta_x)*(Wave_old(P3)-Wave_old(P2))/(
770       END IF
780       Waveform$[E2;1]=Defined$         ! Mark the point defined.
790     END WHILE
800   END IF
810   ! Expansion is now complete.  The data is stored in Waveform (first, last,
820   !                                and some intermediate points are defined).
830   ! Now interpolate Waveform.
840   P2=W_min                                          ! P1 is just before the
850   P1=P2-1+POS(Waveform$[P2+1,W_max],Undefined$)     !   1st undefined point.
860   IF P1=P2-1 THEN GOTO Apply_waveform     ! Apply if all points are defined.
870   P2=W_max                                ! Last_point is just after th last
880   Last_point=P2+1-POS(REV$(Waveform$[P1,P2-1]),Undefined$) ! undefined pnt.
890   LOOP                                    ! P2 is the point just before the
900     P2=P1+POS(Waveform$[P1+1,Last_point],Undefined$)-1 ! 1st undefined pnt.
910   EXIT IF P2<P1                           ! Apply when all pnts are defined.
920     P1=P2                                           ! The points from P1+1 to
930     P2=P1+POS(Waveform$[P1+1,Last_point],Defined$) !   P2-1 are now
940     Scale_x=(Waveform(P2)-Waveform(P1))/(P2-P1)     !   undefined.  They will
950     Const_x=Waveform(P1)-P1*Scale_x                 !   be defined here by
960     FOR I=MAX(W_min,P1+1) TO MIN(W_max,P2-1)        !   interpolating between
```

```
970       Waveform(I)=I*Scale_x+Const_x              !   P1 and P2, which are
980     NEXT I                                     !   defined points.
990     P1=P2                            ! Set P1 to the next defined point.
1000  END LOOP
1010 Apply_waveform:REDIM Waveform(W_max-Return_to_start)  ! Remove extra point
1020  OUTPUT 709;"USE 0;DIM A0(2047);FILL A0 ";Waveform(*) !  if Return_to_start
1030  ENTER Timebase$;Scale_x                          ! Decode Timebase$.
1040  IF POS(UPC$(Timebase$),"H") THEN
1050    OUTPUT 709;"FREQ ";Scale_x/W_old_max           ! frequency in Hz
1060  ELSE
1070    OUTPUT 709;"FREQ ";1/(W_old_max*Scale_x)       ! time increment in S
1080  END IF
1090  OUTPUT 709;"APPLY WFV 1,A0"                      ! amplitude is 1Vpp
1100  END
```

Figure 14-4 shows a typical oscilloscope display of the channel A output when "end interpolation" is used. The output is a sine wave of 10 linear segments which are the result of interpolating between the ten vertices (the ten points of the sine wave specified in **Wave_old**).

Since **Return_to_start=1** was enabled, there is a smooth ramp from **Wave_old** point 10 (-0.588 volts) back to the starting point, **Wave_old** point 0, (0 Volts). Note that each segment lasts 1 msec which is the **Timebase$** input.

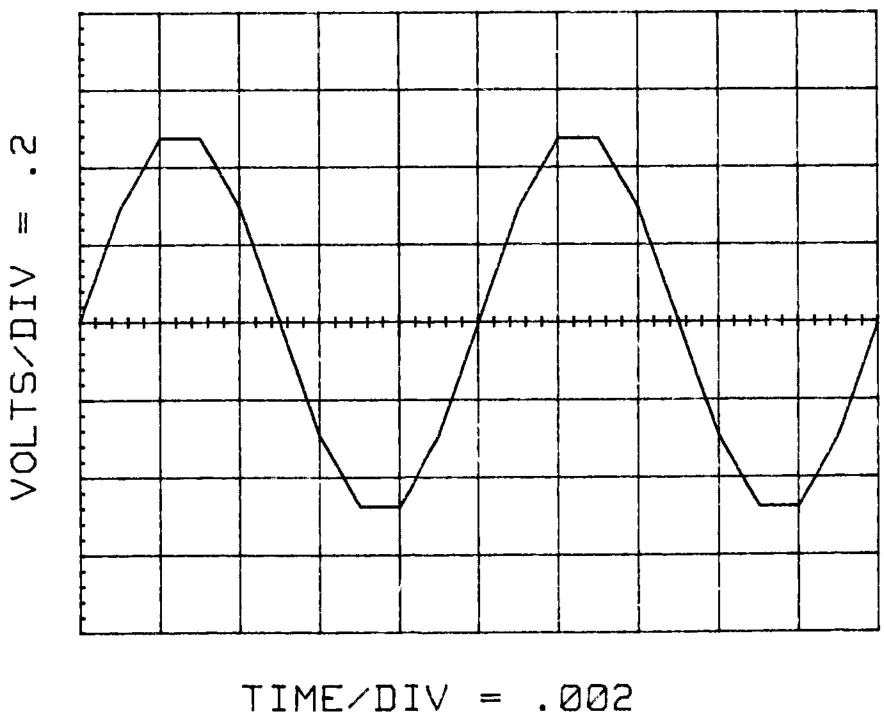


Figure 14-4. Example: Waveform <2048 Pts - End Interpolation

Example 14-7: Waveform <2048 Pts - No Interp (APPNI14)

To run the program in Example 14-6 without "end interpolation", **Return_to_start=0** will be set. (The revised program is stored on the Example Programs: Chapter 14 disc under file name APPNI14). When APPNI14 is run, note that the ten segments are still displayed.

However, the last point from the array to be processed was given a time-weight of zero and it returned to the starting point instantly. There are still 10 points specified at 1 msec per point, so the repetition period of the output is still 0.01 seconds, but each segment now lasts approximately 0.00111 seconds, as shown in Figure 14-5

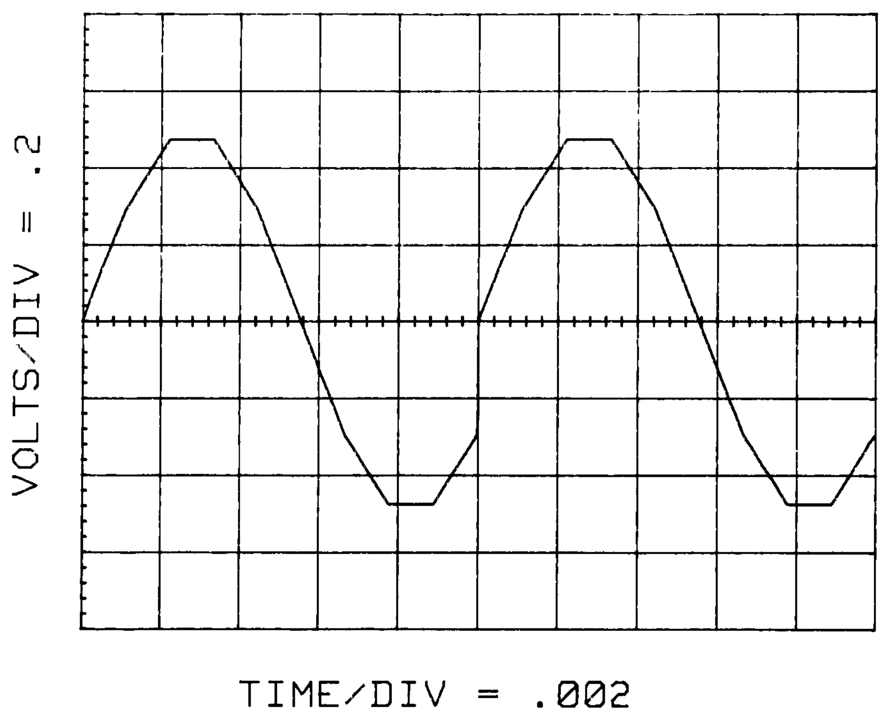


Figure 14-5. Example: Waveform <2048 Pts - No End Interpolation

Example 14-8: Electrocardiogram Waveform (APPEG14)

The program in Example 14-6 is also useful in applications where the waveform is to be specified by a small number of points with unequal time between points. The following segment of the program in Example 14-6 was replaced to demonstrate how to replace the FOR .. NEXT loop which defined the sinewave with data points to represent an electrocardiogram. **W_old_max** is changed to 20 and **Timebase&** to 0.04 seconds. Figure 14-6 shows a typical (electrocardiogram) waveform for the revised program.

```
150   W_old_max=20          !Last element in Wave_old (must be >W_old_min)
160   Timebase$=".04  S"    !Timebase: "frequency H" for Hz, "time S" for sec

280   !*******************************************
290   !    1    2    3    4   5  6    7  8   9   10   11  12  13 14 15   16
300   DATA -.06,-.05,-.01,.07,0.,-.25,1.,-.6,-.1,-.06,-.05,0.,.1,.0,-.04,-.05
310   FOR I=1 TO 16
320     READ Wave_old(I)         !Define first 16 points.  Rest are defined
321     Wave_old$[I;1]=Defined$ !by interpolating between 16 and 1.
322   NEXT I
330   !*******************************************
```

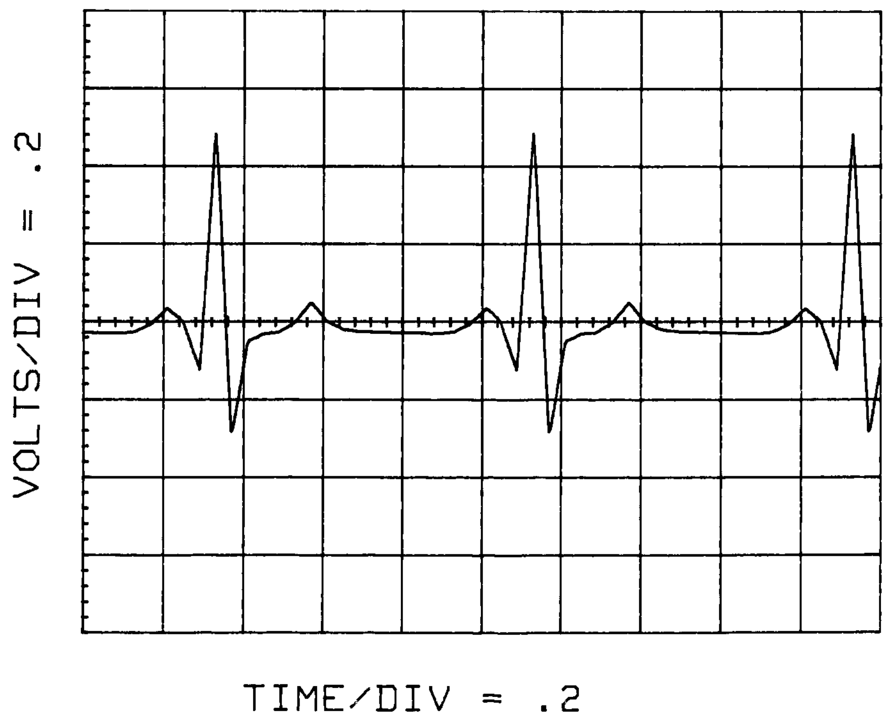


Figure 14-6. Example: Electrocardiogram Waveform

Increasing Sweep Speeds

The HP 3245A has three commands (**FASTFREQ**, **FASTAMP**, and **TRIGFREQ**) which can be used to change amplitude, dc offset, and frequency much faster than **APPLY**, **DCOFF**, or **FREQ**. However, these commands eliminate features like query and **MON STATE** display updating, autoranging, parameter value checking and rounding, and automatic use channel.

NOTE

APPLY ACV, **APPLY SQV**, **APPLY RPV**, *or* **APPLY WFV** *must be executed before* **FASTFREQ**, **FASTAMP**, *or* **TRIGFREQ** *is executed for the first time.*

Fast Frequency Changes (FASTFREQ)

FASTFREQ *ch, freq_mHz*, will change sine, square, ramp, or arbitrary frequency in less than 150 µsec when used in a subroutine. *ch* = 0 for channel A or 100 for channel B, while *freq_mHz* is a real frequency in mHz (the fractional part is truncated). Two example programs using **FASTFREQ** follow.

Example 14-9: Fast Frequency Sweep - Sine Wave (SWSIN14)

An HP 3245A subroutine can be used to vary waveform attributes in "real time" to generate unique waveforms. This program uses **FASTFREQ** to sweep frequency for a sine wave (the program can also be used to sweep amplitude or DC offset). This method of sweeping frequency, amplitude, and offset is available for sine, ramp, square, and arbitrary waveforms.

In the program, since **FASTFREQ** requires that frequency be specified in mHz, F=10000 TO 1000000 STEP 100 (line 40) sweeps the 1 Vac PP sine waveform from 10 Hz to 1 kHz in 0.1 Hz steps. The speed of this program is 300 µsec per iteration.

```
10    !file SWSIN14
20    !
30    OUTPUT 709;"SUB SWEEP"
40    OUTPUT 709;"  FOR F=10000 TO 1000000 STEP 100"
50    OUTPUT 709;"    FASTFREQ 0,F"
60    OUTPUT 709;"  NEXT F"
70    OUTPUT 709;"SUBEND;USE 0;APPLY ACV 1;CALL SWEEP"
80    END
```

Example 14-10: Frequency Sweep - Arb Wave (SWARB14)

The sweep can be arbitrarily defined by predefining a frequency array (such as a sine wave) and accessing it from the sweep subroutine as shown in the following program. Using **FASTFREQ** increases the speed to 200 μsec per iteration. Again, remember the frequency must be specified in mHz when **FASTFREQ** is used.

```
10    !file SWARB14
20    !
30    OUTPUT 709;"INTEGER I"
40    OUTPUT 709;"DIM FARRAY(99)"
50    OUTPUT 709;"SUB DEFINEF"
60    OUTPUT 709;"  FOR I=0 TO 99"
70    OUTPUT 709;"    FARRAY(I)=100.+10.*SIN(6.2832*I/100)" ! 100Hz +- 10Hz
80    OUTPUT 709;"  NEXT I"
90    OUTPUT 709;"SUBEND;CALL DEFINEF"
100   OUTPUT 709;"SUB ARBSWEEP"
110   OUTPUT 709;"  I=-1"
120   OUTPUT 709;"  WHILE 1"
130   OUTPUT 709;"    I=(I+1) MOD 100" ! increment: 0, 1, 2, ..., 99, 0, 1, ...
140   OUTPUT 709;"    FREQ FARRAY(I)"
150   OUTPUT 709;"  END WHILE"
160   OUTPUT 709;"SUBEND;USE 0;APPLY RPV 1;RUN ARBSWEEP" ! sinusoidal FM
170   END                              ! stop ARBSWEEP with ABORT or CLEAR
```

Fast Amplitude/Offset Changes (FASTAMP)

For sine, ramp, or arbitrary waveforms, use **FASTAMP** *ch, amplitude, offset* to change the waveform amplitude and/or offset in less than 90 μsec from a subroutine. Use *ch* = 0 for channel A or 100 for channel B. *amplitude* is an integer from 0 (nominally 0 amplitude) to 1777 (amplitude nominally equal to 100% of range) and *offset* is an integer from -1319 (DC offset nominally equal to -50% of peak-to-peak range) to 1319 (+50% of range).

For square waveforms, use **FASTAMP** *ch, high_level, low_level* to change squarewave levels at the same speed. Use *ch* = 0 for channel A or 100 for channel B. *high_level* is an integer from -1682 (nominally equal to -50% of peak-to-peak range) to 1682 (+50% of range) and *low_level* is an integer from 1612 (nominally equal to -50% of peak-to-peak range) to -1612 (+50% of range). Refer to "Triggered Frequency Changes (TRIGFREQ)" for an example program using **FASTAMP**.

Triggered Frequency Changes (TRIGFREQ)

Use **TRIGFREQ** *ch, freq_mHz* to set up a change of sine, square, ramp, or arbitrary frequency in less than 140 μsec on the use channel. The frequency change occurs on the next high-to-low transition of the input trigger (**TRIGIN**) source for the use channel. Use *ch* = 0 for channel A or 100 for channel B. *freq_mHz* is a real frequency in mHz (the fractional part is truncated). The input trigger (**TRIGIN**) source must be inactive (high level) when **TRIGFREQ** is executed.

To enter triggered-frequency mode, first select the desired AC waveform function with **APPLY ACV**, **APPLY SQV**, **APPLY RPV**, or **APPLY WFV**. Next, execute **TRIGFREQ** with an initial frequency value. Then, use a second **APPLY** (to the same function) or a **FASTAMP** command. (If the second **APPLY** is **APPLY WFV** or **APPLY WFI**, the array must NOT be respecified.) Three example programs to generate triggered frequency sweeps follow.

Example 14-11: Triggered Sweep - Amp/Off/Freq (SWAOF14)

This program sets up a subroutine which sweeps amplitude, DC offset, and frequency from predefined arrays on both channel A and channel B. The precise phase relationship between channels A and B is maintained throughout the sweep by using **TRIGFREQ** to change the frequency of both channels simultaneously when **TRIGIN SGL** is executed.

If **FREQ** or **FASTFREQ** were used, some phase-shifting between the two channels would occur since the frequencies would not be updated at precisely the same time. Note that the last few lines measure and display the time per iteration (800 µsec for this program).

```
10    !file SWAOF14
20    !
30    ASSIGN @Source TO 709
40    !
50    N_points=3000                              ! number of points to sweep
60    !
70    OUTPUT @Source;"RESET;SCRATCH"             ! RESET assures sync
80    OUTPUT @Source;"DIM F0(";N_points;")"      ! freq array
90    OUTPUT @Source;"INTEGER AMP0(";N_points;")"  ! CHANA amplitude array
100   OUTPUT @Source;"INTEGER AMP1(";N_points;")"  ! CHANB amplitude array
110   OUTPUT @Source;"INTEGER OFF0(";N_points;")"  ! CHANA offset array
120   OUTPUT @Source;"INTEGER OFF1(";N_points;")"  ! CHANB offset array
130   OUTPUT @Source;"INTEGER I,A"               ! integer index for speed
140   !
150   OUTPUT @Source;"SUB DEF_ARRAYS"            ! define arrays
160   OUTPUT @Source;"  FOR I=1 TO ";N_points
170   OUTPUT @Source;"    F0(I)=I*100."       ! linear sweep to 100*N_points Hz
180   OUTPUT @Source;"    AMP0(I)=888+888.*I/";N_points!ramps from 50 to 100%rng
190   OUTPUT @Source;"    AMP1(I)=888"               ! 50% of range
200   OUTPUT @Source;"    OFF0(I)=0"                 ! zero offset
210   OUTPUT @Source;"    OFF1(I)=660.*I/";N_points ! ramps from 0 to 50% of rng
220   OUTPUT @Source;"  NEXT I"
230   OUTPUT @Source;"SUBEND;CALL DEF_ARRAYS"
240   !
250   OUTPUT @Source;"USE 0;TRIGIN HIGH;TRIGOUT EXT"     ! assure TRIGIN HIGH
260   OUTPUT @Source;"FREQ 0;TRIGMODE ARMWF;APPLY ACV 1" ! hold at zero phase
270   OUTPUT @Source;"USE 100;TRIGIN EXT"                ! (HIGH is inactive)
280   OUTPUT @Source;"FREQ 0;TRIGMODE ARMWF;APPLY RPV 1"
290   OUTPUT @Source;"TRIGFREQ   0,F0(1)"    ! set up TRIGFREQ mode (TRIGIN must
300   OUTPUT @Source;"TRIGFREQ 100,F0(1)"    !   be inactive for this command)
310   OUTPUT @Source;"FASTAMP   0,AMP0(1),0" ! FASTAMP or APPLY must be used
```

```
320   OUTPUT @Source;"FASTAMP 100,AMP1(1),0" !   here to enable TRIGFREQ
330   OUTPUT @Source;"USE 0"                  ! TRIGIN SGL will come from CHANA
340   OUTPUT @Source;"SUB SWEEPA"                      ! define SWEEPA subroutine
350   OUTPUT @Source;"  FOR I=1 TO ";N_points          ! sweep across all points
360   OUTPUT @Source;"    FASTAMP   0,AMP0(I),0"       ! update CHANA amp & off
370   OUTPUT @Source;"    FASTAMP 100,AMP1(I),0"       ! update CHANB
380   OUTPUT @Source;"    TRIGFREQ   0,F0(I)"          ! set up triggered freqs
390   OUTPUT @Source;"    TRIGFREQ 100,F0(I)"          !   for triggerring
400   OUTPUT @Source;"    TRIGIN SGL"                  ! change frequencies
410   OUTPUT @Source;"  NEXT I"
420   OUTPUT @Source;"SUBEND"
430   !
440   T1=TIMEDATE
450   OUTPUT @Source;"SWEEPA"
460   T2=TIMEDATE
470   DISP (T2-T1)/N_points                            ! display iteration time
480   STOP
490   END
```

Example 14-12: Frequency Hopping - Using Subroutine (SWTSB14)

This program sets up a subroutine which changes frequency (based on a sequence stored in a pre-defined array) on the falling edge of a hardware input signal into the TB0 port. After a new frequency is triggered, the next frequency is automatically loaded. The triggers are applied externally to the TB0 port on the rear panel.

NOTE

The **DEFKEY** *statement (line 380) redefines NUMERIC key* **0** *so that the user can generate required triggers from channel B by pressing the* **0** *key on the front panel.*

```
10    !file SWTSB14
20    !
30    INTEGER N_points,I
40    N_points=100
50    !
60    OUTPUT 709;"RESET;SCRATCH;DIM F(";N_points;")"
70    OUTPUT 709;"INTEGER I,J"
80    OUTPUT 709;"SUB DEF_ARRAY"
90    OUTPUT 709;"  FOR I=0 TO ";N_points-1
100   OUTPUT 709;"    F(I)=(I+1)*10000."     ! F from 10 to 10*N_points Hz
110   OUTPUT 709;"  NEXT I"
120   OUTPUT 709;"SUBEND;CALL DEF_ARRAY"
130   OUTPUT 709;"USE 0"                     ! Use Channel A.
140   OUTPUT 709;"TRIGIN HIGH"               ! Assure TRIGIN HIGH.
150   OUTPUT 709;"APPLY SQV 2.5"             ! Select AC function.
160   OUTPUT 709;"TRIGFREQ 0,F(1)"           ! Set up TRIGFREQ mode and F(1).
```

```
170   OUTPUT 709;"APPLY SQV 2.5"               ! APPLYs must bracket TRIGFREQ.
180   OUTPUT 709;"TRIGIN TB0"                  ! Now take TRIGIN from TB0.
190   OUTPUT 709;"SUB FREQHOP"
200   OUTPUT 709;"  I=1"                       ! Presently at 1st point.
210   OUTPUT 709;"  WHILE 1"                   ! Begin endless loop.
220   OUTPUT 709;"    WAITFOR TB0 HL"          ! Wait for TB0 falling edge.
230   OUTPUT 709;"    I=(I+1) MOD ";N_points   ! incr: 1, 2, .., N_points, 1, ...
240   OUTPUT 709;"    TRIGIN HIGH"             ! Assure TRIGIN HIGH,
250   OUTPUT 709;"    TRIGFREQ 0,F(I)"         ! Then set up CHANA's next freq.
260   OUTPUT 709;"    TRIGIN TB0"              ! Return to TRIGIN TB0,
270   OUTPUT 709;"    WAITFOR TB0 LH"          ! To prevent false triggers wait
280   OUTPUT 709;"  END WHILE"                 !   for rising edge.clear 709
290   OUTPUT 709;"SUBEND"
300   OUTPUT 709;"RUN FREQHOP"
310 !
320 !       User key 0 uses Channel B to generate a 100 Hz square wave on TB0.
330   OUTPUT 709;"DEFKEY 0,""USE 100;SYNCOUT TB0;FREQ 100;APPLY ACV 1"""
340   LOCAL 709                                ! DEFKEY DEFAULT resets keys
350   END
```

Example 14-13: Triggered Sweep - Ext Triggering (SWTXT14)

A running subroutine can execute segments of its code based on hardware events or system times. This program sets up a background subroutine which waits until TB0 goes HIGH and then sweeps frequency from 500 Hz to 1000 Hz. The program then waits until TB0 goes LOW and sweeps frequency from 1000 Hz back to 500 Hz. The signal on TB0 can be supplied externally on the TB0 port or internally from channel B by pressing NUMERIC key **0** followed by the **Enter** key. This sequence uses channel B to generate a 2 Hz square wave on TB0.

```
10    !file SWTXT14
20    !
30    OUTPUT 709;"RESET;SCRATCH;INTEGER I"
40    OUTPUT 709;"USE 0"                       ! Use Channel A.
50    OUTPUT 709;"SUB TSWEEP"
60    OUTPUT 709;"  WHILE 1"                   ! Begin endless loop.
70    OUTPUT 709;"    MEM I"                   ! Read the level of TB0
80    OUTPUT 709;"    TB0?"                    !   in to I.
90    OUTPUT 709;"    WHILE (I=0)"             ! Wait in this loop until I=1.
100   OUTPUT 709;"      MEM I"
110   OUTPUT 709;"      TB0?"
120   OUTPUT 709;"    END WHILE"
130   OUTPUT 709;"    FOR F=50000 TO 100000 STEP 100"
140   OUTPUT 709;"      FASTFREQ 0,F"          ! Ramp freq from 500 to 1000 Hz
150   OUTPUT 709;"    NEXT F"                  !   in .1 Hz steps.
160   OUTPUT 709;"    WHILE (I=1)"             ! Wait in this loop until I=0.
170   OUTPUT 709;"      MEM I"
180   OUTPUT 709;"      TB0?"
190   OUTPUT 709;"    END WHILE"
200   OUTPUT 709;"    FOR F=100000 TO 50000 STEP -100"
210   OUTPUT 709;"      FASTFREQ 0,F"          ! Ramp freq from 1000 to 500 Hz.
```

```
220   OUTPUT 709;"    NEXT F"
230   OUTPUT 709;"  END WHILE"              ! Continue endless loop.
240   OUTPUT 709;"SUBEND"
250   OUTPUT 709;"APPLY ACV 1;RUN TSWEEP"   ! Apply sinewave and RUN.
260 !
270 !      User key 0 uses Channel B to generate a 2 Hz square wave on TB0.
280   OUTPUT 709;"DEFKEY 0,""USE 100;SYNCOUT TB0;FREQ 2;APPLY ACV 1"""
290   LOCAL 709                             ! DEFKEY DEFAULT resets User keys.
300   END
```

Replacing the **WHILE** loop and **TB0?** queries with the following segment will cause the program to wait in the loop until the system timer exceeds the variable **T_CHECK**.

```
xxx   OUTPUT 709;"MEM TIME_NOW"
xxx   OUTPUT 709;"TIME"
xxx   OUTPUT 709;"WHILE (TIME_NOW<T_CHECK)"
xxx   OUTPUT 709;"  MEM TIME_NOW"
xxx   OUTPUT 709;"  TIME"
xxx   OUTPUT 709;"END WHILE"
```

Manual Supplement

HP 3245A Option 002 10x Voltage Amplifier

Introduction

This supplement introduces you to the HP 3245A Option 002 10x Voltage Amplifier and gives an overview of its configuration and operation. The supplement is designed to be used in conjunction with the *HP 3245A Operating and Programming Manual* which is also included with your instrument. It includes the following information:

- Product Overview
- Front/Rear Panel Description
- Selecting the Input Signal to the Amplifier
- Connecting Cabling

To get a better understanding of the instrument, we recommend that you read chapters 1 and 2 in the manual *before* reading this supplement. Without a basic understanding of the HP 3245A, the information in this supplement may not be thoroughly understood. Refer to the remaining chapters in the manual to learn all the details required to take advantage of the entire instrument feature set.

Product Overview

The HP 3245A Universal Source is a high-precision source which combines precise dc, accurate ac, and flexible arbitrary waveform generation in a single instrument. The HP 3245A's Voltage Amplifier gives you the capability to output dc, ac, and arbitrary voltages 10 times greater than the standard HP 3245A source (up to ± 102.5 volts peak). When the dc or ac *current* function is selected, the voltage amplifier is disabled.

The voltage amplifier is capable of driving loads up to ± 40 mA. The output resistance is less than 0.5Ω and overshoot is less than 5%. The dc gain accuracy is 0.03% and the dc offset is about 2 mV (adjustable to zero). The amplifier's bandwidth makes it useful to at least 100 kHz. Combining these capabilities of the voltage amplifier with the precision and resolution of the HP 3245A results in a versatile and cost-effective signal source for many applications.

Note

For complete product specifications for the HP 3245A, see Appendix A.

The voltage amplifier consists of a single printed circuit board which is factory-installed in place of Channel B in the HP 3245A mainframe. The easiest way to understand the voltage amplifier is to think of it as an amplifier with a gain of 10 connected to the Channel A generator (Figure 1). Whatever voltage you program Channel A to output (up to ± 10.25 volts peak) is amplified by 10 and directed to the 10x V Output connector.

The voltage amplifier has no programmable features of its own; its operation is controlled by Channel A. To program the voltage amplifier to output a voltage, send the **APPLY** command to Channel A. For example, to output 45 volts dc from the voltage amplifier, program Channel A to output 4.5 volts dc (**APPLY DCV 4.5**). The two voltages are output simultaneously to the Channel A Output and 10x V Output connectors. In addition, the output terminals selected for Channel A are also used by the voltage amplifier. For example, if you select the rear terminals for Channel A (**TERM REAR**), the voltage amplifier outputs are also directed to the rear panel.

To protect the voltage amplifier from external high-energy sources, 1.5 ampere fuses are placed in series with the front and rear output terminals. A spare 1.5 ampere fuse is included on the voltage amplifier's printed circuit board.

If an external source greater than ± 125 volts peak is connected to the 10x V Output connector, internal protection relays will open to isolate the voltage amplifier. After approximately two seconds, the protection relays will be reconnected. If the overload condition is still present, the protection relays open again. This will result in a faint clicking sound that you will hear from inside the HP 3245A. Instrument damage may result if this overload condition is allowed to exist for an extended period of time. See the instrument specifications in Appendix A for time limitations.

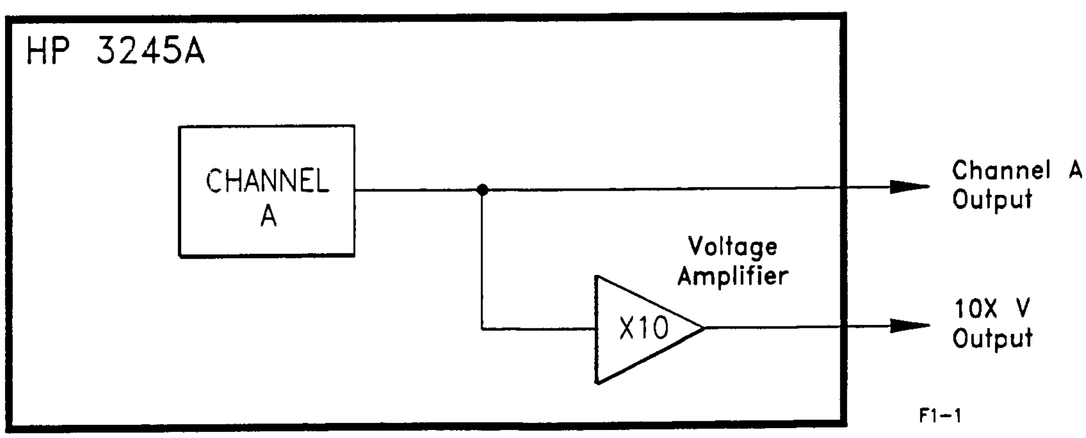


Figure 1. HP 3245A Voltage Amplifier Block Diagram

Front/Rear Panel Description

This section gives an overview of the front and rear panels of the HP 3245A with the Option 002 amplifier installed. Please take a few minutes to get acquainted with the panel functions before attempting to use the voltage amplifier. See also chapter 2 for additional information on the front and rear panel features.

Front Panel Features

As shown in Figure 2, the front panel consists of a power switch, a display window, three groups of pushbutton keys (FUNCTION/RANGE, MENU, and NUMERIC/USER), and a set of input/output connectors. The display window shows the output voltage or current from Channel A, commands ready to be entered, or you can use it to monitor the state of Channel A. The presence of the voltage amplifier is not recognizable from the display window and is not directly controlled by the front panel keys. The output level shown on the display is the level of Channel A *only*.

Use the front panel keys to enter commands, parameters, and numeric data for Channel A. The input/output connectors provide input or output terminals for signal outputs and trigger pulses. See chapters 6 through 14 for additional operating information.

Note

The Chan A/Chan B key is not functional with Option 002 installed.

Rear Panel Features

As shown in Figure 3, the rear panel consists of a set of input/output connectors, the HP-IB interface connector, the line power connector, the line fuse holder, and the line voltage selector switch. The input/output connectors provide similar capability to those connectors located on the front panel. See chapters 6 through 14 for additional operating information.

Display

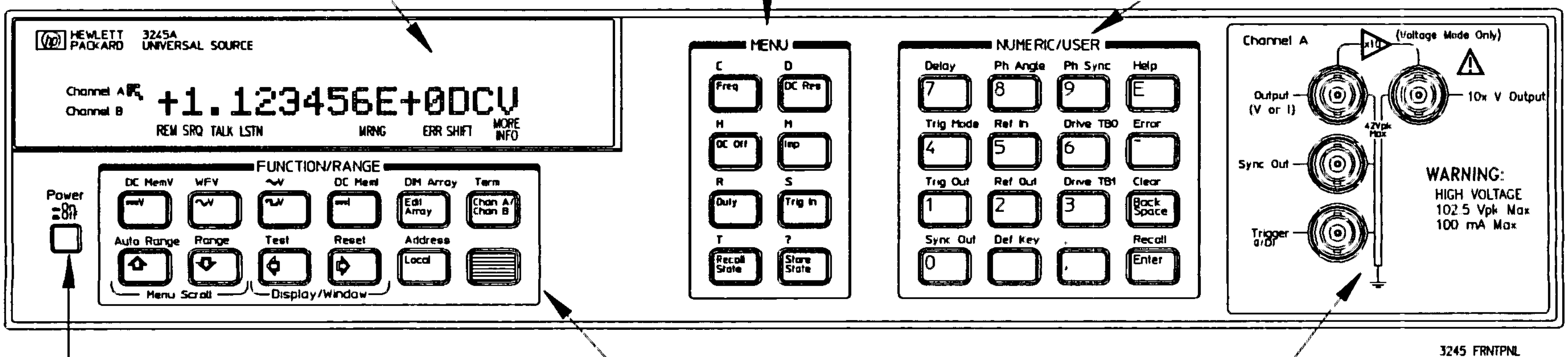
Displays output voltage/current or commands to be entered. The annunciators indicate HP-IB status (REM, SRQ, TALK, and LISTEN), fixed range vs. autorange (MRNG), error condition (ERR), shifted keyboard (SHIFT), and additional information to be scrolled onto the display (MORE INFO).

Menu Keys

Use these keys to select waveform parameters such as frequency, DC resolution and offset, output impedance, and duty cycle. Also, to set the trigger input source and store and recall channel states. In shifted mode, provides entry point for commands to be entered from the front panel.

Numeric/User Keys

In unshifted mode, use these keys to enter numeric data and/or punctuation as required and to enter commands into the HP 3245A. In shifted mode, enter various waveform parameters an input/output port to trigger the HP 3245A and trigger modes.



On/Off Switch

Function/Range Keys

Use these keys to output AC or DC waveforms, edit stored arrays, set fixed range or autorange, test or reset the HP 3245A, or scroll commands or display information.

Note: *Chan A/Chan B key is not functional with Option 002.*

Input/Output Connectors

Channel A has an output connector to outputs its waveform, a sync output to output the sync pulse signal, and a trigger (I/O) connector for input or output triggering.

The 10X V Output (Option 002) has an output connector to output its amplified waveforms.

Figure 2. Front Panel Functions

Freq Ref Connector

When synchronized waveform mode is set, output the reference frequency from this connector.

Channel A Output and Amplified Output Connector

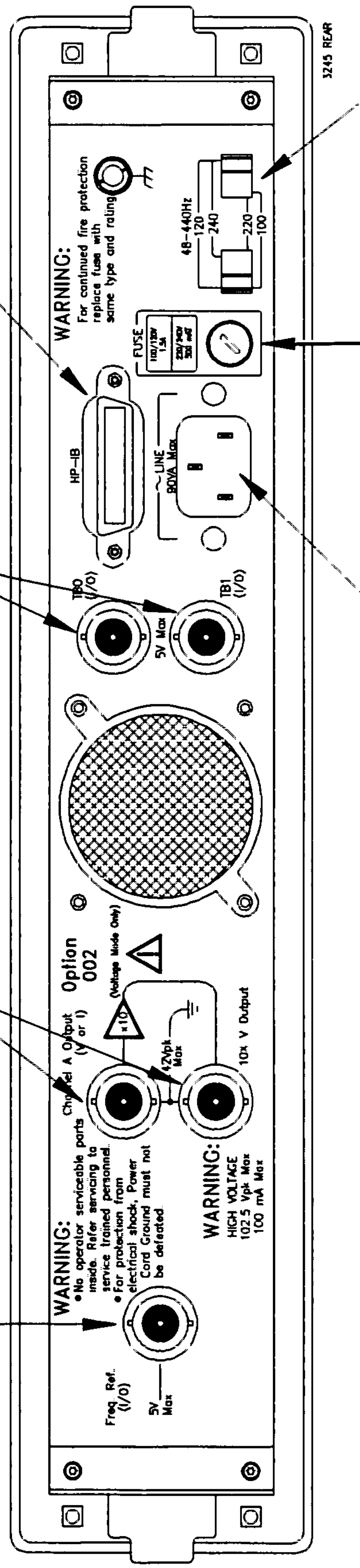
When rear panel output is selected, output the Channel A waveform from the Channel A connector and its amplified output from the 10x amplified connector. When rear panel output is enabled, waveforms are not output from the front panel Output connectors.

Trigger Bus Connectors

The TB0 (I/O) trigger bus connector provides an input/output port to trigger the HP 3245A via the TB0 trigger bus. The TB1 (I/O) trigger bus connector provides access to the TB1 trigger bus.

HP-IB Connector

Connection point for the HP-IB cable from your controller.



Line Power Connector

Connector for AC power cord. Use the proper cord.

Fuse Holder

Contains 1.5A (@ 100V/120V) or 500 mAT (@ 200/240V) fuse.

Line Selector Switch

Switch to set the HP 3245A to the appropriate line voltage (100 V AC, 120 V AC, 220 V AC, or 240 V AC).

Figure 3. Rear Panel Functions

Selecting the Input Signal to the Amplifier

A jumper (J201) is provided on the voltage amplifier's printed circuit board to select the input signal to the amplifier (Figure 4). The jumper is located on the top side of the 03245-66503 printed circuit board.

The "50 OHM" position is the factory default. This setting assures that the voltage amplifier's output is always 10 times greater than the Channel A output. If you set the Channel A output impedance to 50Ω (using the **IMP** command), both the Channel A output and the voltage amplifier output will vary with external Channel A load resistance.

If you desire to have the voltage amplifier's output voltage independent of the Channel A external load (assuming **IMP 50**), move the jumper to the "0 OHM" position. The voltage level of the voltage amplifier will be 10 times the *open circuit* voltage of the Channel A output.

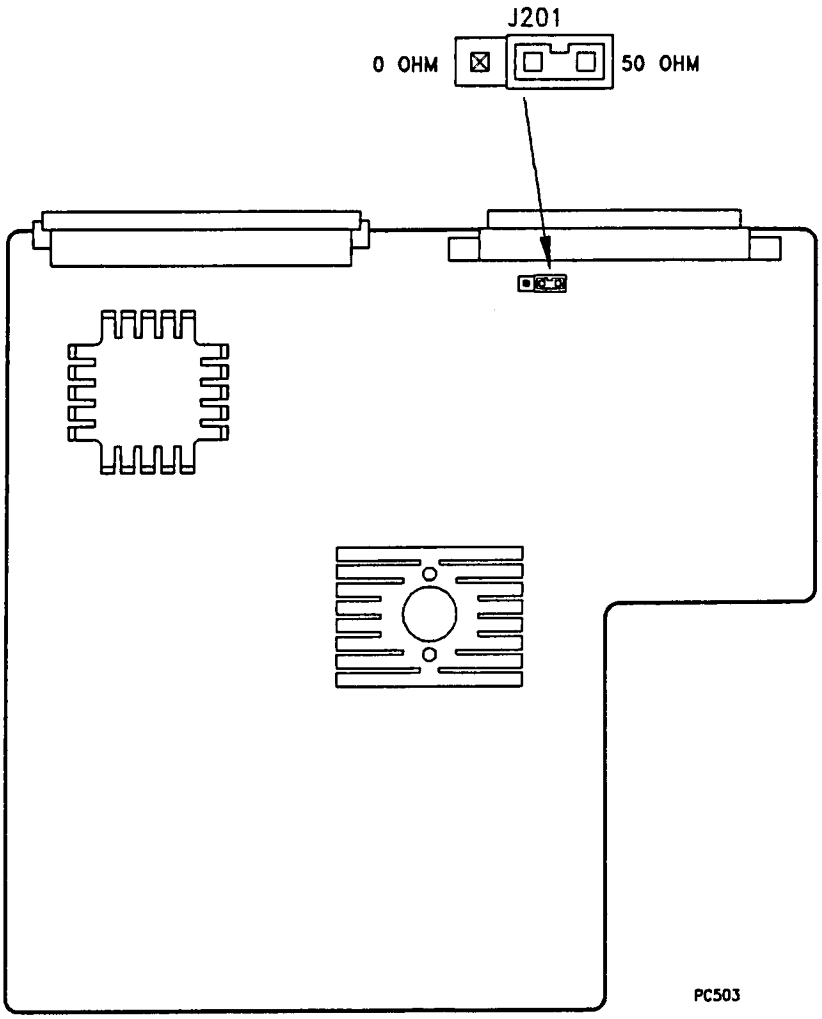


Figure 4. Setting the Input Signal Jumper

Connecting Cabling

After you have completed the HP 3245A installation procedures described in chapter 2, you can connect your BNC cables to the front and/or rear panel connectors as required. The procedure for connecting cabling to the voltage amplifier is the same as for the standard HP 3245A. Make your connection to the 10x V Output just as you would to Channel A. See chapter 2 for complete details on connecting your cabling.

CAUTION

The 10x V Output BNCs are specified to source up to ±40 mA and are current limited at approximately ±100 mA. Application of voltages greater than ±102.5 volts peak external to the output terminals may open the output relays. Each 10x V Output contains output protection relays. The 10x V Outputs are also fused for additional protection (a spare fuse is also included on the printed circuit board).

Appendix A
HP 3245A Specifications

This appendix lists specifications for the HP 3245A's warranted performance over the temperature range of 0°C to 55°C. Information marked by the (TYPICAL) designation is helpful in applying the instrument, but is non-warranted information.

See page A-6 for the Option 002 10x High Voltage Amplifier specifications.

DC Voltage

DC Volts Output Characteristics

Output Ranges

High Resolution Mode (0Ω): 1V, 10V
High Resolution Mode (50Ω): .5V, 5V
Low Resolution Mode (0Ω). .15625V, .3125V, .625V, 1.25V, 2.5V, 5V, 10V
Low Resolution Mode (50Ω). .078125V, .15625V, .3125V, .625V, 1.25V, 2.5V, 5V

Maximum Programmed Output

High Resolution Mode (0Ω). 1.25V on 1V range, 10.25V on 10V range
High Resolution Mode (50Ω): .625V on .5V range, 5.125V on 5V range
Low Resolution Mode. Maximum programmed output equal to output range

Output Resistance

0Ω mode: <.5Ω
50Ω mode: 49.9Ω to 50.6Ω

Resolution

High Resolution Mode

Range	0Ω mode	50Ωmode
1 V	1 μV	.5 μV
10 V	10 μV	5 μV

Low Resolution Mode

Range	0Ω mode	50Ω mode
.078125 V	N/A	40 μV
.15625 V	79 μV	79 μV
.3125 V	157 μV	157 μV
625 V	313 μV	313 μV
1.25 V	625 μV	625 μV
2.5 V	1250 μV	1250 μV
5.0 V	2.5 mV	2.5 mV
10 V	5.0 mV	N/A

Noise & Spurious Response (DC to 30 Hz) (typical)

Mode	Range	Peak-to-Peak	RMS
High Resolution	10 V	50 μV	10 μV
	1 V	7 μV	1.6 μV
Low Resolution	All	003% of range	0007% of range

Noise & Spurious Response (30 Hz to 250 kHz) (maximum)

Mode	Peak-to-Peak	RMS
High Resolution	2 mV	.6 mV
Low Resolution	.2% of range + 2 mV	.02% of range + 60 μV

Current Compliance

100 mA on all ranges

Settling Time (DELAY 0)

High Resolution Mode (% of step)		Low Resolution Mode (% of step)	
.1%	20 mSec	.1% (0Ω):	100 μSec
.001%	40 mSec	(50Ω):	25 μSec
(1Sec if function changed)		.5% (50Ω):	5 μSec

Overshoot

High Resolution Mode: < 5% of step + 15% of range
Low Resolution Mode: < 30% of step + 2% of range

Triggered DC

Up to 2,048 voltage levels can be stored for sequential output. A new level will be output with each trigger input. The sequence will start over after the last level is output. Trigger frequency can be up to 1MHz.

DC Volts Accuracy

± (%of programmed output + Volts), 0Ω impedance mode, <10 Hz noise, >1MΩ load. Tcal is the temperature of calibration from 18°C to 28°C. One hour warm-up

24 Hour: Tcal ±1°C

High Resolution Mode

Range	Accuracy
10 V	.00071% + 85 µV
1 V	00077% + 15 µV

Low Resolution Mode

Range	Accuracy
10 V	.087% + 20 mV
5 V	.087% + 9.7 mV
2.5 V	.087% + 4.9 mV
1.25 V	.087% + 2.5 mV
625 V	.087% + 1.3 mV
.3125 V	087% + .67 mV
.15625 V	087% + 37 mV

90 Day: Tcal ±5°C

High Resolution Mode

Range	Accuracy
10 V	.0038% + 180 µV
1 V	.0042% + 31 µV

Low Resolution Mode

Range	Accuracy
10 V	17% + 37 mV
5 V	17% + 19 mV
2.5 V	.17% + 9.2 mV
1.25 V	.17% + 4.6 mV
.625 V	.17% + 2.5 mV
.3125 V	.17% + 1.3 mV
.15625 V	.17% + .73 mV

1 Year: Tcal ±5°C

High Resolution Mode

Range	Accuracy
10 V	.0054% + 212 µV
1 V	0060% + 37 µV

Low Resolution Mode

Range	Accuracy
10 V	.20% + 43 mV
5 V	.20% + 22 mV
2.5 V	.20% + 11 mV
1.25 V	.20% + 5.6 mV
.625 V	20% + 2.9 mV
.3125 V	20% + 1 6 mV
.15625 V	20% + 88 mV

Temperature Coefficient:

± (% of programmed output + Volts)/°C

High Resolution Mode

Range	Accuracy
10 V	.00031% + 12 µV
1 V	.00037% + 1.8 µV

Low Resolution Mode

Range	Accuracy
10 V	011% + 2400 µV
5 V	.011% + 1200 µV
2.5 V	.011% + 600 µV
1.25 V	.011% + 300 µV
.625 V	.011% + 160 µV
.3125 V	.011% + 82 µV
.15625 V	.011% + 50 µV

Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature ranges specified above.

DC Current

DC Current Characteristics

Output Ranges (for both High & Low Resolution)

1 mA, 1 mA, 10 mA, 100 mA

Maximum Programmed Output is equal to the output range value on all ranges.

Output Resistance

Range	High Resolution	Low Resolution
100 mA	>3.1 MΩ	>.32 MΩ
10 mA	>210 MΩ	> 41 MΩ
1 mA	>2500 MΩ	> 480 MΩ
1 mA	>10000 MΩ	> 4800 MΩ

Resolution

Range	High Resolution	Low Resolution
.1 mA	.1 nA	50 nA
1 mA	1 nA	500 nA
10 mA	10 nA	5 µA
100 mA	100 nA	50 µA

Noise & Spurious Response (DC to 30 Hz) (typical)

Mode	Peak-to-peak	RMS
High Resolution	.0002% of range	.00004% of range
Low Resolution	.004% of range	0007% of range

Noise & Spurious Response (30 Hz to 250 kHz) (maximum)

Mode	Peak-to-peak	RMS
High Resolution	.02% of range + .7µA	.003% of range + 70 nA
Low Resolution	.04% of range + .7µA	01% of range + 70 nA

Voltage Compliance

± 8.0 V on 100 mA range

±10.0 V on all other ranges

Max Resistive Load for specified accuracy = compliance/range in amps

Settling Time (DELAY 0)

High Resolution Mode
- .1% of step: 20 mSec
- 001% of step: 40 mSec
- .003% of step on 10 mA range: 40 mSec
- .01% of step on 100 mA range. 40 mSec
  (1Sec if function changed)

Low Resolution Mode
- .1% of step: 100 µSec
  (30 mSec if range changed)

Overshoot

High Resolution Mode: <5% of step + .15% of range

Low Resolution Mode: <30% of step + 2% of range

Triggered DC

Up to 2048 current levels can be stored for sequential output. A new level output will be output with trigger input. The sequence will start over after the last level is output. Trigger frequency can be up to 1 MHz.

DC Current Accuracy

± (% of programmed output + amps), <10 Hz noise, < Rout/10^6 Ω load.
Tcal is the temperature of calibration from 18°C to 28°C.

24 Hour: Tcal ±1°C

After 1 hour warm-up. Accuracy relative to calibration standard.

High Resolution Mode

Range	Accuracy
100 mA	.0120% + 2.1 µA
10 mA	.0031% + 98 nA
1 mA	.0010% + 9.5 nA
.1 mA	.0009% + 1.9 nA

Low Resolution Mode

Range	Accuracy
100 mA	.16% + 200 µA
10 mA	.16% + 30 µA
1 mA	.15% + 1.90 µA
.1 mA	.15% + 190 µA

90 Day: Tcal ±5°C

After 1 hour warm-up.

High Resolution Mode

Range	Accuracy
100 mA	0202% + 3 3 µA
10 mA	0074% + 220 nA
1 mA	0052% + 20 nA
1 mA	0052% + 3.3 nA

Low Resolution Mode

Range	Accuracy
100 mA	.32% + 400 µA
10 mA	.30% + 52 µA
1 mA	.25% + 3.8 µA
.1 mA	.25% + .38 µA

1 Year: Tcal ±5°C

After 1 hour warm-up

High Resolution Mode

Range	Accuracy
100 mA	.0244% + 3.8 µA
10 mA	.0094% + 270 nA
1 mA	.0073% + 25 nA
.1 mA	.0072% + 3.9 nA

Low Resolution Mode

Range	Accuracy
100 mA	.35% + 470 µA
10 mA	.33% + 61 µA
1 mA	.29% + 4.6 µA
.1 mA	.29% + .46 µA

Temperature Coefficient: Tcal ±1°C

± (% of programmed output + amps)/°C

High Resolution Mode

Range	Accuracy
100 mA	.00070% + 150 nA
10 mA	.00036% + 14 nA
1 mA	.00036% + 1.2 nA
.1 mA	.00036% + .16 nA

Low Resolution Mode

Range	Accuracy
100 mA	.011% + 25 µA
10 mA	.019% + 2.8 µA
1 mA	.011% + 240 nA
.1 mA	.011% + 24 nA

Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature range specified above.

AC Voltage

AC Volts Output Characteristics

Sine, square, ramp, arbitrary.

Frequency Range

0 to 1 MHz for sine, arbitrary and square (at 50% duty cycle)
0 to 100 kHz for ramp
0 to 100 kHz for square with duty cycle not equal to 50%.

Frequency Resolution

0.001 Hz

Frequency Accuracy

± 50 ppm, 18 to 28°C

Frequency Temperature Coefficient

± 1 ppm/°C

Phase Offset

Range: -360 to +360 degrees
Resolution: <.001 degree
Accuracy: (typical)
100Hz: ± 0.01 degree
1 kHz: ± 0.1 degree
10 kHz: ± 1.0 degree
100 kHz: ± 10 degrees
1 MHz: ± 100 degrees

Peak-to-Peak Output Ranges (into 50Ω load):

50Ω mode: .15625V, .3125V, .625V, 1.25V, 5V, 10V
0Ω mode: .3125V, .625V, 1.25V, 5V, 10V, 20V
Maximum programmed peak-to-peak output is equal to the output range on all ranges

Output Resistance

0Ω mode: <.5Ω
50Ω mode: 49.9Ω to 50.6Ω

Current Compliance

100 mA on all ranges

Amplitude and/or Offset Resolution

Range (Peak-to-peak)	Resolution (50Ω Mode)	Resolution (0Ω Mode)
.15625 V	79 µV	N/A
.3125 V	157 µV	157 µV
.625 V	313 µV	313 µV
1.25 V	625 µV	625 µV
2.5 V	1250 µV	1250 µV
5 V	2.5 mV	2.5 mV
10 V	5.0 mV	5.0 mV
20 V	N/A	10.0 mV

Amplitude can be set from 10% to 100% of range

DC offset can be set such that the peak AC value plus the magnitude of the DC offset does not exceed range/2.

Arbitrary waveforms are defined over the unitless range from -1 to 1 with $^{1}/_{2048}$ resolution.

Sine, Ramp (<10 kHz) and Arbitrary Waveform Abberations (50Ω Mode)

< 1% of pk-pk value measured with amplitude > 50% of range

Ramp Linearity to 1 kHz (50Ω Mode)

.3% of pk-pk value measured at 50% duty cycle from 10% to 90% point

Ramp Duty Cycle Range

5% to 95% with <.1% resolution

Sinewave Characteristics (50Ω Mode)

Frequency	Harmonic and Spurious Levels (ampl ≥50% of range)*	THD (ampl ≥50% of range)	Flatness in reference to 1 kHz
< 3 kHz	< -62 dB	< -56 dB	07 dB
to 10 kHz	< -62 dB	< -50 dB	07 dB
to 30 kHz	< -55 dB	< -48 dB	07 dB
to 100 kHz	< -46 dB	< -46 dB	.20 dB
to 300 kHz	< -40 dB	–	.60 dB
to 1 MHz	< -40 dB	–	2.0 dB

Additional fixed spurious response > 4 MHz: 500 μ Vrms

Squarewave Characteristics (50Ω Mode)

Risetime. <250 nSec, 10% to 90%
Settling Time. < 1 μSec to 1% of amplitude
Overshoot: < 5% of Peak-to-Peak amplitude
Duty Cycle Range: 5% to 95%, 0 to 100 kHz,
Fixed at 50% above 100 kHz
Duty Cycle Accuracy: ± (0.8% of period + 120 nSec)
Duty Cycle Resolution.<.1%

Arbitrary Waveform Characteristics

Depth: 2048 points
Amplitude Resolution. 12 bits
Smoothing Filter. 1.25 MHz, 5 pole
Sample Rate. 4.295 MHz
Waveform Repetition Rate: 0 to 1 MHz
Frequency Resolution: .001 Hz
Up to 14 waveforms can be stored

AC Volts Amplitude or Offset Accuracy [1]

Sine, square, ramp, arbitrary: ± (% of programmed output + % of range) at 1 kHz, 50Ω mode into >1 MΩ, <200 pF with amplitude programmed to ≥ 10% of range. The output is specified into >1 MΩ in order to eliminate the effects of the 50Ω output resistance error on accuracy. Tcal is the temperature of calibration from 18°C to 28°C. One hour warm-up.

24 Hour: Tcal ±1°C

Accuracy relative to calibration standard. All ranges [2,3]
Sine, ramp, arbitrary waveforms: .16% output + .25% range
Square waveforms: .084% output + .42% range

90 Day: Tcal ± 5°C

All ranges [2,3]
Sine, ramp, arbitrary waveforms. .29% output + .36% range
Square waveforms. .29% output + .52% range

1 Year: Tcal ± 5°C

All ranges [2,3]
Sine, ramp, arbitrary waveforms. .35% output + .37% range
Square waveforms. .39% output + .56% range

Temperature Coefficient

Range	Sine, ramp, arbitrary Waveforms	Square Waveforms
15625 V	.0180% + 25 μV	.034 + 24 μV
.3125 V	0180% + 50 μV	.034 + 47 μV
.625 V	0180% + 100 μV	.034 + 94 μV
1.25 V	0180% + 200 μV	034 + 188 μV
2.5 V	.0180% + 400 μV	034 + 375 μV
5 V	.0180% + 800 μV	.034 + 750 μV
10 V	.0180% + 1.6 mV	.034 + 1.5 mV

Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature ranges specified above.

[1] The accuracy specifications must be applied once to amplitude and again to offset.

[2] Ranges are peak-to-peak value of amplitude for 50 Ω mode into 50 Ω load

[3] In the 50 Ω output impedance mode, the AC amplitude (Apply ACV) and DC offset (DCOFF) commands assume the termination impedance is identical to the source impedance nominally 50 Ω. The amplitude and offset accuracy specifications also assume that the source and the load impedances are matched. When the load is ≥1 MΩ, the actual amplitude and offset voltages which appear across the load will be twice the programmed values. In this instance, both the percent of output and the percent of range errors must be doubled.

For example, if a 5V peak-to-peak sinewave, with 2.5V of DC offset, was programmed with the source in the 50 Ω mode, and the output was applied across a 1 MΩ load, the following 90 Day amplitude and offset accuracies can be calculated:

Amplitude = 2 [(programmed value) ± (0.29% of output + 0.36% of range)]
= 2 [(5V) ± ((0.0029 x 5V) + (0.0036 x 10V))]
= 10V ± 0.101V

Offset = 2 [(programmed value) ± (0.29% of output + 0.36% of range)]
= 2 [(5V) ± ((0.0029 x 2.5V) + (0.0036V x 10V))]
= 5V ± 0.087V

AC Current

AC Current Output Characteristics (Typical)

Sine, square, ramp, arbitrary

Frequency Range

0 to 100 kHz for sine and arbitrary
0 to 10 kHz for ramp and square

Frequency Resolution

.001 Hz

Frequency Accuracy

± 50 ppm, 18 to 28°C

Peak-to-Peak Output Ranges

.2 mA, 2 mA, 20 mA, 200 mA

Maximum programmed Peak-to-Peak Output is equal to the output range on all ranges.

Output Resistance

Range	Resistance
200 mA	>.32 MΩ
20 mA	> 41 MΩ
2 mA	> 480 MΩ
.2 mA	> 4800 MΩ

Voltage compliance

± 8.0 V on 200 mA range
± 10.0 V on all other ranges

Max Resistive Load for specified accuracy = compliance/range in amps

Amplitude and/or Offset Resolution

Range	Resolution
200 mA	100 µA
20 mA	10 µA
2 mA	1 µA
.2 mA	.1 µA

Amplitude can be set from 5% to 100% of range.

DC offset can be set such that the peak AC value plus the magnitude of the DC offset does not exceed range/2.

Arbitrary waveforms are defined over the unitless range from -1 to 1 with $^1/_{2048}$ resolution.

Sine, Ramp (<10 kHz) and Arbitrary Waveform Aberations (Typical)

<1% of p-p measured with amplitude >50% of range

Ramp Linearity to 1 kHz (Typical)

.3% of p-p value measured at 50% duty cycle from 10% to 90% point

Ramp Duty Cycle Range

5% to 95% with < 1% resolution

Arbitrary Waveform Characteristics

Depth: 2048 points
Amplitude Resolution: 12 bits
Typical 3 dB BW: 100 kHz

AC Current Amplitude or Offset Accuracy (Typical)[1]

Sine, square, ramp, arbitrary: ± (% of programmed output + % of range) at 1 kHz, with amplitude programmed to ≥ 50% of range. Tcal is the temperature of calibration from 18°C to 28°C. One hour warm-up

24 Hour: Tcal ±1°C

Accuracy relative to calibration standard. All ranges.
0.5% output + 0.4% range

90 Day: Tcal ±5°C

All ranges
1% output + 0.5% range

1 Year: Tcal ±5°C

All ranges
1.4% output + 0.5% range

Temperature Coefficient

± (% of programmed output + % of range)/°C

Range	Resolution
200 mA	.011% + 25 µA
20 mA	.019% + 2.8 µA
2 mA	.011% + 240 nA
.2 mA	.011% + 24 nA

Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature ranges specified above.

[1] *The accuracy specifications must be applied once to amplitude and again to offset*

10X High Voltage Amplifier Specifications

DC VOLTAGE OUTPUT CHARACTERISTICS

Output Ranges
Same as Channel A.

Maximum Output
Same as Channel A amplified by a factor of +10.

Output Impedance
Less than 0.5 Ω in series with the parallel combination of a 25 μH inductor, typical, and a 33.2 Ω resistor, typical.

Resolution
Same as Channel A amplified by factor of +10.

Noise & Spurious Response (DC to 30 Hz) (typical)

Mode	Range(3)	Peak-to-Peak	RMS
High Resolution	10 V	500 μV	80 μV
	1 V	70 μV	13 μV
Low Resolution	All	.03% of range(1)	.007% of range(1)

Noise & Spurious Response (30 Hz to 250 kHz) (maximum)

Mode	Peak-to-Peak	RMS
High Resolution	20 mV	6 mV
Low Resolution	2% of range(1) + 20 mV	2% of range(1) + 600 μV

Current Compliance
40 mA (0 to +40°C operating temperature)
30 mA (+40 to +50°C operating temperature)

Settling Time (DELAY 0)
High Resolution Mode (% of step)
.1% 20 mSec
.001% 40 mSec
1 Sec, if function is changed.

Low Resolution Mode (% of step)
1% 100 μSec

Output Linearity (typical)
≤ 10 ppm for 0 to ±100 volts

Overshoot
Same as Channel A.

Triggered DC
Same as Channel A.

DC Volts Accuracy
±(% of output + Volts), 0 Ω impedance mode, <10 Hz noise, >10 MΩ load. Tcal is the temperature of calibration from +18 °C to +28°C. One hour Warm-up

24 Hour: Tcal ±1° C

High Resolution Mode

Range	Output	Accuracy
10V	100V	.03% + 1.1 mV
1V	10V	.03% + 450 μV

Low Resolution Mode

Range	Output	Accuracy
10V	100V	.2% + 370 mV
5V	50V	.2% + 190 mV
2.5V	25V	.2% + 94 mV
1.25V	12.5V	.2% + 48 mV
.625V	6.25V	.2% + 27 mV
.3125V	3.125V	.2% + 15 mV
.15625V	1.5625V	.2% + 9 mV

90 Day: Tcal ±5° C

High Resolution Mode

Range	Output	Accuracy
10V	100V	.03% + 3.3 mV
1V	10V	.03% + 1.8 mV

Low Resolution Mode

Range	Output	Accuracy
10V	100V	.12% + 200 mV
5V	50V	.12% + 100 mV
2.5V	25V	.12% + 50 mV
1.25V	12.5V	.12% + 25 mV
.625V	6.25V	.12% + 14 mV
.3125V	3.125V	.12% + 7 mV
.15625V	1.5625V	.12% + 4 mV

1 Year: Tcal ±5 °C

High Resolution Mode

Range	Output	Accuracy
10V	100V	.04% + 6 mV
1V	10V	.04% + 4.4 mV

Low Resolution Mode

Range	Output	Accuracy
10V	100V	.23% + 434 mV
5V	50V	.23% + 224 mV
2.5V	25V	.23% + 114 mV
1.25V	12.5V	.23% + 60 mV
.625V	6.25V	.23% + 33 mV
.3125V	3.125V	.23% + 20 mV
.15625V	1.5625V	.23% + 13 mV

Temperature Coefficient: ±(% of output + Volts)/°C

High Resolution Mode

Range	Output	Accuracy
10V	100V	.00051% + 140 μV
1V	10V	.00057% + 38 μV

Low Resolution Mode

Range	Output	Accuracy
10V	100V	.011% + 24 mV
5V	50V	.011% + 12 mV
2.5V	25V	.011% + 6 mV
1.25V	12.5V	.011% + 3 mV
.625V	6.25V	.011% + 1.6 mV
.3125V	3.125V	.011% + 820 μV
.15625V	1.5625V	.011% + 500 μV

Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature ranges specified below.

AC VOLTAGE OUTPUT CHARACTERISTICS

Sine, Square, Ramp and Arbitrary.

Frequency Range
Same as Channel A.

Frequency Resolution
Same as Channel A.

Frequency Accuracy
Same as Channel A.

Frequency Temperature Coefficient
Same as Channel A.

Peak-to-Peak Output Ranges
Same as Channel A.

Maximum Peak-to-Peak Output
Same as Channel A amplified by a factor of +10.

Output Impedance
Less than 0.5 Ω in series with the parallel combination of a 25 µH inductor, typical, and a 33.2 Ω resistor, typical.

Current Compliance
40 mA peak (0 to +40°C operating temperature)
30 mA peak (+40 to +50°C operating temperature)

Amplitude and/or Offset Resolution
Same as Channel A amplified by a factor of +10

Ramp Duty Cycle Range
Same as channel A

Sinewave Characteristics

Frequency	Harmonic and Spurious Levels (ampl ≥50% range)*	THD (ampl ≥50% range)	Flatness in ref. to 1kHz
< 1 kHz	< -54 dB	< -49 dB	.2 dB
to 10 kHz	< -54 dB	< -44 dB	.2 dB
to 100 kHz	< -40 dB	< -37 dB	.6 dB

* Additional fixed spurious response > 4 MHz: 1 µV rms

Squarewave Characteristics
Risetime: < 3 µSec, 10% to 90%

Settling Time: < 4 µSec to 1% of amplitude
Overshoot: < 5% of peak-to-peak amplitude
Duty Cycle Range: Same as Channel A
Duty Cycle Accuracy: Same as Channel A
Duty Cycle Resolution: Same as Channel A

Arbitrary Waveform Characteristics
Same as Channel A

AC Volts Amplitude or Offset Accuracy (1)

Sine, Square, Ramp, Arbitrary: ±(% of output + % of range)(3) at 100 Hz, into >10 kΩ and <200 pF with amplitude programmed to r 10% of range. Tcal is the temperature of calibration from +18°C to +28°C. One hour warm-up

24 Hour: Tcal ±1°C
Accuracy relative to calibration standard. All ranges (2)

Sine, Ramp, Arbitrary waveforms:	.20% output + 2.5% range(3)
Squarewave:	.12% output + 4.2% range(3)

90 Day: Tcal ±5°C
All ranges (2)

Sine, Ramp, Arbitrary waveforms:	.32% output + 3.6% range(3)
Squarewave:	.32% output + 5.2% range(3)

1 Year: Tcal ±5°C
All ranges (2)

Sine, Ramp, Arbitrary waveforms:	.38% output + 3.7% range(3)
Squarewave:	.42% output + 5.6% range(3)

Temperature Coefficient
Multiply the temperature coefficient by the difference between the operating temperature and Tcal to determine the additional error for operation outside temperature ranges specified above.

Range	Sine, Ramp, Arbitrary Waveforms	Square Waveforms
15625V	018% + 250 mV	034% + 240 mV
.3125V	018% + 500 mV	.034% + 470 mV
625V	018% + 1 mV	034% + 940 mV
1.25V	.018% + 2 mV	.034% + 1.9 mV
2.5V	.018% + 4 mV	.034% + 3.8 mV
5V	.018% + 8 mV	.034% + 7.5 mV
10V	.018% + 16mV	.034% + 15 mV

(1) The accuracy specifications must be applied once to amplitude and again to offset.

(2) Ranges are peak-to-peak value of Channel A output for 50 Ω Mode into 50 Ω load. When the load is ± 1 MΩ, the actual amplitude and offset voltages which appear across the load and the input to the X10 Voltage Amplifier input will be twice the programmed value. In this instance, both the percent of output and the percent of range errors must be doubled.

For example, if a 5V peak-to-peak sinewave with 2.5V DC offset on the 10V Range (50V peak-to-peak sinewave and 25V DC offset at the X10 Voltage output) was programmed with the source in the 50 Ω mode and the Channel A output was applied across a 1 MΩ load, the following 90 Day amplitude and offset accuracies can be calculated:

Amplitude = 2[(programmed X10 Voltage Output amplitude) ±(0.32% output + 3.6% range)]

= 2[(50V)±(0.0032 x 50V + .036 x 10V)]
= 100V peak-to-peak ± 0.52V

Offset = 2[(programmed X10 Voltage Output offset) ±(.32% output + 3.6% range)]

= 2[(25V)±(0.0032 x 25V + .036 x 10V)]
= 50V ± 0.44V

(3) "Range" refers to the Channel A range.

Output Protection
Standard outputs are protected against signals up to 15 volts peak.

The X10 Voltage output is protected against signals up to ±100 volts continuously and up to ±150 volts peak for 30 Seconds max.

Index

I

K

L

M