Errata

Title & Document Type: **8770A Arbitrary Waveform Synthesizer Operating Manual**

Manual Part Number: **08770-90024**

Revision Date: **September 1968**

About this Manual

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

HP References in this Manual

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

Support for Your Product

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

www.agilent.com

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

Agilent Technologies

OPERATING MANUAL

HP 8770A
ARBITRARY WAVEFORM SYNTHESIZER

SERIAL NUMBERS

Attached to the rear panel of the instrument is a serial number label. The serial number is in the form: 0000A00000. The first four digits and the letter are the serial number prefix. The last five digits are the suffix. The prefix is the same for identical instruments; it changes only when a configuration change is made to the instrument. The suffix, however, is assigned sequentially and is different for each instrument. The contents of this manual apply directly to instruments having the serial number prefix 2540A.

HEWLETT PACKARD

1501 PAGE MILL ROAD, PALO ALTO, CALIFORNIA, U.S.A.

OPERATING MANUAL PART NO. 08770-90024
SERVICE MANUAL PART NO. 08770-90019
Microfiche Operating Manual Part No. 08770-90025
Microfiche Service Manual Part No. 08770-90020

Printed: SEPTEMBER 1988

CONTENTS

CONTENTS (cont'd)

LIST OF ILLUSTRATIONS

TABLES

EXAMPLES

SAFETY CONSIDERATIONS

GENERAL

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

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

BEFORE APPLYING POWER

Verify that the product is set to match the available line voltage and the correct fuse is installed.

SAFETY EARTH GROUND

An uninterruptible safety earth ground must be provided from the main power source to the product input wiring terminals, power cord, or supplied power cord set.

WARNINGS

Any interruption of the protective (grounding) conductor (inside or outside the instrument) or disconnecting the protective earth terminal will cause a potential shock hazard that could result in personal injury. (Grounding one conductor of a two conductor outlet is not sufficient protection.) In addition, verify that a common ground exists between the unit under test and this instrument prior to energizing either unit.

Whenever it is likely that the protection has been impaired, the instrument must be made inoperative and be secured against any unintended operation.

If this instrument is to be energized via an autotransformer (for voltage reduction) make sure the common terminal is connected to neutral (that is, the grounded side of the mains supply).

Servicing instructions are for use by service-trained personnel only. To avoid dangerous electric shock, do not perform any servicing unless qualified to do so.

Adjustments described in the manual are performed with power supplied to the instrument while protective covers are removed. Energy available at many points may, if contacted, result in personal injury.

Capacitors inside the instrument may still be charged even if the instrument has been disconnected from its source of supply.

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

SAFETY SYMBOLS

Instruction manual symbol: the product will be marked with this symbol when it is necessary for the user to refer to the instruction manual (see Table of Contents for page references).

Indicates hazardous voltages.

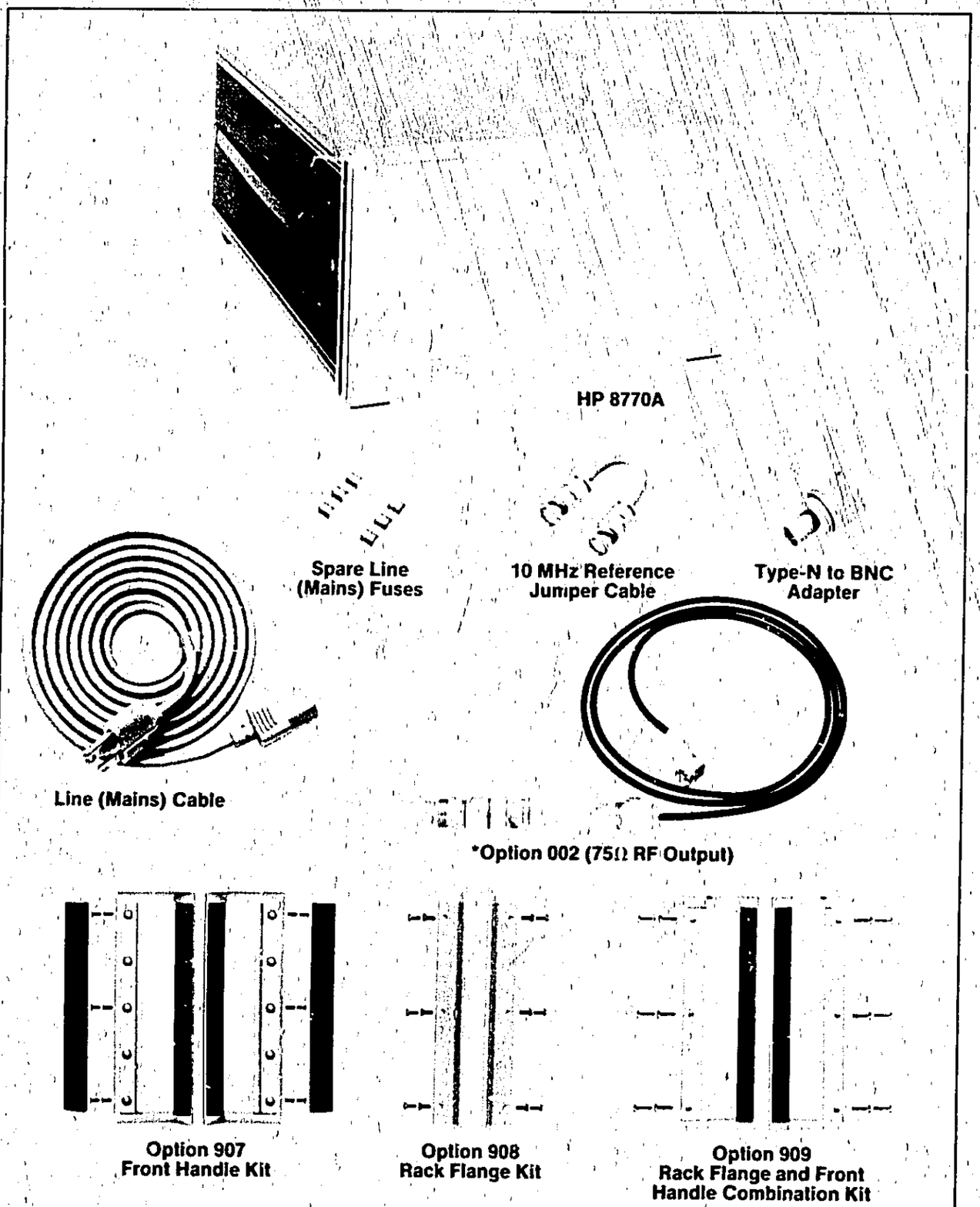
Indicates earth (ground) terminal.

WARNING

The WARNING sign denotes a hazard. It calls attention to a procedure, practice, or the like, which, if not correctly performed or adhered to, could result in personal injury. Do not proceed beyond a WARNING sign until the indicated conditions are fully understood and met.

CAUTION

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



NOTE: Refer to ACCESSORIES SUPPLIED for more details.
*If Option 002 was installed, cable and hardware shipped with instrument will be different than that pictured above. Refer to ACCESSORIES SUPPLIED paragraph e.

Figure 1-1. HP Model 8770A Arbitrary Waveform Synthesizer with Accessories Supplied and Options Available

SECTION I
GENERAL INFORMATION

1-1. INTRODUCTION

The HP 8770A Operating and Service Manual consists of an Operating Manual and a Service Manual. These two volumes contain all the information required to install, operate, test, adjust and service the Hewlett-Packard Model 8770A Arbitrary Waveform Synthesizer. Figure 1-1 shows the Synthesizer with all of its externally supplied accessories.

The Operating Manual, which is shipped with each instrument, has four sections:

Section I, General Information
Section II, Installation
Section III, Operation
Section IV, Performance Tests

The Service Manual, which is shipped with the instrument as Option 915 or ordered separately, has four sections:

Section V, Adjustments
Section VI, Replaceable Parts
Section VII, Manual Changes
Section VIII, Service

Additional copies of the Operating Manual or the Service Manual can be ordered separately through your nearest Hewlett-Packard office. The part numbers are listed on the title page of this manual.

Listed on the title page of this manual, below the manual part number, is a microfiche part number. This number may be used to order 100 x 150 millimetre (4 x 6 inch) microfilm transparencies of this manual. Each microfiche contains up to 96 photo-duplicates of the manual pages. The microfiche package also includes the latest Manual Changes supplement, as well as all pertinent Service Notes.

1-2. DESCRIPTION

The HP 8770A Arbitrary Waveform Synthesizer is a dc to 50 MHz signal generator. It has the capability of reproducing arbitrary waveforms; the waveform is reproduced using its frequency components up to 50 MHz.

Waveform data is first developed using a computer. The data is then loaded into the Synthesizer where digital data is converted into an analog output. Many different waveforms can be loaded into the Synthesizer's 128k of memory at any one time. Whole waveforms or portions of waveforms can be output in any order desirable.

The Synthesizer has a standard 125 MHz system clock; there is a rear panel connector that can accept an external clock of 60 to 130 MHz. Using a rear panel connector (10 MHz REFERENCE INPUT), the Synthesizer can be locked to an external reference.

With a very fast 12-bit Digital-to-Analog Converter (DAC), which has a 125 MHz sampling rate (with the internal clock), the Synthesizer can reproduce waveforms as fast as 8 ns per data point.

The Synthesizer has a 0 to 110 dB step attenuator which provides output power with a range of +10 dBm to −110 dBm.

The Synthesizer is compatible with the Hewlett-Packard Interface Bus (HP-IB) to the extent indicated by the following bus functions: SH1, AH1, T5 ,TE0, L3, LE0, SR1, RL1, PP1, DC1, DT0 and C0. An explanation of the compatibility code can be found in IEEE Standard 488 (1978), "IEEE Standard Digital Interface for Programmable Instrumentation" or the identical ANSI Standard MC1.1.

There is an EXTERNAL DATA INPUT connector on the rear panel. Data is loaded faster into the Synthesizer's memory using this input instead of the HP-IB input.

1-3. SPECIFICATIONS

Instrument specifications are listed in Table 1-1. These specifications are the performance standards or limits against which the instrument may be tested. Supplemental characteristics are listed in Table 1-2. Supplemental characteristics are not warranted specifications, but are typical characteristics included as additional information for the user.

1-4. SAFETY CONSIDERATIONS

This product is a Safety Class I instrument, that is, one provided with a protective earth terminal.

SAFETY CONSIDERATIONS (cont'd)
The Synthesizer and all related documentation should be reviewed for familiarization with safety markings and instructions before operation. Refer to the Safety Considerations page found at the beginning of this manual for a summary of the safety information. Safety information for installation, operation, performance testing, adjustment, and service is found in appropriate places throughout this manual.

1-5. MANUAL CHANGES SUPPLEMENT

Instruments manufactured after the printing of this manual may be different from those documented in this manual. The manual for this newer instrument is accompanied by a Manual Changes supplement. The supplement contains "change information" that explains how to adapt this manual to the newer instrument.

In addition to change information, the supplement may contain information for correcting errors in the manual. To keep the manual as current and as accurate as possible, Hewlett-Packard recommends that you periodically request the latest Manual Changes supplement. The supplement is identified with the manual print date and part number, both of which appear on the manual title page. Complimentary copies of the supplement are available from Hewlett-Packard.

1-6. OPTIONS

1-7. Electrical Options

Option 002 provides a 75 ohm output impedance instead of 50 ohms for the RF OUTPUT connector.

1-8. Mechanical Options

The following options may have been ordered and received with the Synthesizer. If they were not ordered with the original shipment and are now desired, they can be ordered from the nearest Hewlett-Packard office using the part numbers included in each of the following paragraphs.

Front Handle Kit (Option 907). Ease of handling is increased with the front panel handles. The Front Handle Kit part number is 5061-9691.

Rack Mount Flange Kit (Option 908). The Synthesizer can be solidly mounted to the instrument rack using the flange kit. The Rack Mount Flange Kit part number is 5061-9679.

Rack Mount Flange and Front Handle Combination Kit (Option 909). This is a unique part which combines both functions. It is not simply a front handle kit and a rack flange kit packaged together. The Rack Mount Flange and Front Panel Combination Kit part number is 5061-9685.

1-9. HEWLETT-PACKARD INTERFACE BUS

HP-IB

1-10. Compatibility

HP-IB is Hewlett-Packard's implementation of IEEE Standard 488 and the identical ANSI Standard MC1.1. The Synthesizer's compatibility with HP-IB is defined by the following list of interface functions: SH1, AH1, T5, TE0, L3, LE0, SR1, RL1, PP1, DC1, DT0 and C0. For more detailed information relating to remote operation of the Synthesizer, refer to Section III of this manual.

1-11. ACCESSORIES SUPPLIED

The accessories supplied with the Synthesizer are shown in Figure 1-1.

a. The line power cable is supplied in several configurations, depending on the destination of the original shipment. Refer to Power Cables in Section II of this manual.

b. Three additional fuses are shipped with instruments that are factory configured for 115 Vac operation. One fuse has a 3A rating and is for reconfiguring the instrument for 230 Vac operation. The other two fuses are spares; one fuse is for 115 volt operation and the other is for 230 volt operation.

c. A type-N to BNC adapter is connected to the RF output connector if a BNC output is desired.

d. A short gray BNC—BNC cable (86701-60063) is included, and must be connected from either 10 MHz REFERENCE OUTPUT 1 or OUTPUT 2 to the 10 MHz REFERENCE INPUT on the rear panel.

e. Cabling and hardware needed to convert the RF OUTPUT from 50Ω to 75Ω (Option 002) configuration is included with the Synthesizer. If Option 002 was ordered initially, all cabling and hardware needed to convert the RF OUTPUT from 75Ω to 50Ω is included with the Synthesizer. Refer to Section II, Installation, for retrofit procedures.

1-12. EQUIPMENT REQUIRED BUT NOT SUPPLIED

For the Synthesizer to be operational, it must be used with a controller, that has HP-IB capabilities, such as the HP 9000 Series 200 Model 236 or 216.

1-13. RECOMMENDED TEST EQUIPMENT

Table 1-3 lists the test equipment recommended for use in testing, adjusting and servicing the Synthesizer. The Critical Specification column describes the essential requirements for each piece of test equipment. Other equipment can be substituted if it meets or exceeds these critical specifications.

The Recommended Model column may suggest more than one model. The first model shown is usually the least expensive, single-purpose model. Alternate models are suggested for additional features that would make them a better choice in some applications.

Table 1-1. Specifications (1 of 2)

Electrical Specifications	Performance Limits	Conditions
INTERNAL ATTENUATOR		
Range	110 dB	
Steps	10 dB	
Accuracy	Attenuator Setting (dB) / Accuracy (dB): 10 ±0.2; 20 ±0.4; 30 ±0.5; 40 ±0.8; 50 ±0.9; 60 ±1.0; 70 ±1.2; 80 ±1.4; 90 ±1.5; 100 ±1.6; 110 ±1.8	Referenced at attenuator setting of 0 dB for all frequencies from dc to 50 MHz.
SINE WAVE PERFORMANCE		
Output Power	+10 dBm ±0.25 dB[1]	At 10 MHz, 0 dB attenuation into 50Ω for std. and 75Ω for Opt. 002
Opt. 002 (75Ω RF Output)	+8.24 dBm ±0.25 dB[1]	
		Harmonics and spurious referenced to +10 dBm while using the 125 MHz internal clock.
Harmonics	<−50 dBc	For carrier frequencies ≤10 MHz
	<−40 dBc	For carrier frequencies from >10 to 50 MHz.
Spurious and Nonharmonic Distortion	<−50 dBc	For carrier frequencies ≤25 MHz
	<−40 dBc	For carrier frequencies from >25 to 50 MHz.
OUTPUT		
Update rate[2]	125 MHz (Divisible by 2, 4, 8, 16, 32, 64, 128 or 256.)	Internal Clock
Update rate stability	Identical to reference oscillator aging rate.	

Table 1-1. Specifications (2 of 2)

Electrical Specifications	Performance Limits	Conditions
REFERENCE OSCILLATOR (10 MHz Quartz)		
Aging rate	$<5 \times 10^{-10}$/day	After a 24 hour warm-up and an oscillator off-time of less than 24 hours.(3)
HP-IB INTERFACE	IEEE STD 488-1978 Compatibility Code: SH1, AH1, T5, TE0, L3, LE0, SR1, RL1, PP1, DC1, DT0, C0.	
GENERAL		
Line Voltage	115 or 230 Vac	
Line Voltage Tolerance	115 Vac +10%, −25% 230 Vac +10%, −15%	
Line Frequency	48—66 Hz	
Temperature:		
Specification Range	+15°C to +40°C	
Operating Range	0°C to +55°C	
Power Dissipation	445 VA	
Weight:		
Net	23.6 kg (52 lb.)	
Dimensions:(4)		
Height	235 mm (9.25 in.)	
Width	426 mm (16.75 in.)	
Depth	622 mm (24.5 in.)	
ELECTROMAGNETIC COMPATIBILITY		
Electromagnetic Interference	Conducted and radiated interference is within the requirements of methods VDE 0871 level B and FTZ 1046/1984.	

(1) Specification includes 0.11 dB measurement uncertainty and 0.1 dB for drift over temperature range.
(2) Accuracy is dependent on the internal 10 MHz reference oscillator. The internal 125 MHz clock is phased locked to the 10 MHz reference oscillator. When the external clock (60 to 130 MHz) is used, accuracy is dependent on the external clock used.
(3) If the oscillator has been off for 0 to 24 hours, a warm-up period of 24 hours is required to obtain an aging rate less than 5×10^{-10}/day. If the off-time has been greater than 24 hours, such as for shipping or instrument storage, typically 48 hours are required to reach the specified aging rate.
(4) When rack mounting or ordering cabinet accessories the module sizes are 8-3/4H, 1MW (module width) and 23D.

Table 1-2. Supplemental Characteristics (1 of 2)

Supplemental characteristics are intended to provide information useful in applying the instrument by giving nominal and typical, but non-warranted, performance parameters.

OUTPUT

Impedance: 50Ω (75Ω for Option 002)

Source SWR: <1.5:1

DC Drift: <±2.5% of full scale output voltage within specification temperature range.

Connector: Type N (female); standard instrument
BNC (female); Option 002.

Rise Time (10—90%): <8 ns.

Peak Output Voltage:

Attenuator Setting (dB)	Equivalent Voltage (Peak-to-Peak)*
0	2V
10	630 mV
20	200 mV
30	63 mV
40	20 mV
50	6.3 mV
60	2.0 mV
70	630 μV
80	200 μV
90	63 μV
100	20 μV
110	6.3 μV

*Into 50 ohms for a standard instrument; into 75 ohms for an Option 002 instrument.

Number of DAC bits: 12

Resolution: 0.024% of full scale output voltage at frequencies >1 kHz.

SINE WAVE PERFORMANCE

Output Power: +10 dBm to −110 dBm

Dynamic Range: 72 dB (plus the 110 dB step attenuator.)

Single Sideband Phase Noise: <−120 dBc/Hz at 10 kHz offset from 10 MHz carrier.

Amplitude Stability: ± 0.3% (0.013 dB) in 1 hour after 24 hour warm-up.

Spurious and Nonharmonic Distortion: <−60 dBc for carrier frequencies ≤25 MHz;
<−50 dBc for carrier frequencies >25 MHz to 50 MHz.

Out of Band Spurs (for a carrier of dc to 50 MHz): <−50 dBm (plus the attenuator setting) or
<−100 dBm, whichever level is greater.

Amplitude Flatness: ± 0.8 dB (Total instrument flatness including the attenuator.)

Two-Tone Intermodulation Distortion (124 kHz separation): <−60 dBc.

Phase Linearity: ±5°

INTERNAL MEMORY

Length: 131 072 12-bit words

Element: A 12-bit word representing a waveform sample point.

Table 1-2. Supplemental Characteristics (2 of 2)

INTERNAL MEMORY (cont'd)

Wave Segment: A group of consecutive elements; wave segments are stored in memory. The wave segment must be a multiple of eight (8).

Scan: A scan is one pass through a wave segment.

Packets: Packets are one or more integer-multiple scans through a wave segment. They can advance either under sequencer control, HP-IB command or an external hardware trigger.

Sequence: A sequence is a series of packets in a user-defined order.

Maximum Number of Packets: 2048

Minimum Wave Segment Length: 56 elements

Maximum Scans/Packet: 65 536 under automatic sequencer advance. Unlimited under HP-IB or external hardware trigger advance.

Minimum Packet Dwell Time: 344 words (2.74 μs with 125 MHz internal clock).

MARKER OUTPUTS

Sequence Start: TTL trigger 370 ±25 ns prior to the beginning of the first packet in a sequence.

Packet Start: TTL trigger 370 ±25 ns prior to the start of each new packet when a packet is first entered.

Scan Start: TTL trigger 370 ± 25 ns prior to the beginning of every scan of a packet.

Address Equal: TTL trigger 300 ±25 ns prior to when a particular memory location is accessed whose address is set over HP-IB.

AUXILIARY OUTPUTS

10 MHz Reference Output 1 and 2: Sine wave output of 10 MHz crystal reference at 0 dBm (nominal) into 50Ω.

Memory Clock: TTL output of memory clock running at (sample rate)/8.

Packet Advance Ready: TTL high trigger when Sequencer is ready to advance to next packet.

AUXILIARY INPUTS

External Clock: Sine wave input from 50 to 130 MHz at 0.1 to 0.7 Vrms (0 dBm to +10 dBm) into 50Ω. The external clock is divisible by 2, 4, 8, 16, 32, 64, 128 or 256. When using the external clock instrument accuracy, stability and spectral characteristics will be determined by the external clock.

Packet Advance: Needs TTL high trigger to advance to the next packet of a sequence. Packet's advance must be set by user for external hardware trigger. Advance will be delayed until present memory scan is completed.

External Data Port: Parallel data port for high-speed data transfer to the internal waveform memory or sequencer memory. The External Data Port has sixteen data lines and two handshake lines. For transfer with the HP 9000 Series 200 Model 236, it requires an HP 11738A cable and HP 98622A GPIO interface card and optionally an HP 98620B DMA controller card. Binary data transfer time with HP 9000 Series 200 Model 236 and External Data Port for entire 128k word memory is:

HP 9000 Series 200 Model 236 + HP 98622A: 4 seconds typically;
HP 9000 Series 200 Model 236 + HP 98622A + HP 98620B: 1 second typically.

HP-IB INTERFACE

Binary Data Transfer Rate: Typically 20 seconds for entire 128k memory with HP 9000 Series 200 Model 236.

Bus Address: Set by slide switches on rear panel of instrument.

Table 1-3. Recommended Test Equipment (1 of 2)

Instrument	Critical Specifications	Recommended Model	Use[1]
Amplifier	Gain: >20 dB Output Power: >+6 dBm for 1 dB gain compression Bandwidth: 0.1 to 100 MHz Noise Figure: <5 dB	HP 8447A (Two amplifiers required)	P
Controller and BASIC Operating Language	HP-IB Compatibility as defined by IEEE Std 488 and the identical ANSI Std MC1.1: SH1, AH1, T2, TE0, L2, LE0, SR0, PP0, DC0, DT0 and C1, 2, 3, 4, 5 Run HP BASIC Memory: >=16k RAM GPIO Compatible[4]	HP 9000 Series 200 Model 226 or 236/ 98613A/11738A/ 98622A[2] or HP 85B[3]	A, O, P, T
Digital Multimeter	Modes: AC, DC and ohms Range: −40 Vdc to +20 Vdc Resolution: 10 mVdc Accuracy: 0.1% of reading	HP 3478A, HP 3455A, or HP 3466A	A, T
Extender Boards	Half Extender Full Extender	HP 08770-60113 HP 08770-60114	A, T
50Ω Termination	50Ω at 10 MHz	HP 11593A	A, P
Frequency Counter	Short Term Time Base Accuracy: $\pm 1 \times 10^{-8}$ Resolution: ± 10 Hz Frequency Range: 10 kHz to 125 MHz Sensitivity: −10 dBm	HP 5335A, HP 5342A, or HP 5343A	A, O, P, T
Frequency Standard	Short Term Stability: $\leq 5 \times 10^{-12}$/second Long Term Stability: $<5 \times 10^{-10}$/day	HP 5065A or HP 5061A (Option 004)	A, P
High Speed Data Extender Cable	No Substitute	HP 08770-60201	T
Oscilloscope	Bandwidth: 100 MHz Channels: Two	HP 1980B or HP 1715A	A, O, P, T
Power Meter	Frequency Range: 100 kHz to 125 MHz Power Range: 1 μW to 100 mW Accuracy: ±0.1 dB Resolution: ± 0.1 dB	HP 436A	A, O, P, T
Power Sensor	Sensitivity: 1 μW to 100 mW Impedance: 50Ω Connector: Type-N	HP 8482A	A, O, P, T
Sampler Delay Adjustment Tool	No Substitute	HP 8710-1659	A, T

Table 1-3. Recommended Test Equipment (2 of 2)

Instrument	Critical Specifications	Recommended Model	Use[1]
Signature Analyzer	Compatibility: TTL Clock: >10 MHz	HP 5006A or HP 5005A	T
75Ω to 50Ω Adapter[5]	Conversion Loss: 1.76 dB ±0.1 dB Connectors: Type-N or BNC Frequency Range: dc to 50 MHz	HP 11687A	A, O, P, T
Spectrum Analyzer	Frequency Range: 100 kHz to 125 MHz Dynamic Range: ≥70 dB Bandwidth: Variable Minimum Bandwidth: <1 kHz Maximum Input Level: +20 dBm Vertical Amplitude Resolution: 1 dB or 2 dB and 10 dB/division	HP 8557A/853B, HP 8566B, HP 8569B or HP 8565A	A, O, P, T
Step Attenuator	Range: 0 to 110 dB Step Size: 10 dB Accuracy: ± 0.4 dB or ± 1.7% of setting (whichever is greater) Frequency Range: dc to 1 GHz Impedance: 50Ω	HP 8496A	P

(1) A = Adjustments; P = Performance Tests; T = Troubleshooting; O = Operation Verification
(2) The controller must be GPIO compatible for the External Data Input Check only.
(3) HP 11738A (External Data Port Cable) and HP 98622A (GPIO Interface) are needed for the External Data Input Check only.
(4) This controller can be used for all tests except when the test requires a GPIO compatible controller.
(5) This adapter is used with an Option 002 (75Ω RF OUTPUT) instrument.

SECTION II
INSTALLATION

2-1. INTRODUCTION

This section provides the information needed to install the Synthesizer. Included is information pertinent to initial inspection, power requirements, line voltage selection, power cables, interconnection, environment, instrument mounting, storage and shipment. Also included in this section is an explanation of the HP-IB address switch.

2-2. INITIAL INSPECTION

WARNING

To avoid hazardous electrical shock, do not perform electrical tests when there are signs of shipping damage to any portion of the outer enclosure (covers or panels).

Inspect the shipping container for damage. If the shipping container or cushioning material is damaged, it should be kept until the contents of the shipment have been checked for completeness and the instrument has been checked mechanically and electrically. The contents of the shipment should be as shown in Figure 1-1; however, all of the options shown may not have been ordered. Procedures for checking electrical performance are given in Section IV. If the contents are incomplete, if there is mechanical damage or defect, or if the instrument does not pass the electrical performance tests, notify the nearest Hewlett-Packard office. If the shipping container is damaged, or the cushioning material shows signs of unusual stress, notify the carrier as well as the Hewlett-Packard office. Keep the shipping materials for the carrier's inspection.

2-3. PREPARATION FOR USE

2-4. Power Requirements

WARNINGS

To avoid the possibility of hazardous electrical shock, do not operate this instrument at line voltages greater than 126.5 Vac with line frequencies greater than 66 Hz. Leakage currents at these line settings may exceed 3.5 mA.

This is a Safety Class I product (that is, it is provided with a protective earth terminal). An uninterruptible safety earth ground must be provided from the main power source to the product input wiring terminals through the power cable or supplied power cable set. Whenever it is likely that the protection has been impaired, the product must be made inoperative and be secured against any unintended operation.

If this instrument is to be energized via an external autotransformer, make sure the autotransformer's common terminal is connected to neutral (that is, the grounded side of the line (Mains) supply.)

The Synthesizer requires a power source of 115 Vac (+10%,−25%) or 230 Vac (+10%,−15%) single phase. Line frequency is 48—66 Hz. Power consumption is 445 VA maximum.

2-5. Line Voltage and Fuse Selection

CAUTION

BEFORE PLUGGING THIS INSTRUMENT into the line (Mains) voltage, be sure the correct voltage and fuse have been selected.

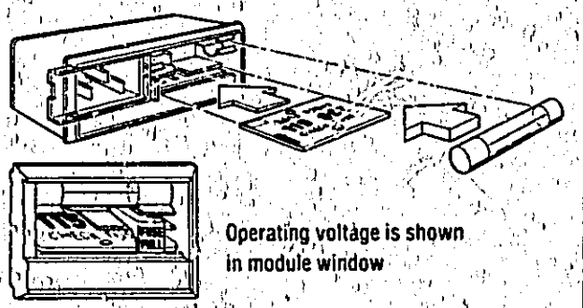
Verify that the line voltage selection card and the fuse are matched to the power source. See Figure 2-1, Line Voltage and Fuse Selection.

Fuses may be ordered using HP part numbers 2110-0750, 6A (250V, time delay) for 115 Vac operation and 2110-0751, 3A (250V, time delay) for 230 Vac operation.

2-6. Power Cables

WARNING

BEFORE CONNECTING THIS INSTRUMENT, the protective earth terminals of this instrument must be con-



Operating voltage is shown in module window

SELECTION OF OPERATING VOLTAGE

1. Slide cover and rotate FUSE PULL to left.
2. Select line (Mains) voltage by orienting PC board with desired voltage on top-left side. Push board firmly into module slot.
3. Rotate FUSE PULL back into normal position and re-insert fuse in holders, using caution to select correct fuse value.

WARNING

To avoid the possibility of hazardous electrical shock, do not operate this instrument at line voltages greater than 126.5 Vac with line frequencies greater than 66 Hz (leakage currents at these line settings may exceed 3.5 mA).

Figure 2-1. Line Voltage and Fuse Selection

Power Cables (cont'd)

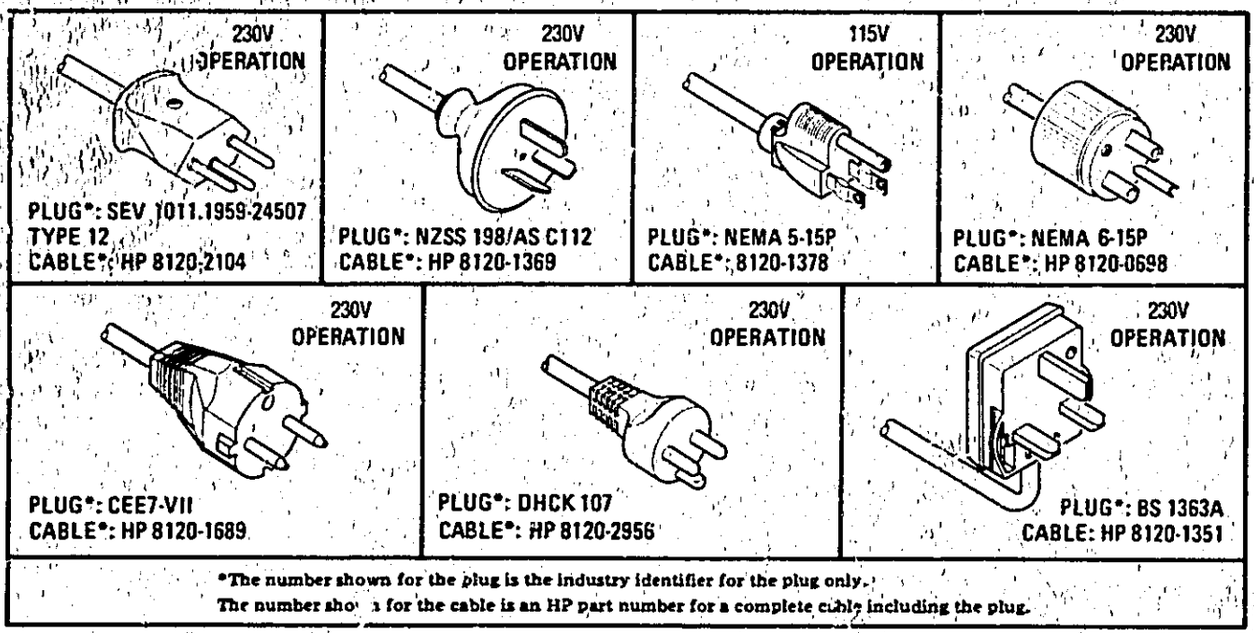
WARNING (cont'd)

nected to the protective conductor of the line (Mains) power cable. The line plug shall only be inserted in a socket outlet provided with a protective earth contact. The protective action must not be negated by the use of an extension cord (power cable) without a protective conductor (grounding). Grounding one conductor of a two conductor outlet is not sufficient protection.

This instrument is equipped with a three-wire power cable. When connected to an appropriate ac power receptacle, this cable grounds the instrument cabinet. The type of power cable plug shipped with each instrument depends on the country of destination. See Figure 2-2, Power Cable and Line (Mains) Plug Part Numbers, for the part numbers of power cables actually shipped. Cables are available in different lengths and some with right angle plugs to the instrument. Check with your nearest HP service center for descriptions and part numbers for these cables.

2-7. 10 MHz Reference Jumper Cable

Connect the gray BNC—BNC cable supplied in the accessory bag from the 10 MHz REFERENCE



*The number shown for the plug is the industry identifier for the plug only. The number shown for the cable is an HP part number for a complete cable including the plug.

Figure 2-2. Power Cable and Line (Mains) Plug Part Numbers

10 MHz Reference Jumper Cable (cont'd)

OUTPUT 2 to the 10 MHz REFERENCE INPUT. This phase locks the internal 125 MHz clock to the internal 10 MHz reference. If an external clock is being used, the jumper cable should not be connected.

2-8. HP-IB Address Selection

NOTE

The HP-IB address can be changed only at instrument power-up. If the Synthesizer is left on, the address will not change simply by changing the address switches.

The Synthesizer's HP-IB address is set by a seven-segment switch (S1) located on the rear panel. The switch is illustrated in Figure 2-3, HP-IB Address Switch Shown as Set by the Factory. The segments labeled 1, 2, 4, 8, and 16, when set to the 1 position, are added together to determine the HP-IB address. The address is factory set to 19 but can be reset by the user. The preset address is shown shaded in Table 2-1. Valid addresses range from 0 to 30. The NORMAL/LISTEN ONLY switch segment sets the Synthesizer to listen only mode when set to 0 or to normal talker/listener mode when set to 1. The unlabeled seventh segment should always be set to 0.

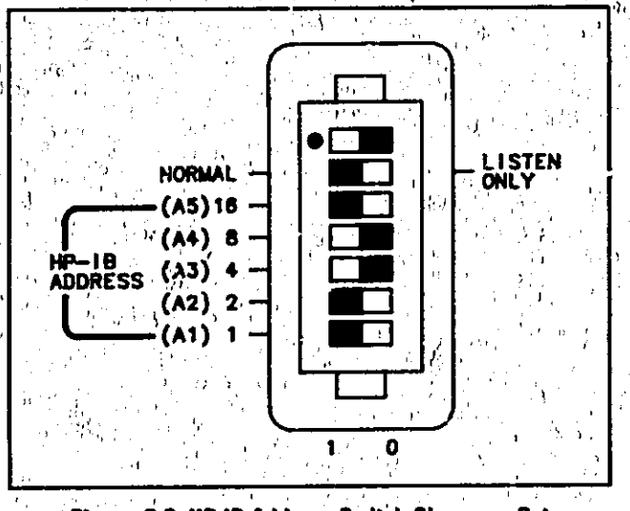


Figure 2-3. HP-IB Address Switch Shown as Set by the Factory

2-9. Mating Connectors

HP-IB Interface Connector. Interconnection data for the Hewlett-Packard Interface Bus is provided in Figure 2-4.

Table 2-1. Allowable HP-IB Address Codes

Decimal Equivalent[1]	Listen Address Character	Talk Address Character	Address Switches[1]				
			A5	A4	A3	A2	A1
0	SP	@	0	0	0	0	0
1	!	A	0	0	0	0	1
2	"	B	0	0	0	1	0
3	#	C	0	0	0	1	1
4	$	D	0	0	1	0	0
5	%	E	0	0	1	0	1
6	&	F	0	0	1	1	0
7	'	G	0	0	1	1	1
8	(	H	0	1	0	0	0
9	)	I	0	1	0	0	1
10	*	J	0	1	0	1	0
11	+	K	0	1	0	1	1
12	,	L	0	1	1	0	0
13	-	M	0	1	1	0	1
14	.	N	0	1	1	1	0
15	/	O	0	1	1	1	1
16	0	P	1	0	0	0	0
17	1	Q	1	0	0	0	1
18	2	R	1	0	0	1	0
19[2]	3	S	1	0	0	1	1
20	4	T	1	0	1	0	0
21	5	U	1	0	1	0	1
22	6	V	1	0	1	1	0
23	7	W	1	0	1	1	1
24	8	X	1	1	0	0	0
25	9	Y	1	1	0	0	1
26	:	Z	1	1	0	1	0
27	;	[	1	1	0	1	1
28	<	\	1	1	1	0	0
29	=	]	1	1	1	0	1
30	>	^	1	1	1	1	0

[1]Decimal characters and the five address switches relate to the last five bits of both talk and listen addresses.

[2]Factory-set address.

External Data Input Connector. Interconnection data for the External Data Input is provided in Figure 2-5.

Coaxial Connectors. Except for the RF OUTPUT connector, all coaxial mating connectors used with the Synthesizer should be BNC male. The RF OUTPUT connector mates with a 50Ω Type N male. (For Option 002, the mating is a 75Ω

Mating Connectors (cont'd)
BNC male.) All mating connectors should be compatible with US MIL-C-39012.

2-10. Operating Environment

The operating environment should be within the following limitations:

Temperature ..... 0°C to +55°C
Humidity ..... 40°C at 5% to 95% relative
Altitude ..... <4600 meters (15 000 feet)

Specifications are warranted for a +15 to +40°C operating environment.

2-11. Bench Operation

The instrument cabinet has plastic feet. The plastic feet are designed to ensure self-aligning of the instruments when stacked.

2-12. Rack Mounting

Rack mounting information is provided with the rack mounting kits. If the kits were not ordered with the instrument as options, they may be ordered through the nearest Hewlett-Packard office. Refer to the paragraph titled "Mechanical Options" in Section I.

2-13. RF OUTPUT CONNECTOR RETROFIT PROCEDURES

NOTE

The following retrofit procedures should only be performed by qualified service-trained personnel.

2-14. Standard to Option 002 Retrofit

Follow the procedure below to convert the 50Ω RF output to a 75Ω RF output.

1. Disconnect the power cable from the Synthesizer.
2. Unlock and open the rear door.
3. Disconnect W4, the black RF output cable, at the attenuator end.
4. Using an open-ended wrench, remove two hex-nuts one at a time from the RF output connector on the rear panel door. Slide hex-nuts, washer and rear panel spacer insert down cable and over connector.
5. Remove N-type connector and cable assembly from rear panel door.
6. Remove hex-nut and the first (and smallest) of four washers from 75Ω adapter, HP part number 11658-60001. Leave other three washers on threaded end of adapter.
7. Push threaded end of adapter through rear panel from the inside out. Threaded end should now protrude through outside of rear door D-hole.
8. Install small washer and hex-nut on outside to secure adapter. There should be three large washers installed on inside and one small washer on outside of rear panel door.
9. Connect BNC end of accessory black cable, HP part number 08770-60028, to rear panel adapter. Twist to lock in place.
10. Screw other end of cable onto attenuator output connector.
11. Close and lock rear door.
12. Perform Sine Wave Output Power and Attenuator Accuracy Performance Test in Section IV of manual.

2-15. Option 002 to Standard Retrofit

Follow the procedure below to convert the 75Ω RF output to a 50Ω RF output.

1. Disconnect the power cable from the Synthesizer.
2. Unlock and open the rear door.
3. With an open-ended wrench, disconnect hex-nut from RF output connector on outside of door. Slide connector/cable assembly out of D-hole in rear panel door.
4. Disconnect black RF output cable, W4, at other end from attenuator output connector.
5. Remove connector and cable assembly from instrument.
6. Install 50Ω N-type cable assembly (in accessory bag) by sliding small cable end through outside of rear panel D-hole. Continue inserting cable until threaded end of N-type connector protrudes through inside of rear door.
7. In the following order, slide the spacer, washer and two retainer nuts onto the other end of cable. Small end of spacer should be inserted into hole in rear panel door.

Option 002 to Standard Retrofit (cont'd)

8. Tighten inside nut followed by second retainer nut.
9. Connect other end of cable to attenuator output connector.
10. Close and lock rear door.
11. Perform Sine Wave Output Power and Attenuator Accuracy Performance Test in Section IV of manual.

2-16. STORAGE AND SHIPMENT

2-17. Environment

The instrument should be stored in a clean, dry environment. The following environmental limitations apply to both storage and shipment:

Temperature ..... −55°C to +75°C
Humidity ..... 40°C at <95% relative
Altitude ..... <15 300 meters (50 000 feet)

2-18. Packaging

Tagging for Service. If the instrument is being returned to Hewlett-Packard for service, please complete one of the blue repair tags located at the end of this manual and attach it to the instrument.

Original Packaging. Containers and materials identical to those used in factory packaging are available through Hewlett-Packard offices. Mark the container "FRAGILE" to assure careful handling. In any correspondence refer to the instrument by model number and full serial number.

Other Packaging. The following general instructions should be used for repackaging with commercially available materials.

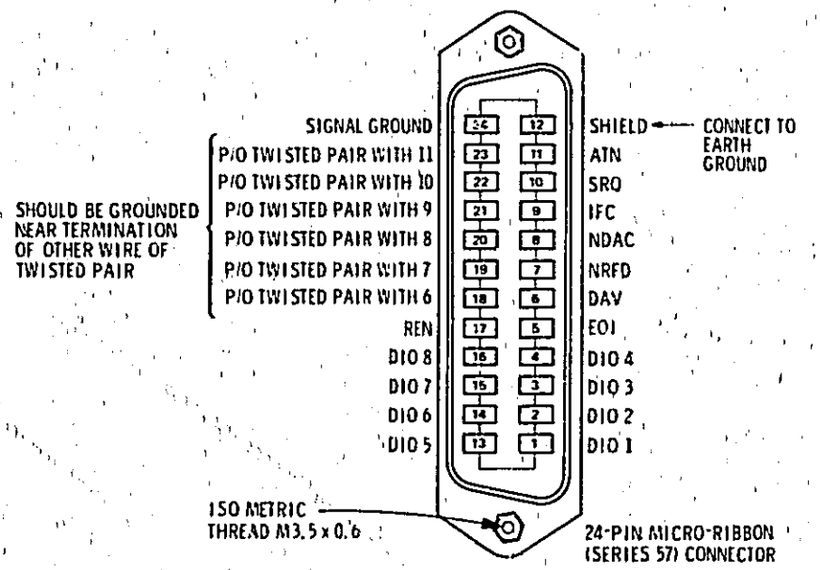
a. Wrap the instrument in heavy paper or plastic. If shipping to a Hewlett-Packard office or service center, complete one of the blue tags mentioned above and attach it to the instrument.

b. Use a strong shipping container. A double-wall carton made of 2.4 MPa (350 psi) test material is adequate.

c. Use enough shock-absorbing material (75 to 100 mm layer-3 to 4 inches) around all sides of the instrument to provide a firm cushion and prevent movement in the container. Protect the front panel with an appropriate type of cushioning material to prevent damage during shipment.

d. Seal the shipping container securely.

e. Mark the shipping container "FRAGILE" to encourage careful handling.



Logic Levels

The Hewlett-Packard Interface Bus Logic Levels are TTL compatible, i.e., the true (1) state is 0.0 Vdc to +0.4 Vdc and the false (0) state is +2.5 Vdc to +5.0 Vdc.

Programming and Output Data Format

Refer to Section III, Operation.

Mating Connector

HP 1251-0293; Amphenol 57-30240.

Mating Cables Available

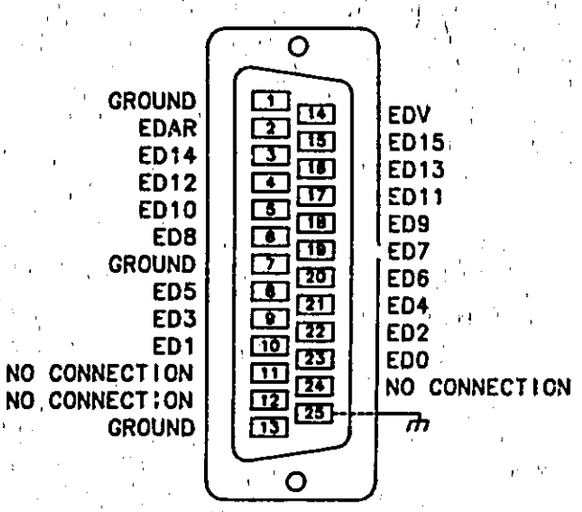
HP 10833A, 1 metre (3.3 ft), HP 10833B, 2 metres (6.6 ft)
HP 10833C 4 metres (13.2 ft), HP 10833D, 0.5 metres (1.6 ft)

Cabling Restrictions

1. A Hewlett-Packard Interface Bus system may contain no more than 2 metres (6.6 ft) of connecting cable per instrument.

2. The maximum accumulative length of connecting cable for any Hewlett-Packard Interface Bus system is 20.0 metres (65.6 ft).

Figure 2-4. Hewlett-Packard Interface Bus Connection



Logic Levels

The External Data Input has logic levels that are TTL compatible. For example, lines ED0—ED15, the false (0) state is 0.0 Vdc to +0.8 Vdc and the true (1) state is 2.5 Vdc to +5.0 Vdc. Lines EDV and EDAR power up the same as lines ED0—ED15. The active level for EDV and EDAR can be programmed; refer to Section III, Operation.

Mating Connector

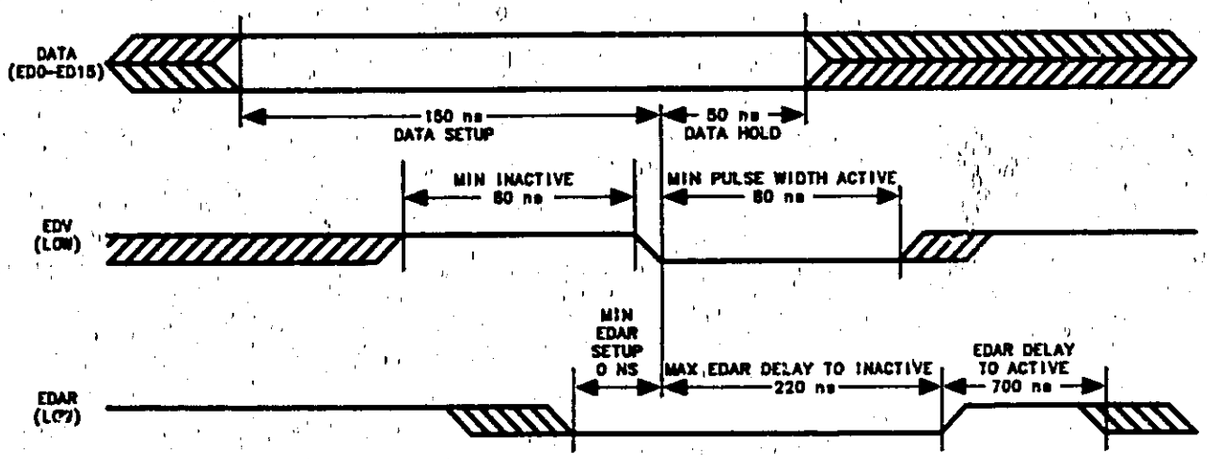
HP part number 1251-0063; TRW DBM-25P.

Mating Cables Available

HP 11738A, 2 metres (6.5 ft.)

External Data Input Restrictions

When using the External Data Input, there are critical timing considerations that must be met as shown in the following timing diagram:



External Data Transfer Timing

NOTE

If there seems to be a timing problem when using the following Hewlett-Packard equipment:

- *HP 9000 Series 200 Model 236 Computer*
- *HP 98620B DMA Controller Card*
- *HP 98622A GPIO Interface Card*
- *HP 11738A External Data Input Cable*

refer to the PCTL Delay Adjustment in the HP 98622A GPIO Interface Installation manual. Typically, installing a 200 pf capacitor (C6) on the GPIO Interface Card will solve the timing problem.

Figure 2-5. External Data Input Connector

SECTION III
OPERATION

3-1. INTRODUCTION

3-2. General

This section provides complete operating information for the Arbitrary Waveform Synthesizer.

Operating Characteristics. Table 3-1 briefly summarizes the major operating characteristics of the Synthesizer. This table is not intended to be a complete listing of all operations and parameters, but gives a general idea of the instrument's capabilities. For more information on the Synthesizer's capabilities, refer to Table 1-1, Specifications, and Table 1-2, Supplemental Characteristics. For information on HP-IB capabilities, refer to Table 3-4, Message Reference Table.

Panel Features. Front and rear panel features are described in detail in Figures 3-1 through 3-4.

General Operating Instructions. Instructions relating to the Synthesizer turn-on procedure, power-up checks and general operating procedures are presented to acquaint the user with the general operation of the instrument.

HP-IB Remote Operation. The Synthesizer is capable of remote operation via the Hewlett-Packard Interface Bus (HP-IB). This section includes discussions on capabilities, addressing, input and output formats, the status byte and service requests.

Programming the Status Register. The Status Register is an 8-bit register that monitors several Synthesizer conditions. This section explains how to program the Synthesizer to report back to the controller whenever a predetermined condition occurs.

Programming Commands. All Synthesizer programming commands are described at the end of this section. The commands are listed in alphabetical order.

NOTE

All examples in this section are written in an enhanced BASIC language using the HP 9000 Series 200 Model 236. The examples assume the select code of the HP-IB interface is 7 and the Synthesizer's address is 19.

3-3. Operator's Maintenance

WARNING

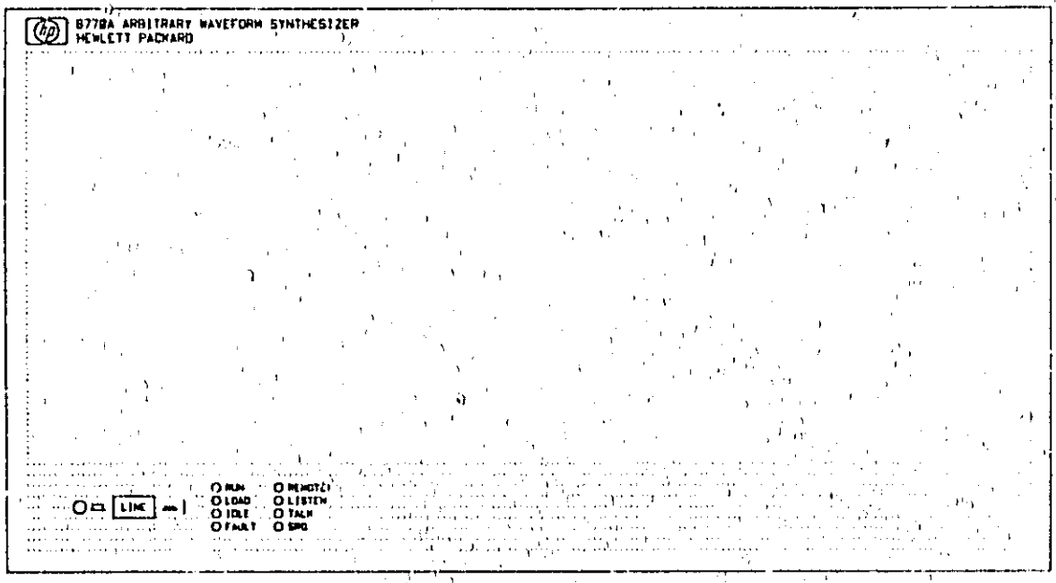
For continued protection against fire hazard, replace the line fuse with a 250V fuse of the same rating only. Do not use repaired fuses or short-circuited fuseholders.

Operator's maintenance consists of replacing defective fuses. The primary power fuse is located within the line module assembly. Refer to Figure 2-1 for instructions on how to change the fuse.

If the instrument does not operate properly and is being returned to Hewlett-Packard for service, please complete one of the blue tags located at the end of this manual and attach it to the instrument. Refer to Section II for packaging instructions.

Table 3-1. Operating Characteristics

Parameter	Capability
Internal Waveform Memory	Size: 131 072 12-bit words Maximum Number of Files: 1400
Sampling Clock	Internal: 125 MHz External: 60 to 130 MHz Divisible by 1, 2, 4, 8, 16, 32, 64, 128 and 256.
Wave Segment	Length: 56 to 131 072 elements in multiples of 8.
Packet	Minimum Length (number of elements in wave segment x scans): 344
Sequence	Maximum Number of Packets: 2048
RF Output	Frequency Range: dc to 50 MHz Attenuation: 0 to 110 dB in 10 dB steps Peak Output: 2V peak-to-peak (+10 dBm) with 0 dB of attenuation



LINE SWITCH AND FRONT PANEL ANNUNCIATORS

LINE Switch. Applies power to the Synthesizer when set to the ON position.

RUN. Lights when the sequencer is running.

LOAD. Lights when waveform memory is being loaded.

IDLE. Lights when power is being applied to the Synthesizer but the sequencer is not running and waveform memory is not being loaded.

FAULT. Lights when a hardware error occurs. Refer the instrument to qualified maintenance personnel or contact your nearest Hewlett-Packard office for service.

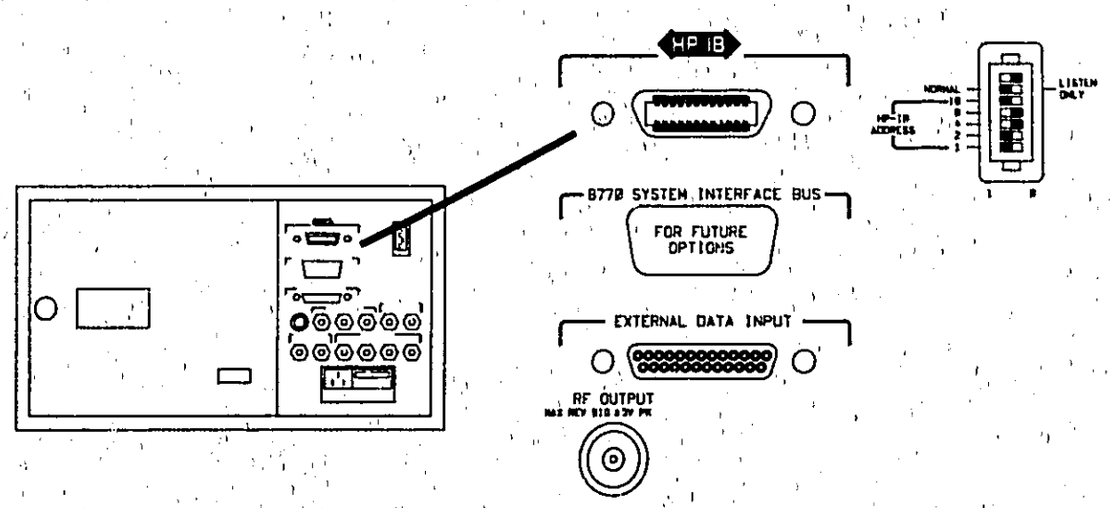
REMOTE. Lights when the Synthesizer is in remote mode.

LISTEN. Lights when the Synthesizer is addressed to listen.

TALK. Lights when the Synthesizer is addressed to talk.

SRQ. Lights when the Synthesizer is sending a Require Service message to the controller.

Figure 3-1. Front Panel Features



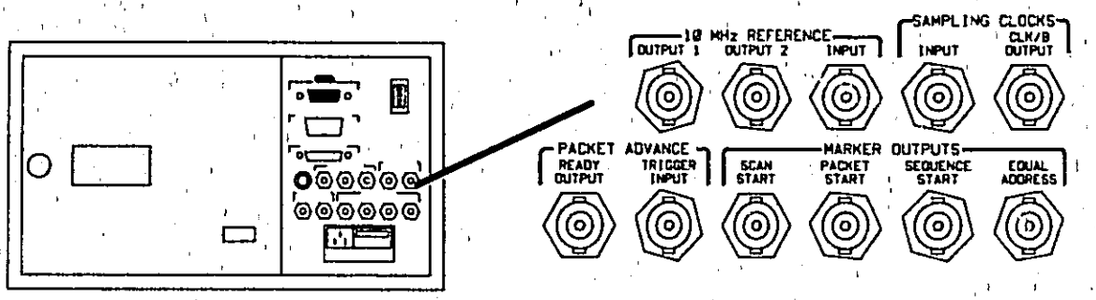
HP-IB Connector. Connects Synthesizer to Hewlett-Packard Interface Bus for remote operation. See Figure 2-4 for additional information.

HP-IB Address Switch. Switch setting determines Synthesizer address to be used for communications via HP-IB. Factory set to 19. See Section II for additional information.

EXTERNAL DATA INPUT. Used for high speed data transfer into waveform or sequencer memory. See Figure 2-5 for additional information.

RF OUTPUT. 50 ohm Type N female connector. Output is typically −1 to +1V into 50 ohms (75 ohm BNC Connector for option 002).

Figure 3-2. HP-IB, External Data Input and RF Output Rear Panel Features



10 MHz REFERENCE

OUTPUT 1 and OUTPUT 2. Provide typically 0 dBm, 10 MHz reference signals derived from the internal crystal reference.

INPUT. Connects to a 10 MHz reference for locking the internal 125 MHz phase-locked loop. To use the internal 10 MHz reference, connect a jumper cable between this connector and either OUTPUT 1 or OUTPUT 2. To use an external standard, disconnect the jumper and connect the external source to this connector. The external reference should be 10 MHz ±150 Hz and provide 0 to +10 dBm into the 50 ohm BNC connector. Without the 10 MHz reference, the internal 125 MHz oscillator will free run. This results in slightly degraded frequency accuracy performance.

SAMPLING CLOCKS

INPUT. Connects to an external clock source. The external source must be between 60 and 130 MHz and provide 0 to +10 dBm into the 50 ohm connector. The HP-IB CLKSEL command must be set to EXT mode.

CLK/8 OUTPUT. Provides a 50% duty cycle, TTL level clock signal at a frequency of 1/8 the sampling rate.

PACKET ADVANCE

READY OUTPUT. TTL compatible output. Provides a TTL high signal when the sequencer is ready to advance to the next packet via an external trigger.

TRIGGER INPUT. TTL compatible input. When READY OUTPUT signal is high, applying an external TTL high signal triggers the sequencer to advance to the next packet. This input is positive edge sensitive and must be high for a minimum of 25 ns.

MARKER OUTPUTS

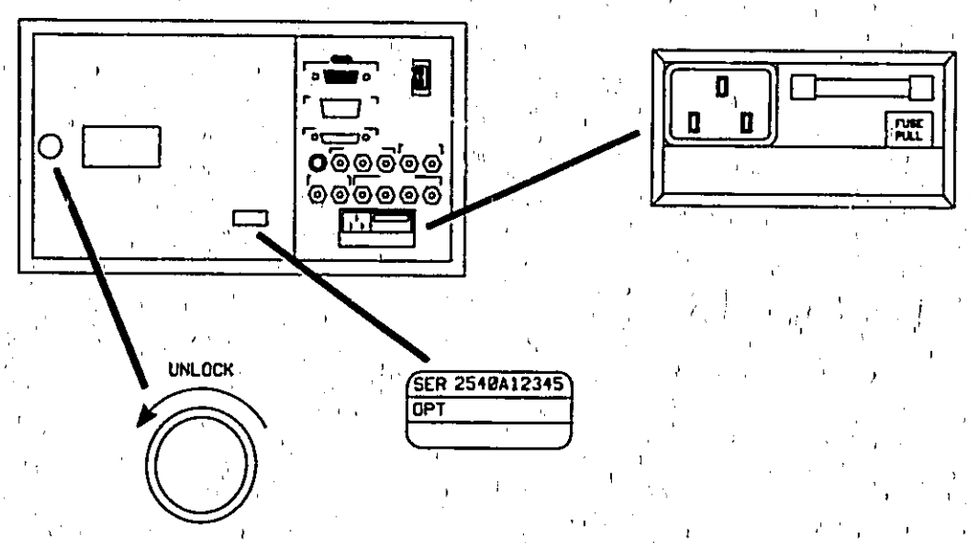
SCAN START. Provides a TTL high signal. The pulse width remains high for a minimum of 8 x (1/clock). For example, the output remains high for at least 64 ns when using a 125 MHz sampling clock. Start of pulse will occur 370 ±25 ns prior to the start of every new scan.

PACKET START. Provides a TTL high signal. The pulse width remains high for a minimum of 8 x (1/clock). Start of pulse will occur 370 ±25 ns prior to the start of every packet.

SEQUENCE START. Provides a TTL high signal. The pulse width remains high for a minimum of 8 x (1/clock). Start of pulse will occur 370 ±25 ns prior to the start of every sequence.

EQUAL ADDRESS. Provides a TTL high signal. The pulse width remains high for a minimum of 8 x (1/clock). Start of pulse will occur 300 ±25 ns prior to accessing a memory location whose address has been set with the MARKER or MARKADD command.

Figure 3-3. 10 MHz Reference, Sampling Clocks, Packet Advance and Marker Outputs Rear Panel Features



Line Module. Permits operation from 115 or 230 Vac. The number visible in the window indicates the nominal line voltage to which the instrument must be connected (see Figure 2-1). Center conductor is safety earth ground.

Fuse. 6A (250V, time delay) for 115 Vac. 3A (250V, time delay) for 230 Vac.

Knob. Enables access to inside of instrument for service.

Serial Number Plate. First four numbers and letter form the prefix that denotes instrument configuration. The last five digits form the suffix that is unique to each instrument.

Figure 3-4. Miscellaneous Rear Panel Features

3-4. GENERAL OPERATING INSTRUCTIONS

3-5. Turn-On Procedure

WARNING

Before the Synthesizer is switched on, all protective earth terminals, extension cords, autotransformers, and devices connected to the instrument should be connected to a common protective earth grounded socket. Any interruption of the protective earth grounding will cause a potential shock hazard that could result in personal injury.

CAUTION

Before the Synthesizer is switched on, it must be set to the voltage of the power source or damage to the instrument may result.

The Synthesizer receives commands and data over the HP-IB. Therefore, an HP-IB (IEEE 488) compatible controller must be used with the Synthesizer to form a system.

If the internal 10 MHz reference is to be used to phase lock the internal 125 MHz clock, be sure that the jumper, W28, connects the 10 MHz REFERENCE OUTPUT to the 10 MHz INPUT.

If the Synthesizer is being used with the HP 11775A Waveform Generation Software, it must be the first instrument in the system to be turned on.

If the Synthesizer is already plugged in, press the LINE switch to ON.

If the Synthesizer is not plugged in, follow these instructions:

1. On the rear panel, check the line voltage card for correct voltage selection.
2. Check that the fuse rating is appropriate for the line voltage used (see Figure 2-1). Fuse ratings are printed on the rear panel.
3. Plug in the power cable.

On the front panel, press the line switch to ON.

3-6. Power-Up Checks

The Synthesizer performs a quick internal check at turn-on. During this check all front panel annunciators light for approximately two seconds to allow for a quick visual inspection. At the conclusion of the check, each annunciator is turned on, one at a time, in sequence. At the end of this sequence the IDLE annunciator remains on. All other annunciators should be off.

If the FAULT, REMOTE, TALK, LISTEN and SRQ annunciators all remain lit at the end of the power-up check, the HP-IB address switch is set to an invalid address. Turn off the instrument and change the address switch setting. Refer to "HP-IB Address Selection" in Section II for instructions.

If a hardware failure was detected, the FAULT annunciator will remain on. Refer the instrument to qualified maintenance personnel or contact your nearest Hewlett-Packard office for service.

The Synthesizer always powers up to the Reset state. The complete list of settings is described under the *RST Programming Command located near the end of this section. The Synthesizer does not retain instrument settings or save anything in its memory when power is removed. A procedure for backing up the entire Synthesizer memory is shown in Example 3-3 in this section.

3-7. Overview

The input to the Synthesizer is data from a computer. The data represents successive amplitude samples of a desired waveform. A high speed digital-to-analog converter (DAC) inside the Synthesizer converts the digital data to an analog waveform. The Synthesizer can produce waveforms with frequency components up to 50 MHz.

The Synthesizer contains three main assemblies to reproduce waveforms: a buffer memory to hold wave segments, a sequencer to control how the wave segments are combined to produce a desired waveform, and a DAC to convert the wave segment digital data to an analog voltage.

The minimum time from sample to sample is 8 ns. This is equivalent to a sampling rate of 125 MHz. (8 ns=1/125 MHz.) Due to the very fast sampling rate, eight channels of multiplexed, buffered memory are required. An external computer pre-loads the Synthesizer's buffer memory (also known as waveform memory) for use by the sequencer and DAC.

Overview (cont'd)
The Synthesizer's memory is partitioned into eight channels of 16k words x 12 bits each. A multiplexer strobes the channels so the first element comes to the DAC from channel 1, the second from channel 2 and so on. Each channel of memory runs at one eighth of the sampling clock.

There are 128k waveform elements in Synthesizer memory (16k words x 8 channels). When reproducing low frequency waveforms, sampling every 8 ns and storing an element wastes Synthesizer memory. Therefore, the sampling clock rate can be divided by powers of 2 from 2 to 256.

To further conserve memory the Synthesizer uses a sequencer, which allows wave segments in memory to be repeated many times and in any order before moving on to the next wave segment.

An amplifier at the output of the DAC gives the user a ±1V amplitude range at the RF OUTPUT connector (2V peak-to-peak). An output attenuator can lower the output level in 10 dB steps from 0 to −110 dB. The output power can range from +10 to −110 dBm.

See Figure 3-5 for a simplified block diagram of the Synthesizer.

3-8. Definition of Terms

Elements. An element is a twelve-bit value representing the amplitude of a sample point.

Wave Segment. A wave segment is a group of consecutive elements.

Scan. A scan is a pass through a wave segment.

Packet. A packet is one or more scans through a wave segment.

Sequence. A sequence is a combination of one or more packets.

Waveform Memory. This is the Synthesizer's 12 bit buffer memory where wave segments are stored. Data in waveform memory is sent to the DAC.

Sequencer Memory. This is the memory that stores information necessary to execute packets. Sequencer memory determines what packet data in waveform memory is sent to the DAC and the order in which data is sent.

Figure 3-6 shows the relationship between scans, packets and sequences.

3-9. CREATING WAVEFORMS

The Synthesizer uses digital waveform data from a computer to generate waveforms. The following steps must be performed in order for the Synthesizer to generate waveforms:

1. Create the desired waveform data on an external computer. The easiest way to generate waveform data is to use the HP 11775A

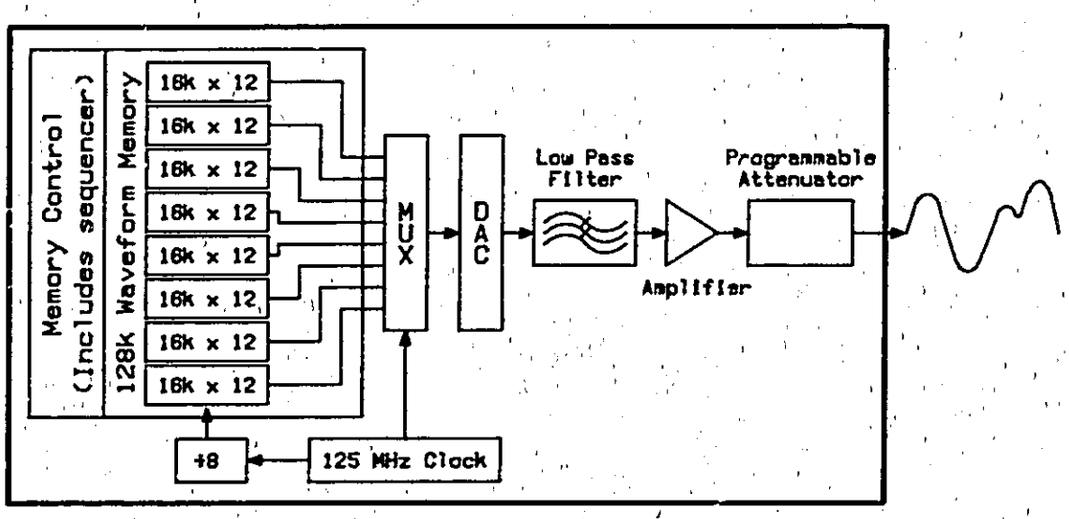


Figure 3-5. Simplified Synthesizer Block Diagram

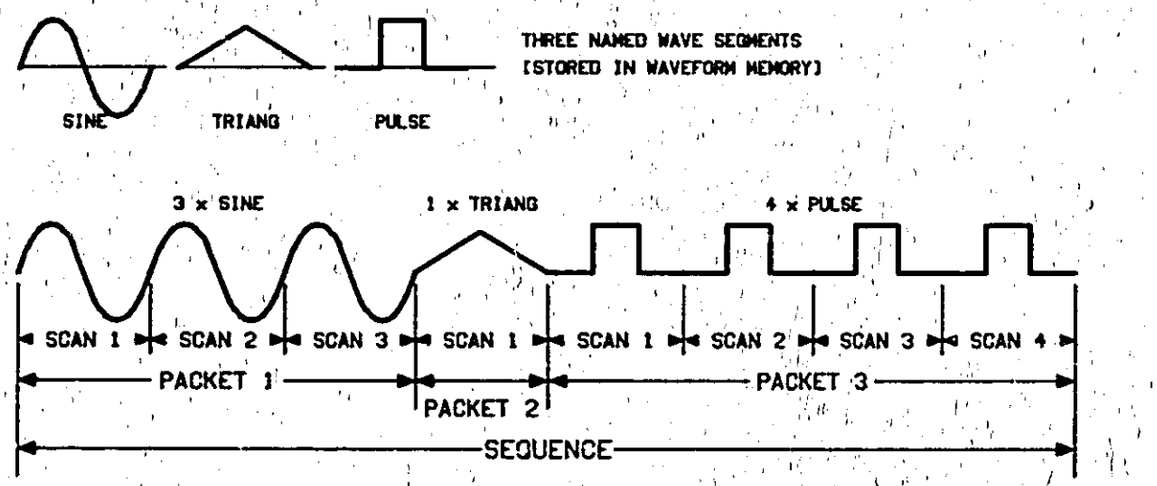


Figure 3-6. Scan, Packet and Sequence

Creating Waveforms (cont'd)

Waveform Generation Software. However, data can also be generated by describing waveforms mathematically in a program and by using HP-IB program commands.

2. Download data into waveform memory.
3. Download packet sequencing instructions into sequencer memory.
4. Turn RF output on, if it is not already on.
5. Start sequencer.

Example 3-1 shows a simple program demonstrating these steps.

Example 3-1. Creating a Simple Waveform

```
10    ! CREATING A SIMPLE WAVEFORM
20    !
30    INTEGER A(1:1024),I          ! DIMENSION INTEGER ARRAYS
40    !
41    OUTPUT 719;"PURGE BOTH"      ! PURGE WAVE AND SEQ. MEMORY TO DELETE
42                                 ! ANYTHING THAT MAY CURRENTLY BE STORED
43                                 ! THERE.
45    !
50    FOR I=1 TO 1024              ! CREATE DATA ARRAY TO LOAD INTO WAVEFORM
60        A(I)=I                   ! MEMORY. SYNTHESIZER REQUIRES DATA TO BE
70    NEXT I                       ! MULTIPLES OF 8 BEFORE IT CAN BE USED IN
80                                 ! IN A PACKET.
90    !
100   OUTPUT 719;"WAVE RAMP,";A(*) ! LOAD DATA ARRAY IN WAVEFORM
110                                ! MEMORY FILE NAMED "RAMP"
120   OUTPUT 719;"PACKET RAMP,64,AUTO"
130                                ! LOAD SEQUENCER MEMORY WITH ONE PACKET.
140                                ! THE PACKET CONSISTS OF ONE WAVE SEGMENT
141                                ! STORED IN SYNTHESIZER'S WAVEFORM MEMORY
142                                ! UNDER THE FILE NAME "RAMP." WAVE SEGMENT
143                                ! RAMP WILL BE SCANNED (REPEATED) 64
150                                ! TIMES. AUTO IS THE PACKET ADVANCE MODE.
160                                ! BECAUSE ONLY ONE PACKET IS STORED IN
161                                ! SEQUENCER MEMORY, THE PACKET IS REPEATED
162                                ! CONTINUOUSLY.
163                                !
170   OUTPUT 719;"OUTPUT ON"       ! TURN RF OUTPUT ON
171                                !
180   OUTPUT 719;"GO"              ! START SEQUENCER
190   END
```

Creating Waveforms (cont'd)

Example 3-1 creates the waveform shown in Figure 3-7, as seen when the Synthesizer's RF OUTPUT connector is connected to an oscilloscope.

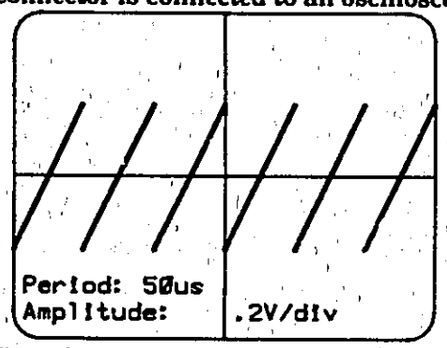


Figure 3-7. Waveform Created by Example 3-1

3-10. Loading Waveform Memory

Data must be loaded into waveform memory before data is loaded into sequencer memory. Data can be loaded into waveform memory using either named files or memory address locations. Example 3-1 uses program command WAVE to load data into a file named RAMP. To load the same data into address locations instead of a named file, use "WMEM 0" instead of "WAVE RAMP." 0 is the starting address of where the data will be placed. The 1024 elements in the data array will be stored in addresses 0 through 1023.

Working with named files is much easier than working with addresses. It can be difficult to keep track of the data when it is stored by address. Also, it is possible to write over data that is already in waveform memory.

Working With File Names. File names in waveform memory must have six or less characters. The first character must be a letter. The remaining characters can be A through Z, "_" (underscore) or 0 through 9. Lower case characters are treated as upper case. The following words are reserved and cannot be used for file names:

- ASCII
- BIN
- WAVE
- SEQ
- BOTH

Before a file can be used in a sequence (that is, used in PACKET command), it must have a length that is a multiple of eight. However, data files whose length is not a multiple of eight can be downloaded to waveform memory. Use the WAVEAP command to append data to that file to make it a multiple of eight before using it in a sequence.

Working With Address Locations. The maximum size of the waveform memory is 131 072 elements. Valid addresses range from 0 to 131 071. As with named files, data must be a multiple of eight before it can be used in a sequence.

Format. The FORMAT command sets the numbering system for data in waveform memory. The input to the Synthesizer is a 12 bit computer word. This means the Synthesizer can supply 4096 different amplitude levels. The FORMAT command determines how the Synthesizer interprets the data. FORMAT UNSIGN interprets the data from 0 to 4095; FORMAT SIGN interprets the data as −2048 to 2047.

Converting Data To Be Compatible With Synthesizer. All data to the Synthesizer's waveform memory must be within the legal bounds of 0 to 4095 for FORMAT UNSIGN and −2048 to 2047 for FORMAT SIGN. Example 3-2 shows how to convert waveform data stored as real numbers in an array into integer values between 0 and 4095 that are compatible with the Synthesizer.

(Example on next page)

Example 3-2. Converting Data to be Compatible with Synthesizer

```
10    ! CONVERT DATA TO BE COMPATIBLE WITH SYNTHESIZER
20    !
30    DIM In1(1:40)
40    INTEGER Out1(1:40)
50    FOR I=1 TO 40                                    ! GENERATE AND PRINT
60      In1(I)=2.0^I                                   ! WAVEFORM DATA
70      IF I MOD 2 THEN
80        In1(I)=-In1(I)
90      END IF
100     PRINT I,In1(I)
110   NEXT I
120                                                    !
130   Wave_to_aws(100000,In1(*),Out1(*))               ! CONVERT DATA TO AWS FORMAT
140                                                    !
150   FOR I=1 TO 40
160     PRINT I,Out1(I)                                ! PRINT DAC DATA
170   NEXT I
180                                                    !
190   END
200                                                    !
210                                                    !
220   SUB Wave_to_aws(REAL Scale,Inwave(*),INTEGER Outwave(*)
230     ! Scale   :  CONTAINS SCALING FACTOR THAT Inwave IS DIVIDED BY
240     !         :  Scale=0 AUTO SCALE FOR FULL RANGE OF THE DAC
250     ! Inwave  :  CONTAINS WAVEFORM DATA AS REAL NUMBERS
260     ! Outwave :  CONTAINS WAVEFORM DATA THAT IS COMPATIBLE WITH AWS
270     !
280     ! THIS PROGRAM CONVERTS WAVEFORM DATA STORED IN A REAL ARRAY INTO
290     ! INTEGER VALUES THAT ARE COMPATIBLE WITH AWS. THESE VALUES ARE
300     ! BETWEEN 0 AND 4095. SEE AWS 'FORMAT UNSIGN' HPIB COMMAND.
310     !
320     IF Scale=0 THEN                                ! AUTO SCALE WAVEFORM
330       Mindata=MIN(Inwave(*))                       ! LOWEST DATA VALUE
340       Maxdata=MAX(Inwave(*))                       ! HIGHEST DATA VALUE
350       Scale=MAX(ABS(Mindata),ABS(Maxdata))         ! MAXIMUM MAGNITUDE
360     END IF
370                                                    !
380     Asc=2047.5/Scale            ! AWS SCALE FACTOR FOR 12 BIT RESOLUTION
390                                                    !
400     FOR I=1 TO SIZE(Inwave,1)                      ! GENERATE DAC VALUES THAT
410       Inwave(I)=Inwave(I)*Asc+2047.5               ! ARE BETWEEN 0 & 4095
420       Inwave(I)=MIN(Inwave(I),4095)                ! LIMIT MAXIMUM DAC VALUE
430       Inwave(I)=MAX(Inwave(I),0)                   ! LIMIT MINIMUM DAC VALUE
440       Outwave(I)=INT(Inwave(I))                    ! INTEGERIZE DATA
450     NEXT I
460   SUBEND
```

3-11. Loading Sequencer Memory

Loading sequencer memory consists of creating packets and placing the packets in order (that is, creating a sequence). The number of packets in a sequence can range from 1 to 2048. In Example 3-1, the sequence contains only one packet.

It is a good practice to clear sequencer memory with the PURGE SEQ program command prior to entering a new sequence. Otherwise, new packets are appended to any packets that may have been defined since the last PURGE SEQ command was given.

Creating A Packet. A packet contains the information the sequencer needs to scan a wave segment. This information includes either the name, or starting address (in waveform memory) and length of the wave segment. It also includes the number of times to repeat the segment and the packet advance mode. Example 3-1 uses the program string:

Loading Sequencer Memory (cont'd)
OUTPUT 719; "PACKET RAMP, 64, AUTO" where wave segment RAMP is the file name, 64 is the number of times to repeat (scan) the wave segment and AUTO is the packet advance mode.

A wave segment must contain a minimum of 56 elements. A packet must contain a minimum of 344 elements (packet elements = wave segment elements x scans). Therefore, a packet consisting of a wave segment with 56 elements must be repeated at least seven times to achieve the minimum packet length.

Packets can be created using either the PACKET or PACLIT command. PACKET specifies wave segments by file name; PACLIT specifies wave segments by waveform memory address location. Use PACKET when waveform memory is loaded using WAVE; use PACLIT when waveform memory is loaded using WMEM.

Advancing to the Next Packet. There are three ways to advance to the next packet in a sequence: automatic (AUTO), external trigger (EXT) or HP-IB trigger (BUS). The mode is specified when packet information is entered.

In AUTO mode the sequencer completes the number of scans specified for a wave segment and then advances to the next packet. The specified number of scans can range from 1 to 65 536.

In EXT trigger mode, the packet repeats continuously until the Synthesizer receives an external trigger signal from an external device.

If the specified advance mode is BUS, the packet repeats continuously until the Synthesizer receives the ADVSEQ program command.

When either an EXT or BUS trigger is received, the Synthesizer completes the current scan before advancing to the next packet in the sequence.

After all the packets have been scanned, the sequencer returns to the first packet and repeats the sequence until a command arrives from the controller to do otherwise.

3-12. Turn RF Output On

Use the program command OUTPUT ON to turn on the RF output, if it is off. The Synthesizer always powers up with the RF output turned on, but it is at 0 Vdc when the sequencer is not running.

3-13. Starting the Sequencer

The last step to creating a waveform is to start the sequencer. Use program command GO. This starts the sequencer running at the first packet in the sequence. While the sequencer is running, the front panel RUN annunciator is lit.

3-14. SUMMARY OF NAMED FILES VERSUS ADDRESS LOCATION COMMANDS

Several Synthesizer functions can access data by address or by file name. These functions are shown in Table 3-2, along with the appropriate program commands.

Table 3-2. Program Commands for Functions Using File Names or Addresses

Function	Program Command	
	File Name	Address
Load data into waveform memory	WAVE	WMEM
Set up EXTERNAL DATA INPUT port to accept data into waveform memory	EXDWAVE	EXDWM
Return data in waveform memory	WAVE?	WMEM?
Load data into sequencer memory	PACKET	PACLIT
Return packet parameters for existing sequence	DIR? SEQ	PACLIT?
Set marker	MARKER	MARKADD

3-15. STORING WAVEFORMS TO DISC

Waveforms can be stored from the Synthesizer to disc for playback at a later time. The following HP-IB commands query the Synthesizer for the information necessary to reproduce waveforms being generated by the Synthesizer:

- WMEM? — returns contents of waveform memory
- SEQUEN? — returns contents of sequencer memory
- LRNDIR? — returns waveform memory file directory

Storing Waveforms to Disc (cont'd)

- *LRN? — returns hardware settings

Example 3-3 provides an example of how to use these commands in a program.

3-16. PLAYING BACK WAVEFORMS FROM DISC

Waveforms can be loaded from disc to the Synthesizer. Data should be loaded into the Synthesizer in the following order:

- Load waveform data (use program command WMEM)
- Load file directory (use program command LRNDIR)
- Load sequencer data (use program command SEQUEN)
- Load hardware settings (return file containing settings).

3-17. EXAMPLE 3-3

Example 3-3 shows how to store the entire Synthesizer memory to disc for backup and how to reload the Synthesizer from the disc.

NOTE

Example 3-3 is included in the HP 11775A Waveform Generation Software package. The program, titled Copyaws, is located on WGL Application Disc #1.

Example 3-3 has the following limitations:

- No error checking is performed on data input to the program.
- The program backs up the Synthesizer's memory. The 128k waveform memory will fit on one single-sided disc but there is no room for anything else. When using single-sided discs, it is probably best to store the waveform memory on one disc. Use another disc to store the sequencer information, the file directory and the hardware settings.

To store Synthesizer memory to disc, select the following program options:

Program Option	Description
0	Copy waveform memory to disc
2	Copy sequencer memory to disc
4	Copy file directory to disc (this option is not needed if data in waveform memory was loaded using addresses instead of file names)
6	Copy hardware settings to disc.

The program will ask you to enter a file name. Enter the name of a file to which you want to store the data on disc. Do not put quotation marks (" ") around the file name.

To load the waveforms stored on disc back into the Synthesizer, select the following program options in the order listed:

Program Option	Description
1	Copy waveform memory data to Synthesizer
5	Copy file directory to Synthesizer
3	Copy sequencer memory data to Synthesizer
7	Copy hardware settings to Synthesizer

The program will ask you to enter a file name. Enter the name of the disc file where the data is stored. Do not put quotation marks (" ") around the file name.

Example 3-3. Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory (1 of 4)

```
1     ! COPYING SYNTHESIZER MEMORY TO DISC AND COPYING DISC
2     ! TO SYNTHESIZER MEMORY
3     !
5     DIM A$[200]
6     INTEGER Choice
7     PRINT "0 : COPY ALL WAVEFORM DATA FROM SYN. TO DISC"
8     PRINT "1 : COPY WAVEFORM DATA FROM DISC TO SYN."
9     PRINT "2 : COPY ALL SEQUENCER DATA FROM SYN. TO DISC"
10    PRINT "3 : COPY SEQUENCER DATA FROM DISC TO SYN."
11    PRINT "4 : COPY FILE DIRECTORY FROM SYN.  TO DISC"
12    PRINT "5 : COPY FILE DIRECTORY FROM DISC TO SYN."
13    PRINT "6 : COPY HARDWARE SETTINGS FROM SYN. TO DISC"
14    PRINT "7 : COPY HARDWARE SETTINGS FROM DISC TO SYN."
15    PRINT "8 : COPY WAVEFORM DATA FROM DISC TO SYN. USING GPIO"
16    PRINT "9 : EXIT"
17    INPUT "ENTER A CHOICE",Choice
18    WHILE Choice<>9
19      LINPUT "ENTER FILE NAME",Name$
20      IF TRIM$(Name$)<>"" THEN
21      Copyaws(Choice,Name$)
22      INPUT "ENTER A CHOICE",Choice
23      ELSE
24      Choice=8
25      END IF
26    END WHILE
27    END
28    SUB Copyaws(INTEGER Opcode,Name$)
29      ! THIS SUBROUTINE COPIES ALL WAVEFORM DATA, SEQUENCER DATA, DIRECTORY DATA
30      ! AND HARDWARE SETTINGS FROM THE HP 8770A TO DISC.
31      ! VALID OPCODES
32      ! 0 : COPY ALL WAVEFORM DATA FROM AWS TO DISC.
33      ! 1 : COPY WAVEFORM DATA FROM DISC TO AWS.
34      ! 2 : COPY ALL SEQUENCER DATA FROM AWS TO DISC.    SEQUEN?
35      ! 3 : COPY SEQUENCE DATA FROM DISC TO AWS.         SEQUEN
36      ! 4 : COPY DIRECTORY FROM AWS TO DISC.
37      ! 5 : COPY DIRECTORY FROM DISC TO AWS.
38      ! 6 : COPY ALL HARDWARE SETTINGS FROM AWS TO DISC.
39      ! 7 : COPY HARDWARE SETTINGS FROM DISC TO AWS.
40      ! 8 : COPY ALL WAVEFORM DATA FROM DISC TO AWS WITH GPIO
41      !
42      DIM Hardware$[100]                          ! HOLDS HARDWARE SETTINGS
43      DIM Sequence$[16384]                        ! HOLDS SEQUENCER DATA
44      INTEGER Filetype                            ! USED BY WGL AND THIS
45                                                  ! PROGRAM -1=AWS WAVEFORMS.
46                                                  !         -3=SEQUENCER
47                                                  !         -4=HARDWARE
48                                                  !         -5=DIRECTORY
49      REAL Bytetotal                              ! # OF BYTES IN WAVEFORM.
50      INTEGER Buf(1280) BUFFER,Bytecount          ! BUFFER TO TRANSFER DATA.
51                                                  ! DATA FROM AWS TO BUF(*)
52                                                  ! FROM BUF(*) TO DISC.
53                                                  ! Bytecount=HOW MANY BYTES
54                                                  ! BEING TRANSFERRED.
55      Bytecount=2*1280                            ! MAX BYTES THAT CAN BE
56                                                  ! TRANSFERRED AT ONE TIME
57                                                  ! IS 2560.
58      Hpib=719                                    ! AWS HP-IB ADDRESS.
59      ASSIGN @Aws TO Hpib
60      ASSIGN @Buf TO BUFFER Buf(*);FORMAT OFF     !           "
61                                                  !
62      ON Opcode+1 GOSUB Wave_to_disc,Wave_to_aws,Seq_to_disc,Seq_to_aws,Dir_to_
disc,Dir_to_aws,Hw_to_disc,Hw_to_aws,Gpio_to_aws
63      SUBEXIT
64                                                  !
65                                                  !
66  Wave_to_disc:               ! COPY ALL WAVEFORM DATA ADDRESSES 0-131071 TO DISC
```

Example 3-3. Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory (2 of 4)

```
67   !
68     ASSIGN @In TO Hpib;FORMAT OFF                    ! WAVEFORM COMING FROM AWS
69     Filetype=-1                                      ! WAVEFORM FILE TYPE
70     Bytetotal=131072*2                               ! TOTAL NUMBER OF BYTES
71     Datasize=INT(Bytetotal/256.0+1)                  ! NUMBER OF RECORDS NEEDED
72     CREATE BDAT Name$,Datasize                       ! CREATE Name$ FILE
73     ASSIGN @Out TO Name$;FORMAT OFF                  ! DATA GOING TO Name$
74     DISP "COPYING WAVEFORM DATA TO "&Name$&" FROM AWS"
75     OUTPUT @Aws;"FORMAT UNSIGN"                      ! VALUES BETWEEN 0 AND 4095
76     OUTPUT @Aws;"WMEM? BIN,0,131072"                 ! BINARY, START ADDR, LNGTH
77     OUTPUT @Out;Filetype,Bytetotal                   ! FILE HEADER
78     GOSUB Transfer                                   ! TRANSFER FROM AWS TO DISC
79     ASSIGN @Out TO *                                 ! CLEAR OUTPUT PATH NAME
80     ASSIGN @In TO *                                  ! CLEAR INPUT PATH NAME
81     RETURN
82
83
84
85 Wave_to_aws:    !  COPY WAVEFORMS FROM DISC TO AWS
86     ASSIGN @In TO Name$;FORMAT OFF                   ! WAVEFORM DATA FROM Name$
87     ASSIGN @Out TO Hpib;FORMAT OFF                   ! GOING TO AWS.
88     ENTER @In;Filetype                               ! IS THIS WAVEFORM DATA?
89     IF Filetype=-1 THEN
90       DISP "COPYING WAVEFORM TO AWS FROM "&Name$
91       ENTER @In;Bytetotal                            ! # OF BYTES TO TRANSFER
92       OUTPUT @Aws;"FORMAT UNSIGN"                    ! VALUES BETWEEN 0 AND 4095
93       OUTPUT @Aws USING "#,K";"WMEM 0,"              ! BINARY, START ADDR=0
94       GOSUB Transfer                                 ! TRANSFER FROM DISC TO AWS
95     ELSE
96       PRINT "IMPROPER FILE TYPE"
97     END IF
98     ASSIGN @In TO *                                  ! CLEAR INPUT PATH NAME
99     ASSIGN @Out TO *                                 ! CLEAR OUTPUT PATH NAME
100    RETURN
101  !
102
103
104
105 Seq_to_disc:  ! COPY SEQUENCER DATA TO DISC
106    Filetype=-3                                      ! FILE HEADER
107    ON TIMEOUT 7,1 GOTO Done_                        ! DON'T WAIT FOR EVER
108    OUTPUT @Aws;"SEQUEN? BIN"                        ! ASK FOR SEQ DATA
109    ENTER @Aws;Sequence$                             ! LOAD SEQUENCER DATA
110 Done_:!
111    OFF TIMEOUT
112    Length=LEN(Sequence$)                            ! NUMBER OF BYTES IN STRING
113    Ssize=INT((Length+2)/256.0+1)                    ! # RECORDS NEEDED TO STORE
114    CREATE BDAT Name$,Ssize                          ! CREATE FILE FOR SEQ DATA
115    ASSIGN @Out TO Name$;FORMAT OFF                  ! PATH NAME TO FILE
116    DISP "COPYING SEQUENCE FROM AWS TO"&Name$
117    OUTPUT @Out;Filetype,Sequence$                   ! STORE TO DISC
118    ASSIGN @Out TO *                                 ! CLEAR FILE PATH NAME
119    RETURN
120
121
122
123 Seq_to_aws:    !  COPY SEQUENCER DATA FROM DISC TO AWS
124    ASSIGN @In TO Name$;FORMAT OFF                   ! FILE WITH SEQUENCE DATA
125    ENTER @In;Filetype                               ! READ FILE HEADER
126    IF Filetype=-3 THEN                              ! CHECK FOR SEQUENCE FILE
127      DISP "COPYING SEQUENCE TO AWS FROM "&Name$
128      ENTER @In;Sequence$                            ! READ SEQUENCE DATA
129      OUTPUT @Aws;"PURGE SEQ"                        ! CLEAR SEQUENCER MEMORY
130      OUTPUT @Aws USING "#,K";"SEQUEN "              ! TELL AWS ITS COMING
131      OUTPUT @Aws USING "#,K";Sequence$,END          ! SEND SEQUENCE TO AWS
132    ELSE
```

Example 3-3. Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory (3 of 4)

```
133       PRINT "IMPROPER FILE TYPE"
134     END IF
135     RETURN
136                                                    !
137                                                    !
138                                                    !
139 Dir_to_disc:    ! COPY DIRECTORY FROM AWS TO DISC
140     Filetype=-5                                    ! FILE HEADER
141     Bytetotal=14032                                ! # BYTES IN DIRECTORY
142     CREATE BDAT Name$,55                           ! FILE TO HOLD DIRECTORY
143     ASSIGN @Out TO Name$;FORMAT OFF                ! PATH NAME TO FILE
144     ASSIGN @In TO Hpib;FORMAT OFF                  ! USED TO SPEED TRANSFER
145     OUTPUT @Aws;"LRNDIR?"                          ! ASK FOR WAVE DIRECTORY
146     DISP "COPYING DIRECTORY FROM AWS TO "&Name$
147     OUTPUT @Out;Filetype                           ! OUTPUT FILE HEADER
148     GOSUB Transfer                                 ! COPY DIRECTORY TO DISC
149     ASSIGN @In TO *                                ! CLEAR FILE PATH
150     ASSIGN @Out TO *                               ! CLEAR FILE PATH
151     RETURN
152                                                    !
153                                                    !
154                                                    !
155 Dir_to_aws:    !  COPY DIRECTORY FROM DISC TO AWS
156     ASSIGN @In TO Name$;FORMAT OFF                 ! FILE HOLDING DIRECTORY
157     ASSIGN @Out TO Hpib;FORMAT OFF                 ! FOR SPEED
158     ENTER @In;Filetype                             ! READ FILE HEADER
159     IF Filetype=-5 THEN
160       Bytetotal=14032                              ! # BYTES IN DIRECTORY
161       OUTPUT @Aws USING "#,K";"LRNDIR "            ! TELL AWS DIR. IS COMING
162       DISP "COPYING DIRECTORY TO AWS FROM "&Name$
163       GOSUB Transfer                               ! COPY DIRECTORY TO AWS
164     ELSE
165       PRINT "IMPROPER FILE TYPE"
166     END IF
167     ASSIGN @In TO *                                ! CLEAR FILE PATH NAME
168     ASSIGN @Out TO *                               ! CLEAR FILE PATH NAME
169     RETURN
170                                                    !
171                                                    !
172                                                    !
173 Hw_to_disc:    !  COPY HARDWARE SETTINGS FROM AWS TO DISC
174     Filetype=-4                                    ! FILE HOLDS HW SETTINGS
175     CREATE BDAT Name$,1                            ! FILE TO HOLD HW SETTINGS
176     ASSIGN @Out TO Name$;FORMAT OFF                ! PATH TO HARDWARE FILE
177     OUTPUT @Aws;"*LRN?"                            ! ASK FOR HARDWARE SETTINGS
178     ENTER @Aws;Hardware$                           ! READ SETTINGS
179     DISP "COPYING HARDWARE SETTINGS FROM AWS TO "&Name$
180     OUTPUT @Out;Filetype                           ! FILE HEADER
181     OUTPUT @Out;Hardware$                          ! STORE SETTINGS ON DISC
182     ASSIGN @Out TO *                               ! CLEAR FILE PATH NAME
183     RETURN
184                                                    !
185                                                    !
186                                                    !
187 Hw_to_aws:     !  COPY HARDWARE SETTINGS FROM DISC TO AWS
188     ASSIGN @In TO Name$;FORMAT OFF                 ! FILE WITH HW SETTINGS
189     ENTER @In;Filetype                             ! READ FILE HEADER
190     IF Filetype=-4 THEN                            ! HARDWARE FILE?
191       DISP "COPYING HARDWARE SETTINGS TO AWS FROM "&Name$
192       ENTER @In;Hardware$                          ! GET SETTINGS FROM DISC
193       OUTPUT @Aws;Hardware$                        ! SEND TO AWS
194     ELSE
195       PRINT "IMPROPER FILE TYPE"
196     END IF
197     ASSIGN @In TO *                                ! CLEAR FILE PATH NAME
198     RETURN
```

Example 3-3. Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory (4 of 4)

```
199                                                          !
200                                                          !
201                                                          !
202 Gpio_to_aws:  ! COPY WAVEFORMS FROM DISC TO AWS USING GPIO
203     ASSIGN @In TO Name$;FORMAT OFF                       ! WAVEFORM DATA FROM Name$
204     ASSIGN @Out TO 12;WORD                               ! GOING TO AWS. 12=GPIO
205     ENTER @In;Filetype                                   ! IS THIS WAVEFORM DATA?
206     IF Filetype=-1 THEN
207       DISP "COPYING WAVEFORM TO AWS FROM "&Name$
208       ENTER @In;Bytetotal                                ! # OF BYTES TO TRANSFER
209       OUTPUT @Aws;"PURGE BOTH"                           ! CLEAR WAVEFORM MEMORY
210       OUTPUT @Aws;"FORMAT UNSIGN"                        ! VALUES BETWEEN 0 AND 4095
211       OUTPUT @Aws;"EXDWAV ALL, "&VAL$(Bytetotal/2)       ! BINARY, WAVE NAME= ALL
212       GOSUB Transfer                                     ! TRANSFER FROM DISC TO AWS
213     ELSE
214       PRINT "IMPROPER FILE TYPE"
215     END IF
216     ASSIGN @In TO *                                      ! CLEAR INPUT PATH NAME
217     ASSIGN @Out TO *                                     ! CLEAT OUTPUT PATH NAME
218     RETURN
219                                                          !
220                                                          !
221                                                          !
222 Transfer:    ! TRANSFER WAVEFORM OR DIRECTORY DATA
223     REPEAT                                               !
224       Bytecount=MIN(Bytetotal,2560)                      ! START TRANSFER OF DATA
225       TRANSFER @In TO @Buf;COUNT Bytecount,END,WAIT      ! # BYTES TO BE TRANSFERRED
226       IF Bytecount=Bytetotal THEN                        !  XFER TO BUF
227         TRANSFER @Buf TO @Out;COUNT Bytecount,END,WAIT   !  CHECK IF DONE
228       ELSE                                               !  YES? CLOSE OUT FILE
229         TRANSFER @Buf TO @Out;COUNT Bytecount,WAIT       !  NO? LEAVE FILE OPEN
230       END IF
231       Bytetotal=Bytetotal-Bytecount                      ! HOW MUCH LEFT TO XFER
232     UNTIL Bytetotal=0                                    ! END OF REPEAT
233     RETURN
234 SUBEND
```

3-16. EXTERNAL DATA INPUT

The EXTERNAL DATA INPUT port is located on the rear panel of the Synthesizer. It is used for high speed transfer of data to the internal waveform or sequencer memory.

Data cannot be output using this port. Only the HP-IB port can output data. Although data is input through the EXTERNAL DATA INPUT port, the commands to prepare the port to accept data are sent over the HP-IB.

Table 3-3 shows the HP-IB program commands available for use with the EXTERNAL DATA INPUT port.

Typically, the EXTERNAL DATA INPUT port is connected to a GPIO port on the computer via an HP 11738A cable. However, the port can mate with any port that meets the critical timing of the handshake and data lines. See Figure 2-5.

Example 3-4 is a program that transfers data into waveform memory using the EXTERNAL DATA INPUT port. Running the program in Example 3-4 requires a computer with a GPIO interface and GPIO BASIC binary extension. Refer to Table 1-3, Recommended Test Equipment, for specific Hewlett-Packard equipment fulfilling these requirements.

3-19. ERRORS

The Synthesizer can report three types of errors over the HP-IB: bus errors, command errors and execution errors.

Bus errors are caused by hardware failures. They are described in Section VIII of the Operating and Service Manual.

Table 3-3. External Data Input Commands

Command	Description
EDAR	Programs edge sense of EDAR handshake line.
EDV	Programs edge sense of EDV handshake line.
EXDABT	Aborts any pending transfers through the EXTERNAL DATA INPUT port.
EXDAT?	Returns the number of words remaining to be transferred through the EXTERNAL DATA INPUT port.
EXDSEQ	Sets up EXTERNAL DATA INPUT port to accept data into sequencer memory.
EXDWAV	Sets up EXTERNAL DATA INPUT port to accept data into waveform memory. Data is accessed by file names.
EXDWM	Sets up the EXTERNAL DATA INPUT port to accept data for waveform memory. Data is accessed by addresses.

ERRORS (cont'd)

Command errors are generated when the Synthesizer receives a command string that it cannot execute. Command strings cannot be executed for three reasons:

1. invalid mnemonic
2. invalid data type
3. improper terminator.

Refer to CMDERR in the Programming Commands for additional information on command errors.

Execution errors are generated when the Synthesizer cannot execute a command because a parameter for that command is out of range. Refer to EXERR in the Programming Commands for additional information on execution errors.

The Status Register can be programmed so that command errors and/or execution errors will generate a Service Request (SRQ) and light the front panel SRQ annunciator. Example 3-5 in Programming the Status Register shows an example of an error checking routine that interrupts the computer when a command or execution error occurs.

Example 3-4. Loading Waveform Memory Using the EXTERNAL DATA INPUT Port

```
10    ! LOADING WAVEFORM MEMORY USING THE EXTERNAL DATA INPUT PORT
20                                          !
30    INTEGER A(0:1023),I                   ! DIMENSION ARRAYS
40    ASSIGN @Gpio TO 12;FORMAT OFF,WORD    ! GPIO ADDRESS=12; SEND ALL DATA
50                                          ! OVER GPIO AS WORDS (2 BYTES)
60                                          !
70    OUTPUT 719;"EXDABT; PURGE BOTH"       ! ABORT ANY PENDING DATA TRANSFERS
80                                          ! OVER EXTERNAL DATA INPUT PORT;
90                                          ! PURGE WAVEFORM AND SEQUENCER MEMORY
100                                         !
110   OUTPUT 719;"EXDWAV TEMP,1024"         ! PREPARE EXTERNAL DATA INPUT PORT TO
120                                         ! RECEIVE 1024 ELEMENTS
130                                         ! AND STORE THEM IN A
140                                         ! WAVEFORM MEMORY FILE NAMED "TEMP"
150                                         !
160   FOR I=0 TO 1023                       ! CREATE DATA ARRAY
170     A(I)=I*2
180   NEXT I
190   BEEP 400,.5                           ! BEEP TO INDICATE DATA ARRAY HAS
200                                         ! BEEN GENERATED
210                                         !
220   OUTPUT @Gpio;A(*)                     ! SEND THE DATA OVER THE GPIO
230                                         !
240   OUTPUT 719;"PACKET TEMP,1,AUTO;GO"    ! CREATE SEQUENCE OF 1 PACKET USING
250                                         ! WAVEFORM MEMORY FILE "TEMP" AND
260                                         ! START SEQUENCER
270   END
```

3-20. HEWLETT-PACKARD INTERFACE BUS REMOTE OPERATION

The Synthesizer can be operated through the Hewlett-Packard Interface Bus (HP-IB). HP-IB is Hewlett-Packard's implementation of IEEE Standard 488 and the identical ANSI Standard MC1.1. Except for the LINE switch, all Synthesizer operations are fully programmable via HP-IB.

In this manual, program codes are listed in ASCII code. Table 3-5, Commonly Used Code Conversions, includes a listing of ASCII characters and some commonly used equivalent codes. The table is located at the end of this discussion on HP-IB.

For more information about HP-IB, refer to IEEE Standard 488 (or the identical ANSI Standard MC1.1), the Hewlett-Packard Electronic Systems and Instruments catalog, and the booklet, "Tutorial Description of the Hewlett-Packard Interface Bus" (HP part number 5952-0156).

3-21. HP-IB Compatibility

The Synthesizer's complete bus compatibility as defined by IEEE Standard 488 (and the identical ANSI Standard MC1.1) is described at the end of Table 3-4. The programming capability is further described in the twelve HP-IB messages in the left-hand column of Table 3-4. Foremost among these is the Data message. Data messages contain the program codes that set the instrument's mode of operation.

3-22. Remote Mode

The Synthesizer always powers up in local mode. It switches to remote operation upon receipt of the Remote message. The Remote message has two parts:

1. remote enable (REN) bus control line set true, and
2. device listen address received once (while REN is true).

When the Synthesizer switches to remote, the front panel REMOTE annunciator turns on.

In remote, the Synthesizer can be addressed to talk or listen. When addressed to listen, the Synthesizer responds to the Data, Clear (SDC), and Local messages. When addressed to talk, the Synthesizer can issue the Data and Status Byte messages. Whether addressed or not, the Synthesizer responds to the Clear (DCL), Local Lockout, Clear Lockout/Set Local, and Abort messages. In addition, the Synthesizer may issue the Require Service and Status Bit messages.

3-23. Local Mode

In local, the Synthesizer responds to to the Remote message. Whether it is addressed or not, it also responds to the Clear, Local Lockout, Clear Lockout/Set Local and Abort messages. When addressed to talk, the instrument can issue Data messages and the Status Byte message. Whether addressed or not, the instrument can issue the Require Service and Status Bit messages.

3-24. Addressing

The Synthesizer's address is set to 19 at the factory by a switch located on the rear panel. However, any address between 0 and 30 can be assigned to the Synthesizer. Section II describes a procedure to set the Synthesizer's address switch.

The Synthesizer interprets the byte on the eight HP-IB data lines (DIO1—8) as an address or a bus command if the bus is in the command mode. The command mode is defined as attention control line (ATN) true and interface clear control line (IFC) false. Whenever the Synthesizer is addressed (whether in local or remote), either the TALK or LISTEN annunciator on the front panel turns on.

If the NORMAL/LISTEN ONLY segment of the HP-IB address switch is set to "0," the Synthesizer is placed in listen only mode. The instrument then responds to all Data messages, and the Clear message. It is, however, inhibited from responding to the Local or Abort messages and from responding to a serial poll with the Status Byte message.

Listen Only mode is provided to allow the Synthesizer to accept programming from devices other than controllers.

3-25. Data Messages

The Synthesizer communicates on the interface bus primarily with Data messages. Data messages consist of one or more bytes sent over the bus' eight data lines when the bus is in the data mode (ATN bus control line false).

3-26. Receiving the Data Message

The Synthesizer responds to Data messages when it is enabled to remote (REN bus control line true)

Table 3-4. Message Reference Table (1 of 2)

HP-IB Message	Appli-cable	Response	Related Commands & Controls	Interface Functions*
Data	Yes	All Synthesizer functions except setting the LINE switch are bus-programmable. Most instrument settings can be read over the HP-IB		AH1 SH1 T5 TE0 L3 LE0
Trigger	No	The Synthesizer does not have a device trigger capability.	GET	DT0
Clear	Yes	The Synthesizer responds equally to Device Clear (DCL) and Selected Device Clear (SDC) bus commands by clearing any incomplete entries or messages. The Clear capability does not reset instrument parameters.	DCL SDC	DC1
Remote	Yes	Remote mode is enabled when the REN bus control line is true. Remote mode is not entered, however, until the first time the Synthesizer is addressed to listen. The front panel REMOTE annuncinator lights when the instrument is actually in remote mode.	REN	RL1
Local	Yes	The Synthesizer returns to local mode. It responds to the Go To Local (GTL) bus command. When entering local mode, no instrument settings or functions are changed.	GTL	RL1
Local Lockout	Yes	By strict definition, the Synthesizer can respond to the Local Lockout message. However, this capability would probably not be used because there are no front panel keys to lock out.	LLO	RL1
Clear Lockout/ Set Local	Yes	By strict definition, the Synthesizer returns to local and local lockout is cleared when the REN bus control line goes false.	$\overline{\text{REN}}$	RL1
Pass Control/ Take Control	No	The Synthesizer has no controller capability.		C0
Require Service	Yes	The Synthesizer sets the SRQ bus control line true when one or more of the service request conditions occurs, if it has been enabled to send the Require Service message for that condition. Refer to Programming the Status Register in this section for a description of conditions that can be enabled to generate the Require Service message.	SRQ	SR1
Status Byte	Yes	The Synthesizer responds to a Serial Poll Enable (SPE) bus command by sending an 8-bit byte when addressed to talk. If the instrument is holding the SRQ bus control line true (issuing the Require Service message), bit 7 in the Status Byte and the bit representing the condition causing the Require Service message to be issued will both be true. The bits in the Status Byte are latched but can be cleared by reading the Event Register(s) of the true bit(s) or by sending the *CLS program command to the Synthesizer.	SPE SPD	T5

Table 3-4. Message Reference Table (2 of 2)

HP-IB Message	Appli-cable	Response	Related Commands & Controls	Interface Functions*
Data Bit	Yes	The Synthesizer responds to a Parallel Poll Enable (PPE) bus command, if enabled, by sending a bit on a controller selected HP-IB data line.	PPE PPD PPC PPU	PP1
Abort	Yes	The Synthesizer stops talking and listening.	IFC	T5 TE0 L3 LE0

*Commands, control lines, and interface functions are defined in IEEE Std 488-1978. Knowledge of these may not be necessary if your controller's manual describes programming in terms of the twelve HP-IB messages shown in the left column.

Complete HP-IB capability as defined in IEEE Std 488-1978 and ANSI Std MC1.1 is SH1, AH1, T5, TE0, L3, LE0, SR1, RL1, PP1, DC1, DT0, and C0.

Receiving the Data Message (cont'd)

and it is addressed to listen. The instrument remains addressed to listen until it receives its talk address, an Abort message, or a universal unlisten command.

Data Message Input Format. Input Data messages program virtually all instrument functions. The Data message contains a string of device-dependent commands (program commands). Program commands within a Data message are executed when a message unit terminator is received. Valid message unit terminators are a semicolon (;), linefeed (LF) and End-or-Identify (EOI) asserted with the last byte in the message unit. The following format rules must be observed for all input Data messages:

- Each Data message must be terminated by a <LF> or by asserting the EOI bus signal line with the last byte in the message.
- The carriage return (CR) is not required before <LF>. Preceding <LF>, <CR> is treated as "no operation" and may be repeated indefinitely.
- When several commands are sent in the same Data message, a semicolon (;) must separate program commands from each other.
- Multiple arguments for a command must be separated by commas (,).

Program Command Format. Program commands consist of a header, which in most cases is followed by a parameter field. Command parameters can be words or numbers. For example in the command string "SINPQ FREQ1, 1, 1024," SINPQ is the header and FREQ1, 1 and 1024 are command parameters.

Program command parameters can be one of four types:

- Decimal Numeric — Any integer, floating point, or exponential value. Valid characters are 0 through 9, E, the plus sign (+), the minus sign (−) and the decimal point (.). Spaces are allowed between the +, − or E and digits but not between digits or between a decimal point and digits.
- Nondecimal Numeric — Binary, octal or hexidecimal data. Valid binary characters are 0, 1 and X (for don't care). Valid octal characters are 0 through 7. Valid hexidecimal characters are 0 through 9 and A through F.
- Character — Some commands require alpha arguments, such as "ON," "OFF" or file names. These arguments are ASCII strings that start with an alpha character and are followed by characters A through Z, 0 through 9 or "_" (underscore).
- Blocks — A block of binary data in the A, B, C, I or L format as defined in IEEE Std 728. Blocks are useful for transferring large quantities of data.

Receiving the Data Message (cont'd)

The format for each block is shown below:

#A <length _ 16> <data bytes>

#B <length _ 16> <data bytes> <checksum>

#C <length _ 16> <data bytes> <check byte> <check byte>

#I <data bytes> <data byte with EOI>

#L <length _ 32> <data bytes>

#A, #B, #C, #I and #L are ASCII bytes. They specify the format of the data to follow.

Length _ 16 is a 16 bit unsigned binary integer. Length _ 32 is a 32 bit unsigned binary integer. Length specifies the number of data bytes to follow. The most significant bit is always sent first. In the B-block format, the length includes the checksum byte. In the C-block format, the length includes the two check bytes.

Data bytes are 8-bit binary data. It takes two bytes to load a data point in waveform memory. Therefore, a parameter field must contain an even number of data bytes before the Data message is terminated. It is possible to have an odd number of bytes in an A, B, C or L block but another block with an odd number of bytes must be appended to the first block before terminating the message. An I block is terminated by asserting EOI with the last data byte. Therefore, I blocks must always contain an even number of bytes.

The checksum is calculated by adding all the data bytes in a B block together and then dividing by 256. The negative value of the remainder is the checksum. The length bytes are not included in the checksum.

For C blocks, the two check bytes contain the remainder from division of the data byte stream by the CRC16 forward polynomial (that is, $x^{16} + x^{15} + x^2 + 1$).

The general rules of program command format are as follows:

- The Synthesizer sends and receives Data messages in standard ASCII code, unless otherwise noted.
- The Synthesizer responds equally to upper and lower case characters.
- A space is always required between a header and its parameter field.
- Parameter fields containing multiple parameters require a comma (,) to delimit individual parameters.

Errors in Data message syntax are trapped and can be reported via the HP-IB. Refer to CMDERR in the Programming Commands for details.

Program Order Considerations. Commands are interpreted as they are received and found to be syntactically correct. Commands preceding an error in a multi-command message are executed up to the point where the error is detected. The invalid entry is ignored and subsequent valid entries are processed in the normal fashion.

Waveform memory must be loaded with data prior to defining sequences (for example, program command "WAVE" must be executed before "PACKET").

3-27. Sending the Data Message

The Synthesizer sends Data messages when addressed to talk. The instrument remains configured to talk until it is unaddressed to talk by the controller. To unaddress the Synthesizer, the controller must send the Synthesizer's listen address, a new talk address, an Abort message or a universal untalk command.

Before the instrument is addressed to talk, the desired output data must be specified with the appropriate input Data message, a query. Otherwise, the instrument cannot complete the bus transaction. Queries are program commands that end with a question mark (?). The Synthesizer responds to a query by outputting a Data message containing the value or state of the associated parameter.

Queries, when executed, cause their replies to be placed in the output buffer. Multiple queries on one line result in the last reply writing over the previous replies and an execution error being generated.

All output Data messages are sent with EOI true with the last byte. Refer to Programming Commands in this section for the output format of each query.

3-28. Receiving the Clear Message

The Synthesizer responds to the Clear message by clearing any incomplete entries or messages. The Synthesizer responds equally to the Selected Device Clear (SDC) bus command when addressed to listen, and the Device Clear (DCL) bus command whether addressed or not.

3-29. Receiving the Trigger Message

The Synthesizer does not respond to the Trigger message.

3-30. Receiving the Remote Message

The Remote message has two parts. First, the remote enable bus control line (REN) is held true, then the device listen address is sent by the controller. These two actions combine to place the Synthesizer in remote mode. Thus, the Synthesizer is enabled to go into remote when the controller begins the Remote message, but it does not actually switch to remote until addressed to listen the first time. No instrument settings are changed by the transition from local to remote.

3-31. Receiving the Local Message

The Local message is the means by which the controller sends the Go To Local (GTL) bus command. If addressed to listen, the Synthesizer returns to local mode when it receives the Local message.

3-32. Receiving the Local Lockout and Clear Lockout/Set Local Messages

Electrically, the Synthesizer can respond to the Local Lockout and Clear Lockout/Set Local messages. However, it is unlikely these messages would be used because there are no front panel keys to lock out.

3-33. Receiving the Pass Control Message

The Synthesizer does not respond to the Pass Control message because it cannot act as a controller.

3-34. Sending the Require Service Message

The Synthesizer sends the Require Service message by setting the Service Request (SRQ) bus control line true when a previously programmed condition occurs. The Require Service message is cleared when a serial poll is executed by the system controller. During serial poll, the SRQ control line is reset as soon as the instrument places the Status Byte message on the bus. Refer to Programming the Status Register in this section for a description of the conditions that can be enabled to generate the Require Service message. If no conditions are enabled, the Require Service message is disabled.

When the Synthesizer is sending the Require Service message, the front panel SRQ annunciator lights. The annunciator is turned off during the serial poll when the SRQ control line is reset.

3-35. Sending the Status Byte Message

After receiving a Serial Poll Enable (SPE) bus command and when addressed to talk, the Synthesizer sends the Status Byte message. The Status Byte message consists of one 8-bit byte in which the bits are set according to the conditions described in Programming the Status Register. (The Status Byte is identical to the Status Register.)

Bits in the status byte are set depending on the instrument state. If a condition occurs that causes one of the bits in the status byte to be set and if its corresponding bit is enabled by the Status Register enable field, the RQS (Require Service) bit is set and the Require Service message is sent.

If the RQS bit is set, indicating that the instrument sent the Require Service message, and a serial poll is executed, the RQS bit will be cleared. All other bits in the status byte remain unchanged. Refer to Programming the Status Register for information on clearing individual bits.

If the Synthesizer is set to listen-only mode, it does not respond to the SPE and SPD (Serial Poll Disable) bus commands and cannot send the Status Byte.

3-36. Sending the Status Bit Message

The Synthesizer sends the Status Bit message (if configured to do so) as part of the interface's response to the Parallel Poll Enable (PPE) bus command. In order for the Synthesizer to respond to a PPE bus command, the controller must assign the Synthesizer a single HP-IB data line on which to respond. The controller also assigns the logic sense of the bit. Both tasks are accomplished by the Parallel Poll Configure (PPC) bus command. The Synthesizer sets its assigned status bit true if one or more of the bits in the Status Register is true and has been enabled. Refer to Programming the Status Register.

Sending the Status Bit Message (cont'd)
The Synthesizer can send the Status Bit message without being addressed to talk.

The data line on which the Synthesizer is assigned to respond is cleared by sending the Parallel Poll Unconfigure (PPU) bus command. At turn-on, the data line is unassigned.

3-37. Receiving the Abort Message

The Abort message is the means by which the controller sets the Interface Clear (IFC) bus control line true. When the Abort message is received, the Synthesizer becomes unaddressed and stops talking and listening.

Table 3-5. Commonly Used Code Conversions

ASCII	Binary	Octal	Decimal	Hexa-decimal
NUL	00 000 000	000	0	00
SOH	00 000 001	001	1	01
STX	00 000 010	002	2	02
ETX	00 000 011	003	3	03
EOT	00 000 100	004	4	04
ENQ	00 000 101	005	5	05
ACK	00 000 110	006	6	06
BEL	00 000 111	007	7	07
BS	00 001 000	010	8	08
HT	00 001 001	011	9	09
LF	00 001 010	012	10	0A
VT	00 001 011	013	11	0B
FF	00 001 100	014	12	0C
CR	00 001 101	015	13	0D
SO	00 001 110	016	14	0E
SI	00 001 111	017	15	0F
DLE	00 010 000	020	16	10
DC1	00 010 001	021	17	11
DC2	00 010 010	022	18	12
DC3	00 010 011	023	19	13
DC4	00 010 100	024	20	14
NAK	00 010 101	025	21	15
SYN	00 010 110	026	22	16
ETB	00 010 111	027	23	17
CAN	00 011 000	030	24	18
EM	00 011 001	031	25	19
SUB	00 011 010	032	26	1A
ESC	00 011 011	033	27	1B
FS	00 011 100	034	28	1C
GS	00 011 101	035	29	1D
RS	00 011 110	036	30	1E
US	00 011 111	037	31	1F
SP	00 100 000	040	32	20
!	00 100 001	041	33	21
"	00 100 010	042	34	22
#	00 100 011	043	35	23
$	00 100 100	044	36	24
%	00 100 101	045	37	25
&	00 100 110	046	38	26
'	00 100 111	047	39	27
(	00 101 000	050	40	28
)	00 101 001	051	41	29
*	00 101 010	052	42	2A
+	00 101 011	053	43	2B
,	00 101 100	054	44	2C
—	00 101 101	055	45	2D
.	00 101 110	056	46	2E
/	00 101 111	057	47	2F
0	00 110 000	060	48	30
1	00 110 001	061	49	31
2	00 110 010	062	50	32
3	00 110 011	063	51	33
4	00 110 100	064	52	34
5	00 110 101	065	53	35
6	00 110 110	066	54	36
7	00 110 111	067	55	37
8	00 111 000	070	56	38
9	00 111 001	071	57	39
:	00 111 010	072	58	3A
;	00 111 011	073	59	3B
<	00 111 100	074	60	3C
=	00 111 101	075	61	3D
>	00 111 110	076	62	3E
?	00 111 111	077	63	3F

ASCII	Binary	Octal	Decimal	Hexa-decimal
@	01 000 000	100	64	40
A	01 000 001	101	65	41
B	01 000 010	102	66	42
C	01 000 011	103	67	43
D	01 000 100	104	68	44
E	01 000 101	105	69	45
F	01 000 110	106	70	46
G	01 000 111	107	71	47
H	01 001 000	110	72	48
I	01 001 001	111	73	49
J	01 001 010	112	74	4A
K	01 001 011	113	75	4B
L	01 001 100	114	76	4C
M	01 001 101	115	77	4D
N	01 001 110	116	78	4E
O	01 001 111	117	79	4F
P	01 010 000	120	80	50
Q	01 010 001	121	81	51
R	01 010 010	122	82	52
S	01 010 011	123	83	53
T	01 010 100	124	84	54
U	01 010 101	125	85	55
V	01 010 110	126	86	56
W	01 010 111	127	87	57
X	01 011 000	130	88	58
Y	01 011 001	131	89	59
Z	01 011 010	132	90	5A
[	01 011 011	133	91	5B
\	01 011 100	134	92	5C
]	01 011 101	135	93	5D
^	01 011 110	136	94	5E
—	01 011 111	137	95	5F
`	01 100 000	140	96	60
a	01 100 001	141	97	61
b	01 100 010	142	98	62
c	01 100 011	143	99	63
d	01 100 100	144	100	64
e	01 100 101	145	101	65
f	01 100 110	146	102	66
g	01 100 111	147	103	67
h	01 101 000	150	104	68
i	01 101 001	151	105	69
j	01 101 010	152	106	6A
k	01 101 011	153	107	6B
l	01 101 100	154	108	6C
m	01 101 101	155	109	6D
n	01 101 110	156	110	6E
o	01 101 111	157	111	6F
p	01 110 000	160	112	70
q	01 110 001	161	113	71
r	01 110 010	162	114	72
s	01 110 011	163	115	73
t	01 110 100	164	116	74
u	01 110 101	165	117	75
v	01 110 110	166	118	76
w	01 110 111	167	119	77
x	01 111 000	170	120	78
y	01 111 001	171	121	79
z	01 111 010	172	122	7A
{	01 111 011	173	123	7B
|	01 111 100	174	124	7C
}	01 111 101	175	125	7D
~	01 111 110	176	126	7E
DEL	01 111 111	177	127	7F

3-38. PROGRAMMING THE STATUS REGISTER

3-39. General

The Status Register can be programmed to generate a Service Request (SRQ) whenever a predetermined condition occurs.

The Status Register is an 8-bit register that constantly monitors several possible conditions. Each condition is represented by a bit in the Status Register. Each bit in the Status Register has its own requirements for being set true (to a 1 condition). These requirements are described in detail in the following pages and summarized in Figure 3-12.

3-40. Status Register

Table 3-6 defines the bits in the Status Register and the conditions that can be programmed to set each bit true.

Figure 3-8 shows the decimal value of each bit in the Status Register.

Table 3-6. Status Register Bit Definitions

Bit	Description	Condition That Sets Bit True
8	Always 0	
7	Require	Status Register bits 1, 2, 5, Service and/or 6 are true and have been enabled by the Status Register enable field.
6	Event Bit	One or more of the following conditions occurs and has been enabled by the Event Register enable field: Power On Command Error Execution Error Query Error Operation Complete
5	Message Available	The Synthesizer has received a query and is ready to output data.
4	Always 0	
3	Always 0	
2	Operating Conditions	One or more of the following conditions is true in the Bit 2 Event Register and is enabled by the Bit 2 Event Register enable field: Ready for Bus Trigger External Data Transfer Complete
1	Hardware Errors	One or more of the following conditions is true in the Bit 1 Event Register and is enabled by the Bit 1 Event Register enable field: System Bus Error Reference Oven is Cold No Clock Detected Phase-locked Loop Out of Lock

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	Require Service (RQS)	Events	Message Available	0	0	Operating Conditions	Hardware Errors
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Figure 3-8. Status Register

Status Register (cont'd)
There are two ways to read the current value of the Status Register:

1. Do a serial poll.

2. Send the Synthesizer the query *STB?.

A serial poll returns the value of the register in a single byte. *STB? returns the decimal value of the register in an ASCII string. In both cases, the value of the register is derived by adding together the values of each true bit.

A serial poll clears bit 7 (RQS), if true, but has no effect on other bits in the Status Register. *STB? does not clear any bits in the Status Register.

To clear the Status Register (that is, set the value of all bits in the register to 0), send the Synthesizer the command *CLS.

3-41. Status Register Enable Field

The Status Register enable field defines which bits in the Status Register can generate a Service Request. If no bits are enabled, SRQ cannot be generated.

The Status Register enable field is set by sending the Synthesizer the program command *SRE, followed by a number (0 to 255) representing the value of the Status Register bits to be enabled. It is not necessary to enable bit 7 (RQS).

Bits in the Status Register are logically ANDed with the corresponding bits in the enable field. If the resultant value is equal to one, the Synthesizer sets bit 7 in the Status Register true. This sends a Service Request to the system controller.

For example, to enable bits 1 and 6 (hardware errors and events) to generate a Service Request:

OUTPUT 719;"*SRE 33"

3-42. Bit 1, Hardware Errors

The status of bit 1 in the Status Register is determined by four registers and an enable field:

1. Bit 1 Status Register
2. Bit 1 Event Register
3. Bit 1 Falling Edge Register
4. Bit 1 Rising Edge Register
5. Bit 1 Event Register enable field.

The bits in each of the four registers and the enable field are identical. Each bit represents one of the four conditions capable of generating a hardware error. See Figure 3-9. In addition, the decimal value of each bit is shown. The method by which bits are set is different for each register.

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Locked Loop Out of Lock	No System Clock	Reference Oven Cold	System Bus Error
Value = 8	Value = 4	Value = 2	Value = 1

Figure 3-9. Bit 1, Hardware Errors

Bit 1 Status Register. The Bit 1 Status Register is a read-only register. When one or more of the conditions described in Table 3-7 occurs, the appropriate bit is set true. Bits are cleared (set to 0) when the causing condition is removed.

Table 3-7. Bit 1 Status Register

Bit	Description	Condition That Sets Bit True
4	Phase-locked Loop Out of Lock	Internal 125 MHz phase-locked loop is not phase locked.
3	No System Clock	The system clock is not operational.
2	Reference Oven Cold	Internal 10 MHz reference oscillator oven is cold.
1	System Bus Error	Microprocessor bus error.

To read the Bit 1 Status Register, address the Synthesizer to listen and send the program command B1CST?. Then address the Synthesizer to talk. It will return the decimal value of the register setting. For example,

```
10 OUTPUT 719;"B1CST?"
20 ENTER 719;A
30 PRINT "BIT 1 STATUS REGISTER =";A
40 END
```

If the returned value is 6, the reference oven is cold and no system clock is detected.

Bit 1, Hardware Errors (cont'd)

Bit 1 Event Register. Like the Bit 1 Status Register, the Bit 1 Event Register is a read-only register.

Bits in the Bit 1 Event Register are driven by changes in the Bit 1 Status Register. To set a bit in the Bit 1 Event Register, two things must happen:

1. A bit in the Bit 1 Status Register has changed states (that is, changed from true to false or false to true), and
2. The corresponding bit in the Bit 1 Event Register was previously programmed to respond to the false-to-true (rising edge) or true-to-false (falling edge) transition.

Once set, the bits in the Bit 1 Event Register remain set until the register is read. After the register has been read, it is set to 0. Turning the instrument off or sending the program command *CLS also sets the register to 0.

To read the Bit 1 Event Register, address the Synthesizer to listen and send the program command B1EVT?. Then address the Synthesizer to talk. It returns the value of the true bits.

There are several differences between the Bit 1 Event Register and the Bit 1 Status Register:

1. The Bit 1 Status Register indicates the instrument's current status, whereas the Bit 1 Event Register indicates a change in status since the last time the register was read.
2. The Bit 1 Status Register is strictly informational, whereas the Bit 1 Event Register can be enabled to drive bit 1 in the Status Register.
3. Conditions in the Bit 1 Event Register are edge sensitive; they must be programmed to respond true on the rising edge or falling edge of changes in the Bit 1 Status Register. Conditions in the Bit 1 Status Register are level sensitive; they are true while they are occurring.

Bit 1 Rising and Falling Edge Registers. In order for bits in the Bit 1 Event Register to be set, each bit must be programmed to respond on the rising edge (false-to-true) or falling edge (true-to-false) of changes in the Bit 1 Status Register.

Use program command B1RIS followed by a number from 0 to 15 to cause bits in the Bit 1 Event Register to respond on the rising edge transition. The number represents the value of the bits to be set. (See Figure 3-9.)

Use program command B1FAL followed by a number from 0 to 15 to cause bits in the Bit 1 Event Register to respond on the falling edge transition in the Bit 1 Status Register.

Program commands B1RIS? and B1FAL? return the current setting of the Bit 1 Rising Edge and Bit 1 Falling Edge Register.

Bit 1 Event Register Enable Field. In order for a bit in the Bit 1 Event Register to set bit 1 in the Status Register true, it must be enabled by the Bit 1 Event Register enable field.

To set the Bit 1 Event Register enable field, send the program command B1EN followed by a number from 0 to 15. The number represents the value of the bits being enabled. (See Figure 3-9.)

Bits in the Bit 1 Event Register are logically ANDed with the corresponding bits in the enable field. If the resultant value is one, bit 1 in the Status Register will be set. If bit 1 in the Status Register enable field is enabled, SRQ will be generated.

Example of Bit 1 Generating an SRQ. The following example will show how Bit 1 registers can be programmed to generate a Service Request.

At turn-on, all registers and enable fields related to the Service Request are set to 0. If the instrument has been turned off for some time the reference oven will probably be cold. Lets say we want to generate a Service Request when the oven is warm.

You may want to read the Bit 1 Status Register to determine the current status of the reference oven. Address the Synthesizer to listen and send the program command B1CST?. Then address the Synthesizer to talk.

If the Synthesizer returns a value of 2, 3, 6, 7, 10, 11, 14, or 15 the oven is cold.

If the oven is cold, program the Bit 1 Falling Edge Register so that the Bit 1 Event Register responds on the true-to-false change to the condition "reference oven cold" in the Bit 1 Status Register. Send the Synthesizer the following command string:

OUTPUT 7[illegible]"B1FAL 2"

Bit 1, Hardware Errors (cont'd)
When the "reference oven cold" condition goes from true to false in the Bit 1 Status Register, bit 2 in the Bit 1 Event Register gets set.

Next, to enable this condition to cause a Service Request:

1. Set the Bit 1 Event Register enable field to allow the "reference oven cold" condition to set bit 1 in the Status Register true. Send the Synthesizer the following command string:

   OUTPUT 719, "B1EN 2"

2. Set the Status Register enable field to allow bit 1 in the Status Register to cause a Service Request. Send the Synthesizer the command string:

   OUTPUT 719; "*SRE 1"

When the reference oven is warm, several things happen:

1. The "reference oven cold" condition in the Bit 1 Status Register goes false.
2. The true-to-false transition in the Bit 1 Status Register sets bit 2 in the Bit 1 Event Register true.
3. Bits in the Bit 1 Event Register are logically ANDed with bits in the Bit 1 Event Register enable field. Because bit 2 (reference oven cold) is both set and enabled, bit 1 in the Status Register is true.
4. The Status Register is logically ANDed with the Status Register enable field. Because bit 1 is both set and enabled, bit 7 in the Status Register is set true, thus sending a Service Request to the controller.

3-43. Bit 2, Operating Conditions

Like bit 1, bit 2 of the Status Register is controlled by four registers and an enable field:

1. Bit 2 Status Register
2. Bit 2 Event Register
3. Bit 2 Rising Edge Register
4. Bit 2 Falling Edge Register
5. Bit 2 Event Register enable field.

The bits in each of the four registers and the enable field are identical. See Figure 3-10.

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value=2	Value=1

Figure 3-10. Bit 2, Operating Conditions

The two conditions capable of driving bit 2 in the Status Register are described in Table 3-8, Bit 2 Status Register.

Table 3-8. Bit 2 Status Register

Bit	Description	Condition That Sets Bit True
2	External Data Transfer Complete	Data transfer through rear panel EXTERNAL DATA INPUT port is completed.
1	Ready for Bus Trigger	The selected packet advance mode is BUS and the sequencer can be advanced via the "ADV-SEQ" program command.

The Bit 2 Status, Event, Rising Edge and Falling Edge registers and the Event Register enable field operate in the same manner as the Bit 1 registers.

To read the Bit 2 Status Register, send the command B2CST?. Address the Synthesizer to talk and read the returned value.

To read the Bit 2 Event Register, send the command B2EVT?. Address the Synthesizer to talk and read the returned value.

To set the Bit 2 Event Register enable field, use the command B2EN followed by a number between 0 and 3.

To program the bits in Bit 2 Event Register to respond to rising or falling edge changes in the Bit 2 Status Register, use the command B2RIS or B2FAL, respectively, followed by a number between 0 and 3.

3-44. Bit 5, Message Available

Bit 5 of the Status Register indicates that a message is ready to be read from the Synthesizer. To

Bit 5, Message Available (cont'd)
generate SRQ bit 5 of the Status Register enable field must be enabled. No other registers are associated with this bit.

3-45. Bit 6, Events

Five conditions are capable of driving bit 6 in the Status Register. These conditions are shown in Figure 3-11.

Event Status Register. The Event Status Register is an 8-bit register that is constantly updated as events occur. Refer to Table 3-9 for a description of conditions that set each bit true.

Bits in the Event Status Register are positive edge sensitive; they are set true on the false to true transition. Once a bit is set, it remains set until the register is read, the instrument is turned off, or the *CLS program command is received.

To read the Event Status Register, address the Synthesizer to listen and send the command *ESR?. Then address the Synthesizer to talk. The Synthesizer returns the current value of the true bits. The register is then set to zero.

Event Status Register Enable Field. The Event Status Register enable field enables bits in the Event Status Register to drive bit 6 in the Status Register.

To enable bits in the Event Status Register enable field, send the Synthesizer the program command *ESE followed by a number between 0 and 255. The number represents the values the bits to be enabled. (See Figure 3-11.)

Bits in the Event Status Register are logically ANDed with bits in the Event Status Register enable field. If the resultant value is equal to one, bit 6 in the Status Register is set true.

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Power On	0	Command Error	Execution Error	0	Query Error	0	Operation Complete
Value= 128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Figure 3-11. Bit 6, Events

Table 3-9. Event Status Register

Bit	Description	Condition That Sets Bit True
8	Power on	Power is applied to Synthesizer.
7	Always 0	
6	Command Error	Synthesizer receives a command with an invalid format. Refer to CMDERR under Programming Commands in this section.
5	Execution Error	Synthesizer receives a parameter in an HP-IB command that is out of range. Refer to EXERR under Programming Commands in this section.
4	Always 0	
3	Query Error	Synthesizer has been addressed to talk without first receiving a query.
2	Always 0	
1	Operation Complete	Synthesizer receives program command "*OPC" or "*OPC?".

Example 3-5, Error Checking Program.
Example 3-5 provides an example of how to generate an SRQ and interrupt the controller when a command or execution error occurs. This program can be included as part of any new program as a debugging aid.

3-46. Bit 7, Require Service (RQS)

Bit 7 is set when Status Register bits 1, 2, 5, and/or 6 are true and have been enabled by the Status Register enable field. When bit 7 is true, a Service Request is generated and the front panel SRQ annunciator lights.

Figure 3-12 summarizes the hierarchy of registers related to programming the Status Register.

Example 3-5. Error Checking Program

```
1   !                ERROR CHECKING PROGRAM
2   !
3   !
4   ! PROGRAM EVENT STATUS REGISTER ENABLE FIELD TO ALLOW COMMAND
5   ! OR EXECUTION ERRORS TO SET THE EVENT BIT IN THE STATUS REGISTER.
6   ! NEXT, PROGRAM THE STATUS REGISTER ENABLE FIELD TO GENERATE AN
7   ! SRQ WHENEVER THE EVENT BIT IS TRUE. AN SRQ WILL BE GENERATED
8   ! WHENEVER EITHER ERROR IS GENERATED BY THE SYNTHESIZER.
9   ! THIS CODE CAN BE INCLUDED AT THE BEGINNING OF A NEW PROGRAM AS A
10  ! DEBUGGING AID.
11    COM /Aws_add/ INTEGER Hpib
12    INTEGER A
13    Hpib=719                          ! AWS HPIB ADDRESS
14    ASSIGN @Aws TO Hpib               !
15                                      !
16    OUTPUT @Aws;"*ESE 48"             ! ENABLE AWS STATUS REGISTERS
17    OUTPUT @Aws;"*SRE 32"
18    OUTPUT @Aws;"*ESR?"               ! CLEAR ALL BITS IN STATUS REGISTER
19                                      ! BITS ARE ONLY CLEARED BY *ESR? OR *CLS
20                                      !
21                                      !
22                                      ! SET UP FOR SRQ INTERRUPT
23    DISABLE                           ! ALL INTERRUPTS OFF
24    ON INTR 7 CALL Service            ! WHEN SRQ IS TRUE, SERVICE THE INTERRUPT
25    Mask=2
26    Sp=SPOLL(@Aws)                    ! CLEAR ANY PENDING SRQ
27    ENABLE INTR 7;Mask                ! HPIB INTERRUPT ONLY. SRQ BIT ONLY
28    ENABLE                            ! ENABLE ALL INTERRUPTS
29                                      !
30                                      !
31                                      !
32  !                YOUR PROGRAM CAN BE INSERTED HERE
33                                      !
34                                      !
48                                      !
49  END                                 ! END OF MAIN PROGRAM
50  Service:SUB Service     ! ******** INTERRUPT HANDLING ROUTINE *********
51    ! ONLY A COMMAND ERROR OR AN EXECUTION ERROR FROM SYNTHESIZER WILL CAUSE
52    ! THE SERVICE ROUTINE TO BE CALLED. IF MORE THAN THE HP 8770 IS ON THE BUS
53    ! THEN A PARALLEL POLL WOULD BE NEEDED  TO ESTABLISH WHICH DEVICE CAUSED
54    ! THE SRQ LINE TO GO ACTIVE.
55    !
56      COM /Aws_add/ INTEGER Hpib
57      ASSIGN @Aws TO Hpib             ! AWS HPIB ADDRESS
58      Sp=SPOLL(@Aws)                  ! DO SERIAL POLL. CLEAR SRQ LINE
59      OUTPUT @Aws;"*ESR?"             ! EVENT STATUS REGISTER. NEEDED TO CLEAR
60                                      ! BITS 5,6
61      OUTPUT @Aws;"EXERR?"            ! GET ERROR CODE FROM EXEC. ERROR ROUTINE
62      ENTER @Aws;A                    !
63      PRINT "EXECUTION ERROR OCCURRED. ERROR #",A
64      OUTPUT @Aws;"CMDERR?"           ! GET ERROR CODE FROM COMMAND ERROR ROUTINE
65      ENTER @Aws;A
66      PRINT "COMMAND ERROR OCCURRED. ERROR #",A
67      PRINT
68      ENABLE INTR 7;Mask              !READY FOR NEXT ERROR
69      BEEP 200,.1
70      BEEP 400,.1                     ! BEEP TO INDICATE THAT AN ERROR OCCURRED
71      WAIT .3
72    SUBEND
```

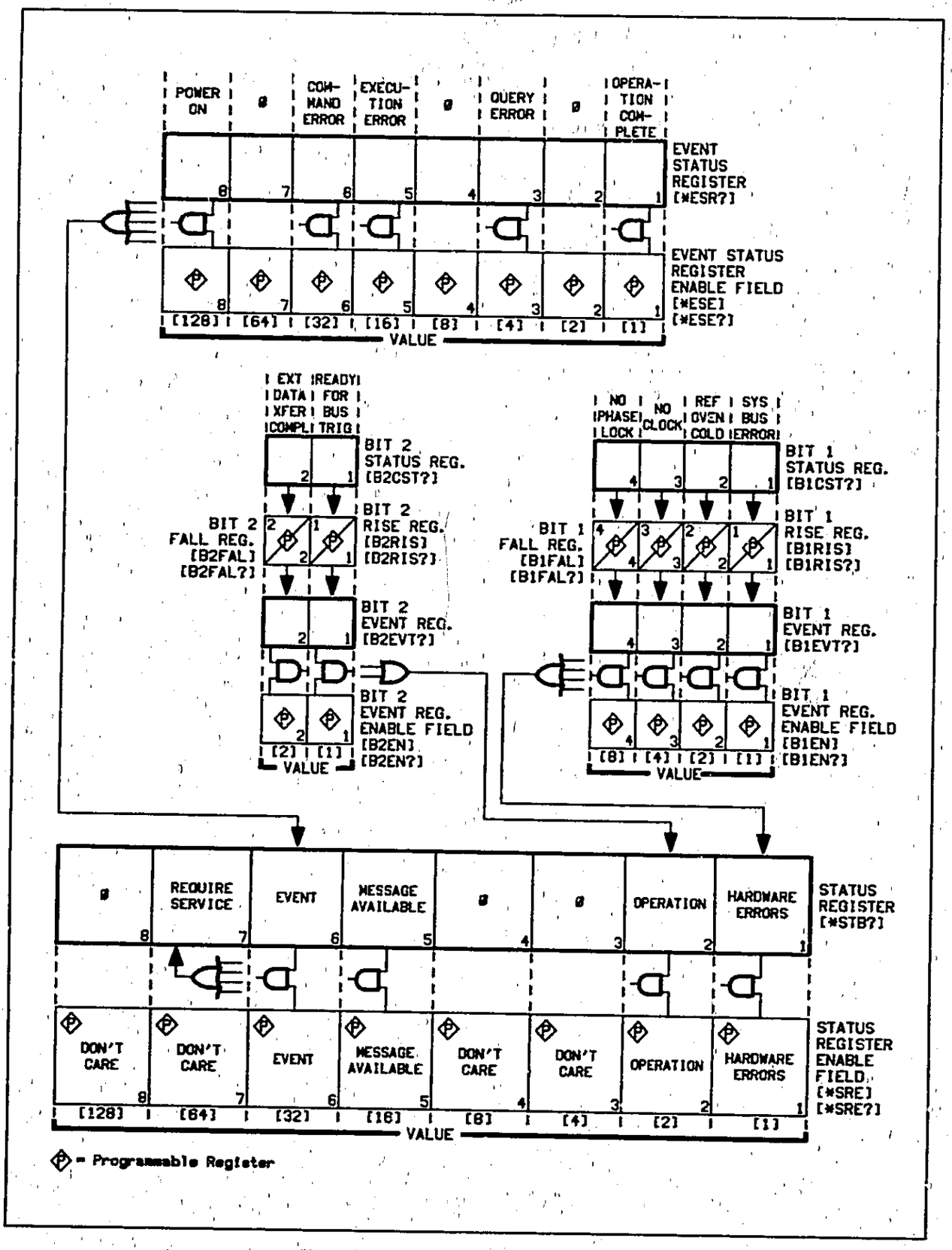


Figure 3-12. Status Register Hierarchy Chart

3-47. Parallel Poll Enable Field

The Synthesizer can be programmed to respond with one bit of status when parallel polled. The Parallel Poll enable field determines what the one bit of status represents. In order for the Synthesizer to respond to a parallel poll, it must be configured to to so by the controller. The controller assigns the data line on which the Synthesizer responds (DIO1—DIO8) and the logic sense of the bit (negative-true or positive-true logic).

Bits in the Parallel Poll enable field are identical to the bits in the Status Register. The enable field is set by sending the Synthesizer the command *PER followed by the value of the bits to be enabled.

The Parallel Poll enable field is logically ANDed with its corresponding bits in the Status Register. How the resultant value is interpreted depends on the previously programmed logic sense of the status bit.

Table 3-10. Summary of Status Register Commands (1 of 2)

Command	Description
B1CST?	Reads the Bit 1 Status Register.
B1EN	Sets the Bit 1 Event Register enable field. Enables conditions to drive bit 1 of the Status Register.
B1EN?	Reads the current setting of the Bit 1 Event Register enable field.
B1EVT?	Reads the Bit 1 Event Register, then sets register to zero.
B1FAL	Sets the Bit 1 Falling Edge Register.
B1FAL?	Reads the current setting of the Bit 1 Falling Edge Register.
B1RIS	Sets Bit 1 Rising Edge Register.
B1RIS?	Reads the current setting of the Bit 1 Rising Edge Register.
B2CST?	Reads Bit 2 Status Register.
B2EN	Sets the Bit 2 Event Register enable field. Enables conditions to drive bit 2 in the Status Register.
B2EN?	Reads the current setting of the Bit 2 Event Register enable field.
B2EVT?	Reads the Bit 2 Event Register, then sets the register to zero.
B2FAL	Sets the Bit 2 Falling Edge Register.
B2FAL?	Reads the current setting of the Bit 2 Falling Edge Register.
B2RIS	Sets the Bit 2 Rising Edge Register.
B2RIS?	Reads the current setting of the Bit 2 Rising Edge Register.
*CLS	Clears the Bit 1 Event Register, Bit 2 Event Register, the Event Status Register and the Status Register.
*ESE	Sets the Event Status Register enable field.
*ESE?	Reads the current setting of the Event Status Register enable field.

Table 3-10. Summary of Status Register Commands (2 of 2)

Command	Description
*ESR?	Reads the Event Status Register, then sets the register to zero.
*OPC	Sets bit 1 in the Event Status Register true.
*OPC?	Always returns a "1" and sets bit 1 in the Event Status Register true.
*PER	Sets Parallel P[illegible] field.
*PER?	Reads the current setting of the Parallel Poll enable field.
*SRE	Sets the Status Register enable field.
*SRE?	Reads the current setting of the Status Register enable field.
*STB?	Reads the Status Register.

3-48. PROGRAMMING COMMANDS

All the commands in this section are executed immediately.

Command syntax is represented pictorially. Refer to the ATTEN command as an example. All characters enclosed by a rounded envelope must be entered exactly as shown. In the diagrams, narrow ovals surround command names. Circles and wide ovals surround secondary keywords, or special numbers and characters. Secondary keywords are described in Table 3-11.

Words enclosed by a rectangular box are names of items also used in the command statement. These items are described in a table below the syntax diagram for each command.

Each line can be followed in only one direction, as indicated by the arrow at the end of the line. Any combination of statement elements that can be generated by following the lines in the proper direction is syntactically correct. An element is optional if there is a path around it.

All examples are written in an enhanced BASIC language using the HP 9000 Series 200 Model 236 Computer. The examples assume the select code of the HP-IB interface is 7 and the Synthesizer's address is 19.

Table 3-12 lists all Synthesizer programming commands alphabetically. Commands used only for diagnostics are described in Section VIII of the Operating and Service Manual. Table 3-13 groups the commands by function.

Table 3-11. Secondary Keywords Enclosed in Circles

Command	Description
A	Designates A-block format
ASCII	ASCII
AUTO	Automatic packet advance mode
B	Designates B-block format
BIN	Binary
BOTH	Waveform and sequencer memory
BUS	HP-IB packet advance mode
C	Designates C-block format
END	Terminates command statement by asserting EOI with last byte
EXT	External trigger packet advance mode
HIGH	Active edge of pulse is low-to-high state change
I	Designates I-block format
L	Designates L-block format
LOW	Active edge of pulse is high-to-low state change
OFF	Turns function off
ON	Turns function on
SEQ	Sequencer memory
SIGN	Signed format for data transfers to waveform memory
SP	Space
UNSIGN	Unsigned format for data transfers to waveform memory
WAVE	Waveform memory
#	Initiates block data field
?	Returns a query response containing the value or state of the associated parameter
;	Terminates command statement

Table 3-12. Summary of Program Commands (1 of 4)

Command	Description
ADVSEQ	Advances the sequencer from the current packet to the next packet in the sequence. The sequencer must be running and the packet advance mode must be BUS.
ADVSEQ?	Returns a 1 if the sequencer is running and the current packet advance mode is BUS. Otherwise, returns a 0.
ATTEN	Sets output attenuator to a value in dB.
ATTEN?	Returns current attenuator setting in dB.
BER?	Returns bus error number. Refer to Section VIII, Service.
BLKMOVE	Moves a block of waveform memory data.
B1CST?	Returns the current value of the Bit 1 Status Register.
B1EN	Sets the enable field for the Bit 1 Event Register.
B1EN?	Returns the current setting of the enable field for the Bit 1 Event Register.
B1EVT?	Returns the current value of the Bit 1 Event Register and sets the register to zero.
B1FAL	Programs the Bit 1 Falling Edge Register. Selected bits in the Bit 1 Event Register are set to one on the one-to-zero transition of changes in the Bit 1 Status Register.
B1FAL?	Returns the current setting of the Bit 1 Falling Edge Register.
B1RIS	Programs the Bit 1 Rising Edge Register. Selected bits in the Bit 1 Event Register are set to one on the zero-to-one transition of changes in the Bit 1 Status Register.
B1RIS?	Returns the current setting of the Bit 1 Rising Edge Register.
B2CST?	Returns the current value of the Bit 2 Status Register.
B2EN	Sets the enable field for the Bit 2 Event Register.
B2EN?	Returns the current setting of the enable field for the Bit 2 Event Register.
B2EVT?	Returns the current value of the Bit 2 Event Register and sets the register to zero.
B2FAL	Programs the Bit 2 Falling Edge Register. Selected bits in the Bit 2 Event Register are set to one on the one-to-zero transition of changes in the Bit 2 Status Register.
B2FAL?	Returns the current setting of the Bit 2 Falling Edge Register.
B2RIS	Programs the Bit 2 Rising Edge Register. Selected bits in the Bit 2 Event Register are set to one on the zero-to-one transition of changes in the Bit 2 Status Register.
B2RIS?	Returns the current setting of the Bit 2 Rising Edge Register.
CLKDIV	Sets the sampling clock divider rate.

Table 3-12. Summary of Program Commands (2 of 4)

Command	Description
CLKDIV?	Returns the current sampling clock divider rate.
CLKSEL	Selects the sampling clock source. Can be either internal or external.
CLKSEL?	Returns INT if internal clock source is selected and EXT if external clock source is selected.
*CLS	Clears all event registers that drive the Status Register and clears the Status Register.
CMDERR?	Returns the error code of the first command error that occurred since the last command error query.
CONST	Fills a waveform memory file with a constant value.
DIR? SEQ	Returns the directory of all packets in the current sequence.
DIR? WAVE	Returns a directory of all named files loaded in waveform memory.
EDAR	Programs the edge sense of the External Data Input port EDAR handshake line.
EDAR?	Returns the currently programmed edge sense of the EDAR handshake line.
EDV	Programs the edge sense of the External Data Input port EDV handshake line.
EDV?	Returns the currently programmed edge sense of the EDV handshake line.
*ESE	Enters the enable field for the Event Status Register.
*ESE?	Returns the current setting of the enable field for the Event Status Register.
*ESR?	Returns the current value of the Event Status Register and sets the register to zero.
EXDABT	Aborts any pending transfers to the External Data Input port.
EXDAT?	Returns the number of words remaining to be transferred through the External Data Input port.
EXDSEQ	Prepares the Synthesizer to receive data through the External Data Input port and to place that data in sequencer memory.
EXDWAV	Prepares the Synthesizer to receive data through the External Data Input port and to place that data in a named file in waveform memory.
EXDWM	Prepares the Synthesizer to receive data through the External Data Input port and to place that data in a specific location in waveform memory.
EXERR?	Returns the error code of the first execution error reported since the last execution error query.
FORMAT	Sets the numbering system mode for waveform memory data transfers. Can be either SIGN or UNSIGN.

Table 3-12. Summary of Program Commands (3 of 4)

Command	Description
FORMAT?	Returns either SIGN or UNSIGN to indicate the current number system mode.
GEN	Causes the Synthesizer to output a 12.5 MHz sine wave.
GO	Starts the sequencer running at the first packet in sequencer memory.
*IDN?	Returns a string identifying the Synthesizer's model number and firmware version.
*LRN?	Returns a string containing eight Synthesizer hardware settings.
LRNDIR	Causes Synthesizer to accept a block of data containing the waveform memory file directory.
LRNDIR?	Returns a message in the form of a #A data block that contains the current waveform memory file directory of the Synthesizer.
MARKADD	Sets address marker to a specific address in waveform memory. Used in conjunction with EQUAL ADDRESS rear panel connector.
MARKADD?	Returns memory address of marker setting.
MARKER	Sets address marker to the starting address plus offset of a named file in waveform memory. Used in conjunction with EQUAL ADDRESS rear panel connector.
MUXSEL	Selectively enables waveform memory channels for diagnostics. Refer to Section VIII, Service.
MUXSEL?	Returns currently enabled waveform memory channels. Refer to Section VIII, Service.
*OPC	Sets bit 1 (operation complete) in Event Status Register true.
*OPC?	Returns a "1" and sets bit 1 (operation complete) of the Event Status Register true.
OUTPUT	Turns RF output ON or OFF.
OUTPUT?	Returns either ON or OFF to describe the current state of the RF output.
PACKET	Defines one or more packets for the current sequence. Specifies waveform memory file.
PACLIT	Defines one or more packets for the current sequence. Specifies waveform memory address.
PACLIT?	Returns the packet parameters for the existing sequence.
PEEK	Reads hardware data value at specified address. Refer to Section VIII, Service.
*PER	Sets parallel poll enable field.

Table 3-12. Summary of Program Commands (4 of 4)

Command	Description
*PER?	Returns current setting of parallel poll enable field.
POKE	Writes data to hardware at specified address. Refer to Section VIII, Service.
PURGE	Purges entire sequence memory, entire waveform and sequencer memory, or a named waveform memory file.
READSA?	Reads DSA signatures. Refer to Section VIII, Service.
*RST	Resets Synthesizer to default parameters.
RSTDSA	Resets DSA accumulator. Refer to Section VIII, Service.
SCALE	Scales a named file.
SEQUEN	Loads sequence memory with data.
SEQUEN?	Returns data string currently loaded in sequencer memory.
SINPQ	Computes a 12-bit sine wave.
*SRE	Sets enable field for the Status Register.
*SRE?	Returns current setting of Status Register enable field.
*STB?	Returns the value of the Status Register.
STOP	Stops sequencer at current packet.
TST?	Performs specified test. Returns 0 if test passes and 1 if test fails. Refer to Section VIII, Service.
*TST?	Performs Synthesizer's power-up test. Refer to Section VIII, Service.
TSTQRY?	Returns any pending test results. Refer to Section VIII, Service.
WAVE	Loads waveform memory with data. The data is placed in a named file.
WAVE?	Returns the data in waveform memory.
WAVEAP	Appends data to an existing named file in waveform memory.
WMEM	Loads waveform memory with data. This data is placed in an unnamed record at a specified address.
WMEM?	Returns data in waveform memory from a starting location for a specified number of words.

Table 3-13. HP-IB Commands in Functional Groupings

Commands Using Addresses
BLKMOVE
EXDWM
MARKADD
PACLIT
WMEM

Commands Using Named Files
CONST
DIR? WAVE
EXDWAV
MARKER
PACKET
PURGE
SCALE
SINPQ
WAVE
WAVEAP

Errors
CMDERR
EXERR

External Data Input
EDAR
EDV
EXDABT
EXDAT
EXDSEQ
EXDWAV
EXDWM

Hardware Control
ATTEN
CLKDIV
CLKSEL
GEN
*LRN
*RST

IEEE Std 728 Common Commands
*CLS
*ESE
*ESR
*IDN
*LRN
*OPC
*PER
*RST
*SRE
*STB

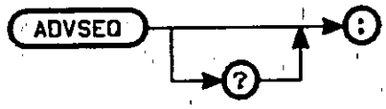
Memory Management
BLKMOVE
DIR? SEQ
DIR? WAVE
*LRN
LRNDIR

Sequencer
ADVSEQ
DIR? SEQ
EXDSEQ
GEN
GO
OUTPUT
PACKET
PACLIT
PURGE
*RST
SEQUEN
STOP

Status Register
B1CST
B1EN
B1EVT
B1FAL
B1RIS
B2CST
B2EN
B2EVT
B2FAL
B2RIS
*CLS
*ESE
*ESR
*OPC
*PER
*SRE
*STB

Waveform Memory
CONST
DIR? WAVE
EXDWAV
EXDWM
FORMAT
GEN
MARKADD
MARKER
PURGE
*RST
SCALE
SINPQ
WAVE
WAVEAP
WMEM

ADVSEQ
(Advance Sequencer)



Description

The ADVSEQ command advances the sequencer from the current packet to the next packet in a sequence. This command is valid only when the sequencer is running and the current packet advance mode is defined as BUS. The packet advance mode is defined in the PACKET or PACLIT command.

When queried (?), ADVSEQ returns the current status of the sequencer. The value 1 is returned if the sequencer is running and the current packet advance mode is BUS. The value 0 is returned if the sequencer is off or the current packet advance mode is not BUS.

Example

To use the ADVSEQ command to advance to the next packet, first enter the following short program.

```
10 OUTPUT 719; "PURGE BOTH"
20 OUTPUT 719; "SINPQ FREQ1, 1, 10000"
30 OUTPUT 719; "SINPQ FREQ2, 4, 10000"
40 OUTPUT 719; "PACKET FREQ1, 0, BUS, FREQ2, 0, BUS"
50 OUTPUT 719; "GO"
60 END
```

Line 10: Purges both waveform and sequencer memory.
Line 20: Creates a wave segment named FREQ1 and stores it in waveform memory. FREQ1 is a sine wave with 1 cycle of 10 000 elements.
Line 30: Creates a wave segment named FREQ2 and stores it in waveform memory.
Line 40: Creates a sequence using 2 packets. In packet 1, FREQ1 is the file name, 0 is the number of scans (FREQ1 is scanned indefinitely until "ADVSEQ" is received) and BUS is the packet advance mode. In packet 2, FREQ2 is the file name, 0 is the number of scans and BUS is the packet advance mode.
Line 50: Starts sequencer running at first packet in sequence (packet 1).

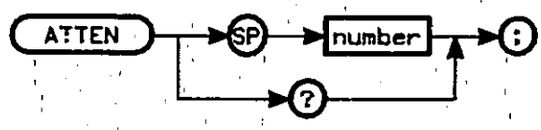
Run the program. To advance to the second packet in the sequence, send the Synthesizer the following:

```
OUTPUT 719; "ADVSEQ"
```

Related Sections

PACKET
PACLIT

ATTEN
(Attenuation)



Item	Description	Range/Restrictions
Number	Represents attenuation in dB	0 through 110 in multiples of 10

Description

The ATTEN command sets the internal RF output attenuator. The attenuator can lower the output level in 10 dB steps from 0 to 110 dB. Entered values that are not multiples of 10 are rounded to the nearest multiple of 10.

When queried (?), ATTEN returns the current attenuator setting in dB.

Example

To set the attenuator to 10 dB:

```
OUTPUT 719;"ATTEN 10"
```

To determine the current attenuator setting:

```
10 OUTPUT 719;"ATTEN?"
20 ENTER 719;A
30 PRINT "ATTEN =";A
40 END
```

Comments

The attenuator setting can be changed anytime, even while the sequencer is running.

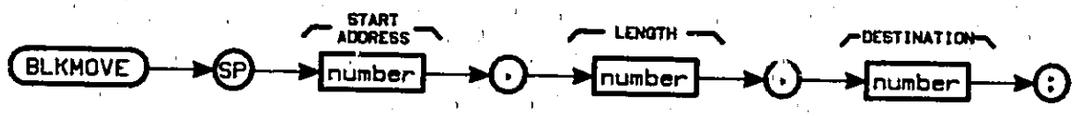
The RF output has 12 full scale amplitude ranges depending on the attenuator setting.

Attenuator Setting	Full Scale Peak-to-Peak Output
0 dB	2V
10 dB	630 mV
20 dB	200 mV
30 dB	63 mV
40 dB	20 mV
50 dB	6.3 mV
60 dB	2 mV
70 dB	630 μV
80 dB	200 μV
90 dB	63 μV
100 dB	20 μV
110 dB	6.3 μV

Related Sections

OUTPUT

BLKMOVE
(Block Move)



Item	Description	Range/Restrictions
Number	Represents starting address in waveform memory of block to be moved.	0 to 131 071
Number	Represents length of block to be moved.	
Number	Represents the starting address of the destination of move.	0 to 131 071

Description The BLKMOVE command moves a block of waveform memory data. This command is useful for compressing data to obtain free space.

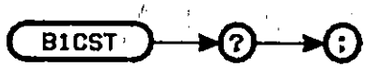
Example To move 1024 elements of waveform memory, starting from address 64, to another location starting with address 4088:

```
OUTPUT 719;"BLKMOVE 64, 1024, 4088"
```

Comments This command should be used only when working directly with waveform memory address locations instead of file names.

Related Sections GO
WMEM

B1CST
(Bit 1 Current Status)



Description B1CST? returns the current value of the Bit 1 Status Register. The Synthesizer returns an integer between 0 and 15 that represents the value of the true bits in the register. The Bit 1 Status Register and the value of each bit, when true, is shown below.

Bit 1 (Hardware Errors) Status Register

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Lock Loop Out of Lock	No System Clock	Reference Oven Cold	Bus Error
Value=8	Value=4	Value=2	Value=1

Example To read the Bit 1 Status Register:

```
10 OUTPUT 719;"B1CST?"
20 ENTER 719;A
30 PRINT "Bit 1 Status Register = ";A
40 END
```

Comments The Bit 1 Status Register is a read-only register. Bits in the register are set as conditions occur and are cleared when the condition is no longer true.

Related Sections

B1EN

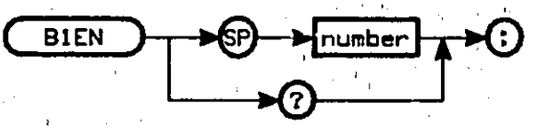
B1EVT

B1FAL

B1RIS

Programming the Status Register

B1EN
(Bit 1 Event Register Enable)



Item	Description	Range/Restrictions
Number	Represents the value of bits in Bit 1 Event Register being enabled.	0 to 15

Description

The B1EN command sets the enable field for the Bit 1 Event Register. Bits in Bit 1 Event Register are logically ANDed with the corresponding bits in the enable field. If the result is one, bit 1 in the Status Register is set.

When queried (?), B1EN returns the current setting of the Bit 1 Event Register enable field.

The value of each bit in the Bit 1 Event Register enable field is shown below.

Bit 1 Event Register Enable Field

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Lock Loop Out of Lock	No System Clock	Reference Oven Cold	Bus Error
Value=8	Value=4	Value=2	Value=1

Example

To enable the condition "reference oven cold" to generate a Service Request when the oven goes from cold to warm:

```
10 OUTPUT 719;"B1FAL 2"
20 OUTPUT 719;"B1EN 2"
30 OUTPUT 719;"*SRE 1"
40 END
```

Line 10: Sets Bit 1 Falling Edge Register
Line 20: Sets Bit 1 Event Register enable field
Line 30: Sets Status Register enable field

To combine lines 10—30 into one command statement:

```
OUTPUT 719; "B1FAL 2; B1EN 2; *SRE 1"
```

B1EN (cont'd)
(Bit 1 Event Register Enable)

Comments

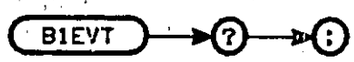
The Bit 1 Event Register enable field is set to zero at power-up.

Related Sections

B1CST
B1EVT
B1FAL
B1RIS
Programming the Status Register
*SRE
*STB

B1EVT

(Read Bit 1 Event Register)



Description B1EVT? returns an integer from 0 to 15, representing the value of bits that are true in the Bit 1 Event Register. Bits in the Bit 1 Event Register, when enabled, drive bit 1 in the Status Register.

Bits in the Bit 1 Event Register are set when two conditions occur:

1. A bit in the Bit 1 Status Register changes states (from true-to-false or from false-to-true)
2. The Bit 1 Rising Edge or Falling Edge Register was previously programmed so that the corresponding bit in the Bit 1 Event Register responds to the transition.

Once a bit has been set, it remains set until the register is read. The register is set to 0 after it has been read.

The Bit 1 Event Register and the value of each bit is shown below.

Bit 1 Event Register

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Lock Loop Out of Lock	No System Clock	Reference Oven Cold	Bus Error
Value=8	Value=4	Value=2	Value=1

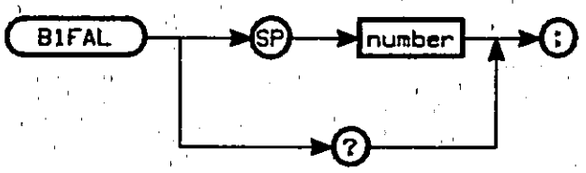
Example To read the current setting of the Bit 1 Event Register:

```
10 OUTPUT 719; "B1EVT?"
20 ENTER 719; A
30 PRINT "Bit 1 Event Register = ";A
40 END
```

Comments The Bit 1 Event Register is set to zero at power-up. Programming commands *CLS and *RST also set the register to zero.

Related Sections
B1CST
B1EN
B1FAL
B1RIS
*CLS
Programming the Status Register
*RST
*SRE
*STB

B1FAL
(Bit 1 Falling Edge Register)



Item	Description	Range/Restrictions
Number	Represents the value of bits in the Bit 1 Event Register that will be set on true-to-false changes in the Bit 1 Status Register.	0 to 15

Description

The B1FAL command programs the Bit 1 Falling Edge Register. Bits in the Bit 1 Event Register are set by changes in the corresponding bit in the Bit 1 Status Register. The Bit 1 Falling Edge Register programs bits in the Bit 1 Event Register to be set on the true-to-false changes in the Bit 1 Status Register.

When queried (?), B1FAL returns the current setting of the register.

Bit 1 Falling Edge Register

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Lock Loop Out of Lock	No System Clock	Reference Oven Cold	Bus Error
Value=8	Value=4	Value=2	Value=1

Example

To program the Status Register to generate a service request (SRQ) when the reference oven goes from cold to warm:

```
10 OUTPUT 719;"B1FAL 2"
20 OUTPUT 719;"B1EN 2"
30 OUTPUT 719;"*SRE 1"
40 END
```

Line 10: Sets Bit 1 Falling Edge Register
Line 20: Sets Bit 1 Event Register enable field
Line 30: Sets Status Register enable field

To combine lines 10—30 into one command statement:

```
OUTPUT 719; "B1FAL 2; B1EN 2; *SRE 1"
```

B1FAL (cont'd)
(Bit 1 Falling Edge Register)

Comments

The Bit 1 Falling Edge Register is set to zero at power-up.

Related Sections

B1CST

B1EN

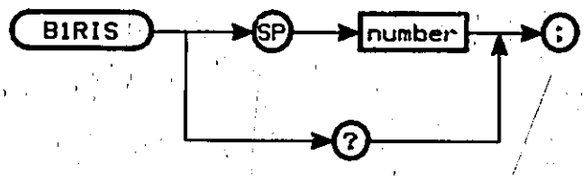
B1EVT

B1RIS

Programming the Status Register

*SRE

B1RIS
(Bit 1 Rising Edge Register)



Item	Description	Range/Restrictions
Number	Represents value of bits in Bit 1 Event Register that will be set on false-to-true changes in the Bit 1 Status Register	0 to 15

Description

The B1RIS command programs the Bit 1 Rising Edge Register. Bits in the Bit 1 Event Register are set by changes of the corresponding bit in the Bit 1 Status Register. The Bit 1 Rising Edge Register programs bits in the Bit 1 Event Register to be true on the false-to-true changes in the Bit 1 Status Register.

When queried (?), B1RIS returns the current setting of the register.

Bit 1 Rising Edge Register

Bit 4	Bit 3	Bit 2	Bit 1
Phase-Lock Loop Out of Lock	No System Clock	Reference Oven Cold	Bus Error
Value=8	Value=4	Value=2	Value=1

Example

To program the Status Register to generate a service request (SRQ) when the phase-locked loop goes out of lock:

```
10 OUTPUT 719;"B1RIS 8"
20 OUTPUT 719;"B1EN 8"
30 OUTPUT 719;"*SRE 1"
40 END
```

Line 10: Sets Bit 1 Rising Edge Register
Line 20: Sets Bit 1 Event Register enable field
Line 30: Sets Status Register enable field

To combine lines 10—30 into one command statement:

```
OUTPUT 719; "B1RIS 8; B1EN 8; *SRE 1"
```

B1RIS (cont'd)
(Bit 1 Rising Edge Register)

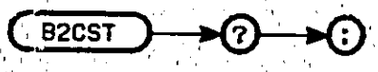
Comments

The Bit 1 Rising Edge Register is set to zero at power-up.

Related Sections

B1CST
B1EN
B1EVT
B1FAL
Programming the Status Register
*SRE

B2CST
(Bit 2 Current Status)



Description B2CST? returns the current value of the Bit 2 Status Register. The Synthesizer returns an integer between 0 and 3, which represents the value of bits in the register that are true. The Bit 2 Status Register and the value of each bit, when true, is shown below.

Bit 2 (Operating Conditions) Status Register

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value = 2	Value = 1

Example To read the Bit 2 Status Register:

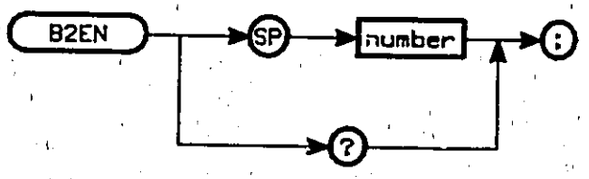
```
10 OUTPUT 719;"B2CST?"
20 ENTER 719;A
30 PRINT "Bit 2 Status Register =";A
40 END
```

Comments The Bit 2 Status Register is a read-only register. Bits are set while conditions are true and are cleared when the condition is no longer true.

Related Sections
B2EN
B2EVT
B2FAL
B2RIS
Programming the Status Register

B2EN

(Bit 2 Event Register Enable)



Item	Description	Range/Restrictions
Number	Represents the value of bits in the Bit 2 Event Register being enabled.	0 to 3

Description

The B2EN command sets the enable field for the Bit 2 Event Register.

Bits in the Bit 2 Event Register are logically ANDed with the corresponding bits in the enable field. If the result is one, bit 2 in the Status Register is set.

When queried (?), B2EN returns the current setting of the Bit 2 Event Register enable field.

The decimal value of each bit in the Bit 2 Event Register enable field is shown below.

Bit 2 Event Register Enable Field

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value = 2	Value = 1

Example

To enable the condition "External Data Transfer Complete" to generate a Service Request (SRQ) when the data transfer is completed:

```
10 OUTPUT 719;"B2RIS 2"
20 OUTPUT 719;"B2EN 2"
30 OUTPUT 719;"*SRE 2"
40 END
```

Line 10: Sets the Bit 2 Rising Edge Register.
Line 20: Sets the Bit 2 Event Register enable field.
Line 30: Sets the Status Register enable field.

To combine lines 10–30 into one program statement:

```
OUTPUT 719;"B2RIS 2; B2EN 2; *SRE 2"
```

Comments

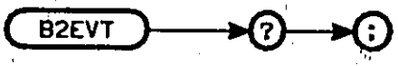
The Bit 2 Event Register enable field is set to zero at power-up.

Related Sections

B2CST	B2RIS
B2EVT	Programming the Status Register
B2FAL	*SRE

B2EVT

(Read Bit 2 Event Register)



Description

B2EVT? returns an integer from 0 to 3, representing the value of bits that are true in the Bit 2 Event Register. Bits in the Bit 2 Event Register, when enabled by the Bit 2 Event Register enable field, drive bit 2 in the Status Register.

Bits in the Bit 2 Event Register are set when two conditions occur:

1. A bit in the Bit 2 Status Register changes states (from true-to-false or from false-to-true)
2. The Bit 2 Rising Edge or Falling Edge Register was previously programmed so that the corresponding bit in the Bit 2 Event Register responds to the transition.

Once a bit has been set, it remains set until the register is read. The register is set to 0 after it has been read.

Bit 2 Event Register

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value = 2	Value = 1

Example

To read the current value of the Bit 2 Event Register:

```
10 OUTPUT 719; "B2EVT?"
20 ENTER 719; A
30 PRINT "BIT 2 EVENT REGISTER =": A
40 END
```

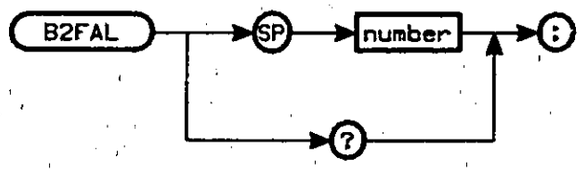
Comments

The Bit 2 Event Register is set to zero at power-up. Programming commands *CLS and *RST also set the register to zero.

Related Sections

B2CST
B2EN
B2FAL
B2RIS
*CLS
Programming the Status Register
*RST
*SRE
*STB

B2FAL
(Bit 2 Falling Edge Register)



Item	Description	Range/Restrictions
Number	Represents value of bits in Bit 2 Event Register that will be set on true-to-false changes in Bit 2 Status Register.	0 to 3

Description

The B2FAL command programs the Bit 2 Falling Edge Register. Bits in the Bit 2 Event Register are set by changes of the corresponding bit in the Bit 2 Status Register. The Bit 2 Falling Edge Register programs bits in the Bit 2 Event Register to be set true on true-to-false (falling edge) changes in the Bit 2 Status Register.

When queried (?), B2FAL returns the current setting of the register, an ASCII decimal integer between 0 and 3.

Bit 2 Falling Edge Register

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value = 2	Value = 1

Example

To program the Bit 2 Falling Edge Register so that bit 1 in the Bit 2 Event Register will be set true on the true-to-false change in the Bit 2 Status Register:

```
OUTPUT 719; "B2FAL 1"
```

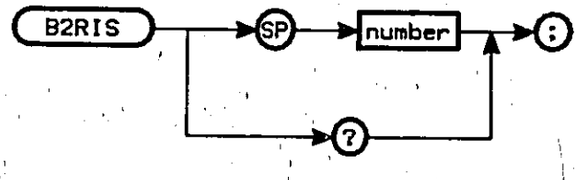
Comments

The Bit 2 Falling Edge Register is set to zero at power-up.

Related Sections

B2CST
B2EN
B2EVT
B2RIS
Programming the Status Register
*SRE

B2RIS
(Bit 2 Rising Edge Register)



Item	Description	Range/Restrictions
Number	Represents the value of bits in the Bit 2 Event Register that will be set on false-to-true changes in the Bit 2 Status Register.	0 to 3

Description

The B2RIS command programs the Bit 2 Rising Register. Bits in the Bit 2 Event Register are set by changes of the corresponding bit in the Bit 2 Status Register. The Bit 2 Rising Edge Register programs bits in the Bit 2 Event Register to be true on the false-to-true (rising edge) changes in the Bit 2 Status Register.

When queried (?), B2RIS returns the current setting of the register, an ASCII decimal integer between 0 and 3.

Bit 2 Rising Edge Register

Bit 2	Bit 1
External Data Transfer Complete	Ready for Bus Trigger
Value = 2	Value = 1

Example

To enable the condition "External Data Transfer complete" to generate a Service Request (SRQ) when the data transfer is completed:

```
10 OUTPUT 719; "B2RIS 2"
20 OUTPUT 719; "B2EN 2"
30 OUTPUT 719; "*SRE 2"
40 END
```

Line 10: Sets the Bit 2 Rising Edge Register.
Line 20: Sets the Bit 2 Event Register enable field.
Line 30: Sets the Status Register enable field.

To combine lines 10—30 into one program statement:

```
OUTPUT 719; "B2RIS 2; B2EN 2; *SRE 2"
```

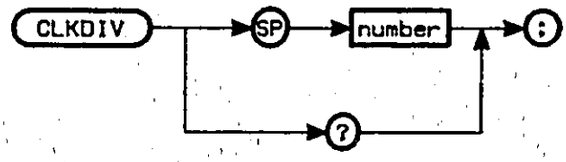
Comments

The Bit 2 Rising Edge Register is set to zero at power-up.

Related Sections

B2CST	Programming the Status Register
B2EN	*SRE
B2EVT	
B2FAL	

CLKDIV
(Clock Divide)



Item	Description	Range/Restrictions
Number	Value by which sampling clock is divided	1, 2, 4, 8, 16, 32, 64, 128, 256

Description

The CLKDIV command sets the sampling clock divider rate. The main sampling clock can be divided by any of the values listed in the above table. The divided sampling clock clocks all system logic.

When queried (?), CLKDIV returns the current value of the clock divider rate.

The CLKDIV command allows waveform memory to be scanned more slowly to give a longer memory capacity.

Example

To divide the sampling clock by 8:

```
OUTPUT 719;"CLKDIV 8"
```

Comments

The output from the CLK/8 OUTPUT connector is divided by the same rate as the sampling clock.

The following table shows how dividing the internal clock affects scan times.

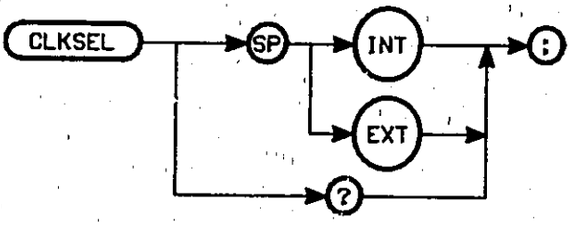
Divisor	Sampling Rate	Scan Time
1	125 MHz	8 ns
2	62.5 MHz	16 ns
4	31.25 MHz	32 ns
8	15.625 MHz	64 ns
16	7.812 MHz	128 ns
32	3.902 MHz	256 ns
64	1.953 MHz	.512 μs
128	976.562 kHz	1.024 μs
256	488.281 kHz	2.048 μs

The Synthesizer has an internal low pass filter in the output path to eliminate the sampling energy due to the sampling clock. This low pass filter has a cutoff frequency of 50 MHz and is optimized for use with the 125 MHz sampling clock. If the clock divider is used, sampling energy will not be eliminated on the output. An external low pass filter may be needed to eliminate the undesired energy near the clock frequency and its harmonics.

Related Sections

CLKSEL
GO

CLKSEL
(Clock Selection)



Description

The CLKSEL command selects the sampling clock source. Either internal (INT) or external (EXT) mode can be selected. Internal mode selects the internal 125 MHz phase-locked loop source. External mode selects a user supplied external source in the 60 to 130 MHz range.

When queried (?), CLKSEL returns the current selected clock source, either INT or EXT.

For internal mode, the internal 125 MHz source can be phase locked to either an internal or external 10 MHz reference. To use the internal reference, the jumper connecting the rear panel 10 MHz REFERENCE INPUT connector to the 10 MHz REFERENCE OUTPUT 2 connector must be installed. To use an external reference, connect a 10 MHz timebase to the rear panel 10 MHz REFERENCE INPUT connector.

For external mode, connect an external source to SAMPLING CLOCK INPUT connector. The internal low pass filter is optimized for a 125 MHz sampling clock. Depending on the external clock frequency, additional filtering may be required.

Example

To select the internal 125 MHz sampling clock:

```
OUTPUT 719; "CLKSEL INT"
```

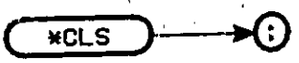
Comments

If INT is selected, any input to the SAMPLING CLOCK INPUT must be disconnected to prevent spurious mixing products.

Related Sections

CLKDIV

*CLS
(Clear Status Register)



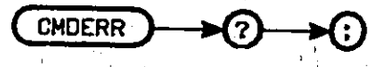
Description The *CLS command clears the Status Register and the event registers that drive it (that is, the Bit 1 Event, Bit 2 Event and Event Status registers). This command also clears any pending messages.

Example To clear the Status Register:

```
OUTPUT 719; "*CLS"
```

Related Sections
B1EVT
B2EVT
ESR
Programming the Status Register
STB

CMDERR
(Command Errors)



Description

Command errors are generated when the Synthesizer receives an invalid program command string. The error codes, described in Table 3-14, are the ones reported following a CMDERR? query.

CMDERR? returns only the first error that occurred since the last time the instrument was queried for command errors. Subsequent errors are not reported.

To recover from a command error, re-enter the command string with the correct mnemonic, data type or message terminator.

Program command *RST clears any command errors.

Table 3-14. Command Errors (1 of 3)

Error Code	Description	Comments
0	No error	
1	Invalid mnemonic received.	
2	Undefined data type.	5 data types are defined by IEEE Std 728: 1. strings - data enclosed in quotes 2. blocks - binary data in A, B, C, I, or L format 3. decimal numeric — +, —, ., 0—9, and E 4. nondecimal numeric — hex, octal, or binary 5. character - A—Z, 0—9 and _(underscore) Any other type of data will generate this error. The Synthesizer does not use string data, although it is a defined data type.
3	Character data is not allowed for the entered mnemonic.	
4	Character data overflow.	The maximum number of characters allowed is 8, although some commands only allow 6.
5	Invalid character token.	A character token is a word that follows a command. For example, INT is a character token for the CLKSEL command.

CMDERR (cont'd)
(Command Errors)

Description (cont'd)

Table 3-14. Command Errors (2 of 3)

Error Code	Description	Comments
6	Invalid data field separator.	Use a comma to separate multiple arguments of a command. Use a semi-colon to separate program commands from each other.
7	Not used.	
8	Not used.	
9	String data not allowed for the entered mnemonic.	
10	Improper block data termination.	
11	Block data not allowed for entered mnemonic.	
12	Invalid block type.	Only A, B, C, I and L block types are allowed.
13	Non-decimal numeric data is not allowed for the entered mnemonic.	
14	Invalid non-decimal numeric character.	Hexidecimal allows 0—9 and A—F; octal allows 0—7; binary allows 0, 1 and X (X = don't care). All other characters are invalid.
15	Non-decimal numeric data overflow.	64 bits is the maximum allowed.
16	Invalid non-decimal numeric terminator.	See Comments.
17	B-block data field checksum error.	The instrument calculates a checksum for the data field and compares it to the one entered.
18	Decimal data not allowed for the entered mnemonic.	
19	Improper decimal format.	
20	Invalid decimal data terminator.	See Comments.
21	Improper character within decimal data field.	The only characters allowed in a decimal data field are +, −, ., E and 0—9

CMDERR (cont'd)
(Command Errors)

Description (cont'd)

Table 3-14. Command Errors (3 of 3)

Error Code	Description	Comments
22	Exponent overflow.	A maximum of 3 digits are allowed for exponents.
23	Suffix not allowed for the entered mnemonic.	
24	Not used.	
25	Not used.	
26	Invalid message termination due to instrument being placed in talker active state (TACS).	
27	Invalid character data type character.	Valid characters are A—Z, 0—9, and _ (underscore).
28	Block length value for C-block is too small.	Include the two check bytes in block length
29	Block data CRC (cyclic redundancy check) error.	

Example

See Example 3-5, Error Checking Program, under Programming the Status Register in this section.

Comments

Command errors are caused by an error in a message unit. A message unit consists of a header, data and a terminator. For example, in the program command "ATTEN 20;" ATTEN is the header, 20 is the data and the semicolon (;) is the terminator.

Data must always be separated from the header by a space. A message unit can have multiple data elements. Multiple data elements are separated from each other by a comma.

Data can be one of four types:

1. decimal numeric
2. non-decimal numeric
3. block
4. character.

Valid decimal numeric characters are 0—9, E, the plus sign (+), the minus sign (−), the decimal point (.) and the underscore (_).

Non-decimal numeric data is hexidecimal, octal or binary. Valid hexidecimal characters are 0 through 9 and A through F. Valid characters for octal data are 0 through 7. Valid characters for binary data are 0, 1 and X (for don't care).

CMDERR (cont'd)
(Command Errors)

Comments (cont'd)

Block data is a block of binary data in one of five formats: A, B, C, I or L.

A-block format = <#><A> <length _ 16> <data>

B-block format = <#><B> <lenght _ 16> <data> <checksum>

C-block format = <#><C> <length _ 16> <data> <check byte> <check byte>

I-block format = <#><I> <data>. I blocks terminate by sending EOI with the last data byte.

L-block format = <#><L> <length _ 32> <data>

Refer to Hewlett-Packard Interface Bus Remote Operation in this section for a more detailed explanation of block data.

Character data always starts with an alpha character. After the first character, valid characters are A through Z, 0 through 9 and underscore (_). Valid message unit terminators are a semicolon (;), a line feed (LF) and a data byte with EOI.

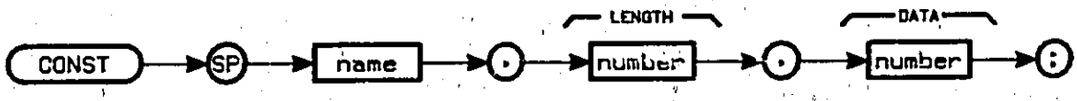
The Status Register can be programmed so that command errors will generate a Service Request (SRQ).

Related Sections

*ESE
*ESR
EXERR
Hewlett-Packard Interface Bus Remote Operation
Programming the Status Register
*SRE
*STB

CONST

(Fill Waveform Memory File With a Constant Value)



Item	Description	Range/Restrictions
Name	Waveform memory file name.	6 characters maximum
Number	Represents the length of the file.	1 to 131 072
Number	Data value with which file is to be filled.	0 to 4095 for signed format; −2048 to 2047 for unsigned format.

Description The CONST command creates a waveform memory file and fills it with a constant value. This command provides a fast method to fill large blocks of memory.

Example To create a file named TEMP with 1024 elements and have every element in TEMP be the decimal value 10:

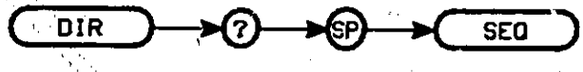
```
OUTPUT 719; "CONST TEMP, 1024, 10"
```

Comments The numbering system (signed or unsigned) depends on the FORMAT command.

Related Sections
FORMAT
GO
WAVE

DIR? SEQ

(Sequencer Directory)



Description

DIR? SEQ returns the directory of packets that define the current sequence. For each packet currently loaded in sequencer memory, three parameters are returned: file name, number of scans through the packet, and the packet advance mode.

The Synthesizer responds to the DIR? SEQ query with an ASCII string containing file name 1, scans 1, mode 1, ..., file name n scans n, mode n, carriage return and line feed. If a packet in the current sequence has no name associated with it, the returned string will contain "******" for the name of that packet. If no packets are defined, the query returns the string "* NO PACKETS IN SEQUENCER MEMORY*"

Example

To read the sequencer directory:

```
10 DIM A$ [5000]
20 OUTPUT 719; "DIR? SEQ"
30 ENTER 719; A$
40 PRINT "SEQUENCER DIRECTORY ="; A$
50 END
```

Comments

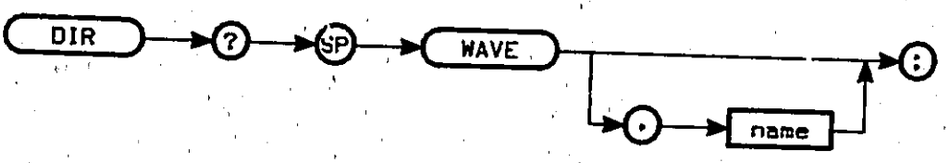
The sequencer directory can have 0 to 2048 entries.

Related Sections

GO
PACKET
PACLIT

DIR? WAVE

(Waveform Memory Named File Directory)



Item	Description	Range/Restrictions
Name	Waveform memory file name.	6 characters maximum

Description

DIR? WAVE returns an ASCII string containing the name and length of all named files that are currently loaded in waveform memory. The format of the string is file name 1, file length 1, file name 2, file length 2, ..., file name n, file length n, carriage return and line feed. If no files are loaded into waveform memory, the string "* NO NAMED DATA IN WAVE MEMORY *" is returned.

The Synthesizer responds to the DIR? WAVE, <file name> query with an ASCII string consisting of the file name, file length, carriage return and line feed. If the file name is not a defined waveform memory file, execution error 47 will be generated.

Example

To determine the length of a file named "Chirp":

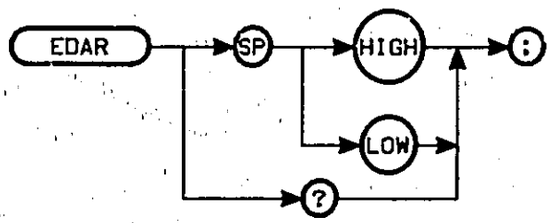
```
10 DIM A$[100]
20 OUTPUT 719; "DIR? WAVE, CHIRP"
30 ENTER 719; A$
40 PRINT A$
50 END
```

Related Sections

EXERR
GO
WAVE

EDAR

(External Data Received)



Description The EDAR command programs the edge sense of the EXTERNAL DATA INPUT port handshake line EDAR. The edges of the EDAR line indicate when a data element can be or has been accepted by the port. EDAR HIGH programs the port to be ready for data on the low-to-high state change. The high-to-low state change indicates that data has been accepted. EDAR LOW programs the port to be ready for data on the high-to-low state change. The low-to-high state change indicates that data has been accepted.

When queried (?), EDAR returns the currently active transition state of the EDAR handshake line, either HIGH or LOW.

Example To set the EDAR handshake line to be ready for data on the low-to-high transition:

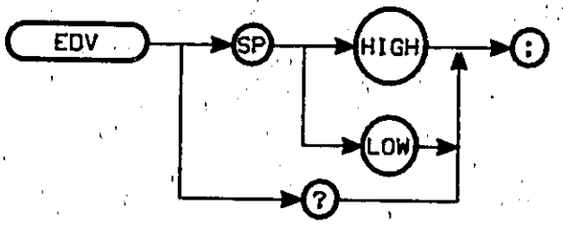
```
OUTPUT 719;"EDAR LOW"
```

Comments Power-up and program command *RST set the EDAR handshake line to LOW.

If the EXTERNAL DATA INPUT port is connected to the GPIO port of an HP 9000 Series 200 computer, make sure that the active edge sense of EDAR is the same as the GPIO's PCTL line.

Related Sections
EDV
*LRN
*RST

EDV
(External Data Valid)



Description

The EDV command programs the active edge sense of the EXTERNAL DATA INPUT port handshake line EDV. The active edge causes data to be accepted by the port. EDV HIGH programs the active edge on the low-to-high state change. EDV LOW programs the active edge on the high-to-low state change.

When queried (?), EDV returns the currently active transition state of the EDV handshake line, either HIGH or LOW.

Example

To set the EDV handshake line to be active on the high-to-low transition:

```
OUTPUT 719; "EDV LOW"
```

Comments

Power-up and program command *RST set the EDV line to low.

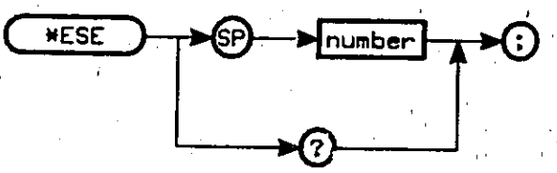
If the EXTERNAL DATA INPUT port is connected to the GPIO port of an HP 9000 Series 200 computer, make sure that the active edge sense of EDV is the same as the GPIO's PFLG line.

Related Sections

EDAR
*LRN
*RST

*ESE

(Event Status Register Enable)



Item	Description	Range/Restrictions
Number	Represents value of bits in Event Register to be enabled	0 to 255

Description

The *ESE command sets the Event Status Register enable field. The Event Status Register enable field enables bits in the Event Status Register, when true, to set bit 6 in the Status Register.

Bits in the Event Status Register enable field are identical to bits in the Event Status Register. The Event Status Register enable field and the value of each bit is shown below.

Event Status Register Enable Field

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Power On	0	Command Error	Execution Error	0	Query Error	0	Operation Complete
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

When queried (?), *ESE returns the current setting of the Event Status Register enable field.

Example

To enable command and execution errors, when they occur, to set bit 6 of the Status Register true and generate a Service Request (SRQ):

```
OUTPUT 719; "*ESE 48; *SRE 32"
```

Comments

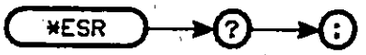
The Event Status Register enable field is set to zero at power-up.

Refer to Example 3-5 for an error checking program that will report command and execution errors when they occur. This program can be included as part of any new program as a debugging aid.

Related Sections

CMDERR
*ESR
EXERR
Programming the Status Register
*OPC
*SRE
*STB

*ESR
(Read Event Status Register)



Description

*ESR? returns an integer between 0 and 255. The integer represents the current value of the Event Status Register. The value of the Event Status Register is the sum of the values of all the true bits.

Bits in the Event Status Register, when enabled, drive bit 6 in the Status Register.

The Event Status Register and the decimal value of each bit is shown below.

Event Status Register

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
Power On	0	Command Error	Execution Error	0	Query Error	0	Operation Complete
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Once a bit is set, it remains set until the register is read. The register is set to zero after it has been read.

Example

To read the current value of the Event Status Register:

```
10 OUTPUT 719; "*ESR?"
20 ENTER 719; A
30 PRINT "EVENT STATUS REGISTER = "; A
40 END
```

Comments

The Event Status Register is set to zero at power-up. Programming commands *CLS and *RST also set the register to zero.

Refer to Example 3-5 for an error checking program that will report command and execution errors when they occur. This program can be included as part of any new program as a debugging aid.

Related Sections

*CLS
CMDERR
*ESE
EXERR
Programming the Status Register
*OPC
*RST
*SRE
*STB

EXDABT
(External Data Input Abort)

Description The EXDABT command aborts any pending EXTERNAL DATA INPUT port data transfers.

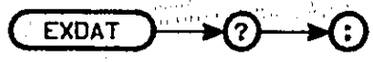
Example To abort a pending transfer of data via the EXTERNAL DATA INPUT port:

```
OUTPUT 719; "EXDABT"
```

Related Sections
EXDSEQ
EXDWAV
EXDWM

EXDAT

(External Data Input Query)



Description EXDAT? returns the number of words remaining to be transferred through the EXTERNAL DATA INPUT port.

If data is being transferred into waveform memory, the Synthesizer returns an integer between 0 and 131 072. If data is being transferred into sequencer memory, the Synthesizer returns an integer between 0 and 8192.

Example To determine the number of words remaining to be transferred:

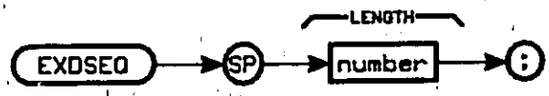
```
10 OUTPUT 719; "EXDAT?"
20 ENTER 719; A
30 PRINT "NUMBER OF WORDS TO BE TRANSFERRED = "; A
40 END
```

Comments This command should be used prior to issuing the GO command to see if all the data has been transferred.

Related Sections
EXDABT
EXDSEQ
EXDWAV
EXDWM

EXDSEQ

(External Data Input to Sequencer Memory)



Item	Description	Range/Restrictions
Number	Length of data in 16-bit words.	4 to 8192 in multiples of 4

Description

The EXDSEQ command prepares the Synthesizer to receive data from the rear panel EXTERNAL DATA INPUT port. This data will be placed in sequencer memory.

The data must be 16-bit binary signed values between −32768 and 32767. (The FORMAT command has no effect on EXDSEQ data transfers.)

Executing EXDSEQ purges the data in sequencer memory before the new data is loaded. Once EXDSEQ is executed, data can be transferred into the EXTERNAL DATA INPUT port. The External Data Input mode is automatically terminated when the Synthesizer receives the number of items specified.

It takes four 16-bit words to define a packet. The data that is loaded is a specially formatted version of the packet data. This data can be obtained using the SEQUEN? command.

Example

The following example shows how to convert sequencer data to the proper format so that it can be returned to the Synthesizer using the EXTERNAL DATA INPUT port. The example also shows how to send sequencer data back to the Synthesizer using the EXTERNAL DATA INPUT port. (In the program the Synthesizer is referred to as AWS, for Arbitrary Waveform Synthesizer.)

The example makes the following assumptions:

1. A sequence is currently defined in sequencer memory.
2. Data is being transferred to the Synthesizer from a computer with a GPIO interface and GPIO BASIC binary extension.

EXDSEQ (cont'd)

(External Data Input to Sequencer Memory)

Example 3-6. EXDSEQ

```
!************************* EXDSEQ EXAMPLE ************************************
COM /Aws_add/ INTEGER Hpib                  ! AWS HPIB ADDRESS
COM /Qq/ A$[16386],INTEGER Seq1(1:8192)
CALL Gpio_2_seq                             !
END
                                            !
                                            !
Gpio_2_seq:SUB Gpio_2_seq
    COM /Aws_add/ INTEGER Hpib
    COM /Qq/ A$[16386],INTEGER Seq1(1:8192)
    INTEGER Header_1                        ! HOLDS "#I"
    Hpib=719                                ! AWS HPIB ADDRESS
    ASSIGN @Aws TO Hpib                     ! PATH TO AWS
    ASSIGN @Gpio TO 12;FORMAT OFF,WORD      ! PATH TO GPIO
    ASSIGN @In TO Hpib;FORMAT OFF           ! PATH FROM AWS FORMAT OFF
                                            !
    OUTPUT @Aws;"PACLIT?"                   ! ASK FOR SEQ DATA IN USER FORMAT
    ENTER @Aws;A$                           ! GET SEQUENCE DATA
    PRINT "SEQUENCE=",A$                    ! PRINT SEQUENCE
    WAIT 3
                                            !
    OUTPUT @Aws;"SEQUEN? BIN"               ! ASK FOR SEQ DATA IN SEQ FORMAT
    ON TIMEOUT 7,2 GOTO Done                ! DO NOT KNOW EXACT AMOUNT
    ENTER @Aws;A$                           ! GET SEQUENCER DATA
Done:                                       ! THE NUMBER OF PACKETS IN THE SEQ.
    OFF TIMEOUT                             ! IS NEEDED FOR THE EXDSEQ COMMAND
                                            !
    Bytes=INT(LEN(A$))                      ! NUMBER OF BYTES RETURNED
    Word_count=(Bytes-2)/2                  ! NUMBER OF INTEGER WORDS. -2 REMOVES
                                            ! "#I" HEADER FROM DATA
    Packet=Word_count/4                     ! NUMBER OF PACKETS IN SEQUENCER
                                            !
    REDIM Seq1(1:Word_count)                ! EXDSEQ EXPECTS DATA SENT AS
                                            ! INTEGER WORDS.
    OUTPUT @Aws;"SEQUEN? BIN"               ! ASK FOR SEQUENCER DATA IN WORDS
    ENTER @In;Header_1,Seq1(*)              ! PLACE DATA IN INTEGER ARRAY
                                            ! SEQ1(*) CAN BE SAVED ON DISC OR
                                            ! SENT TO AWS OVER GPIO
                                            !
!****************** SEND SEQUENCER DATA BACK OVER GPIO ********************
                                            !
    OUTPUT @Aws;"EXDSEQ "&VAL$(Word_count)! TELL AWS WHAT IS COMMING
    WAIT 1                                  !
    OUTPUT @Gpio;Seq1(*),END                ! SEND SEQUENCER DATA
    WAIT 1                                  !
    OUTPUT @Aws;"GO"                        ! START SEQUENCE
    REDIM Seq1(1:8192)                      ! SETUP FOR NEXT TIME
SUBEND
```

EXDSEQ (cont'd)

(External Data Input to Sequencer Memory)

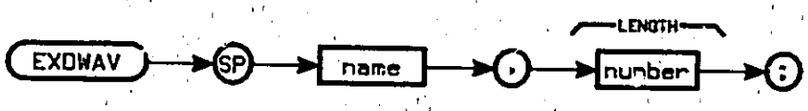
Comments

Data is transferred into the Synthesizer using a two-wire handshake. Refer to the EDAR and EDV commands.

Related Sections

EDAR
EDV
EXDABT
EXDAT
SEQUEN
EXDAT
GO
SEQUEN

EXDWAV
(External Data Input to Waveform Memory)



Item	Description	Range/Restrictions
Name	File name in waveform memory where incoming data is to be placed	6 characters maximum.
Number	Length of the data string in 16-bit words.	1 to 131 072

Description

The EXDWAV command prepares the Synthesizer for input via the rear panel EXTERNAL DATA INPUT port. This data will be placed in a named file in waveform memory.

Once the EXDWAV command is executed, data can be transferred into the EXTERNAL DATA INPUT port. The data is 12-bit binary. The port is 16-bits wide, but the values of the upper 4 bits of the port are ignored.

The numbering convention is defined by the FORMAT command. The allowable range is −2048 to 2047 for signed data and 0 to 4095 for unsigned data. The numbering convention in effect at the time EXDWAV is executed is used for the transfer.

Example

Example 3-4 provides an example of loading waveform memory using the EXTERNAL DATA INPUT port.

Comments

For the fastest transfers into waveform memory, use the unsigned format option.

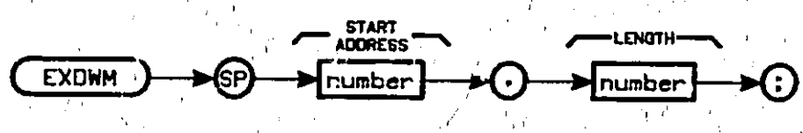
Data is transferred using a two-wire handshake. Refer to the EDAR and EDV commands.

Related Sections

EDAR
EDV
EXDABT
EXDAT
FORMAT
GO

EXDWM

(External Data Input to Waveform Memory)



Item	Description	Range/Restrictions
Number	Starting address in waveform memory where the incoming block of data is to be placed.	0 to 131 071
Number	Length of the data string in 16-bit words.	1 to 131 072
NOTE: The sum of the starting address and the length must be less than or equal to 131 072.		

Description

The EXDWM command prepares the Synthesizer for input via the rear panel EXTERNAL DATA INPUT port. This data will be placed in the specified address location in waveform memory.

Once the EXDWM command is executed, data can be transferred into the EXTERNAL DATA INPUT port. The data is 12-bit binary. The port is 16-bits wide, but the values of the upper 4 bits of the port are ignored.

The numbering convention is defined by the FORMAT command. The allowable range is −2048 to 2047 for signed data and 0 to 4095 for unsigned data. The numbering convention in effect at the time EXDWM is executed is used for the transfer.

Example

To transfer data into waveform memory using the EXTERNAL DATA INPUT port:

```
10  INTEGER A (0:1023), I
20  ASSIGN @Cpio TO 12; FORMAT OFF, WORD
30  OUTPUT 719; "EXDABT; PURGE BOTH"
40  OUTPUT 719; "EXDWM 0, 1024"
50  FOR I=0 TO 1023
60  A(I)=I*2
70  NEXT I
80  OUTPUT @Gpio; A(*)
90  OUTPUT 719; "PACLIT 0, 1, AUTO; GO"
100 END
```

This example is similar to Example 3-4. The only difference is that lines 40 and 90 use commands that require addresses instead of file names. Example 3-4 explains this program in more detail.

Comments

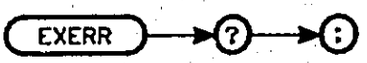
For the fastest transfers into waveform memory, use the unsigned format option.

Data is transferred using a two-wire handshake. Refer to the EDAR and EDV commands.

Related Sections

EDAR
EDV
EXDABT
EXDAT
FORMAT
GO

EXERR
(Execution Errors)



Description

Execution errors are generated when the Synthesizer cannot execute a command because a parameter is out of range. The error codes, described in Table 3-15, are returned following an EXERR? query.

EXERR? returns only the first error that occurred since the last time the instrument was queried for execution errors.Subsequent errors are ignored.

To recover from an execution error, re-enter the command string with the correct parameters. Refer to the detailed description of the command listed in the Command column for additional information on parameter ranges.

Program command *RST clears any execution errors.

Table 3-15. Execution Errors (1 of 4)

Error Code	Command	Description
0		No error
1	BLKMOV	Starting address out of range
2	BLKMOV	Block length value out of range
3	BLKMOV	Source block length too large
4	BLKMOV	Destination address out of range
5	BLKMOV	Destination block length too large
6		Not used
7	CLKDIV	Divide number entered is out of range or invalid.
8	MUXSEL	Entered value is out of range. See Section VIII, Service.
9	PEEK.X POKE.X	Entered addressed value is out of range. See Section VIII, Service.
10	PEEK.W/L POKE.W/L	Entered address refers to an odd address boundary. See Section VIII, Service
11	POKE.B/W/L	Entered data point is too large. See Section VIII, Service
12	SEQUEN	Block transfer does not contain an even number of bytes.
13	SEQUEN	Entered data is out of range

EXERR (cont'd)
(Execution Errors)

Description (cont'd)

Table 3-15. Execution Errors (2 of 4)

Error Code	Command	Description
14	SEQUEN	Data record overflow
15	SEQUEN	Data record is not multiple of 4
16	SEQUEN?	No sequence currently defined
17-19		Not used
20	WMEM	Starting address value out of range
21		Not used
22	WMEM	Data overflow
23	WMEM	Waveform data value out of range
24	WMEM	Block data transfer does not contain even number of bytes
25		Not used
26	WMEM?	Starting address value out of range
27	WMEM?	Length value out of range
28	WMEM?	Defined data block too large
29	PACKET PACLIT	Sequencer memory full
30	PACLIT	Defined packet length too large
31	PACLIT	Packet starting address out of range
32	PACLIT	Packet length value is out of range
33	PACKET PACLIT	Packet scan value is out of range
34	PACKET PACLIT	Length is too small. Need at least 344 elements per packet.
35	PACKET PACLIT	Wave segment length is too small. Must be ≥56 elements.
36	PACLIT	Packet start address is not a multiple of 8.

EXERR (cont'd)
(Execution Errors)

Description (cont'd)

Table 3-15. Execution Errors (3 of 4)

Error Code	Command	Description
37	PACKET PACLIT	Wave segment is not a multiple of 8.
38	PACLIT?	No packets are currently defined
39	ATTEN	dB value out of range
40	All commands	Invalid mnemonic format
41	PEEK POKE	Address field contains a "don't care" character. See Section VIII, Service.
42	POKE	Data field contains a "don't care" character. See Section VIII, Service.
43	MARKADD	Entered address value is out of range
44	B1EN B1FAL B1RIS	Entered bit enable field is out of range
45	B2EN B2FAL B2RIS	Entered bit enable field is out of range
46	*ESE *PER *SRE	Entered bit enable field is out of range
47	General	File name not defined
48	General	Name directory is full
49		Not used.
50	General	File name field contains more than 6 characters or is a reserved name (Reserved names are ASCII, BIN, WAVE, SEQ, BOTH.)
51	GO	No packets are currently defined
52	EXDSEQ	Data block length is out of range
53	EXDSEQ	Data block length is not a multiple of 4
54	EXDWM	Starting address is out of range

EXERR (cont'd)
(Execution Errors)

Description (cont'd)

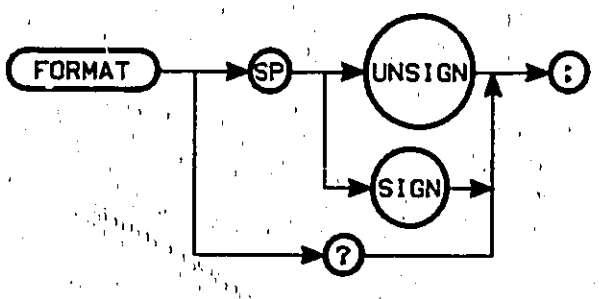
Table 3-15. Execution Errors (4 of 4)

Error Code	Command	Description
55	EXDWM	Block length is out of range
56	EXDWM	Defined block is too large
57	EDAR EDV	Handshake control line sense cannot be changed while EXTERNAL DATA INPUT port is active
58	SINPQ	P or Q is out of range
59	SINPQ	Ratio P/Q > 0.5
60	SINPQ	Entered length, Q, is not a multiple of 8. The length value was truncated to be a multiple of 8.
61	TST?	Entered test number is out of range. See Section VIII, Service.
62	CONST EXDWAV	Data block length is out of range
63	CONST EXDWAY	Data block is too large for current waveform memory free space
64	CONST	Constant data value is out of range
65	WAVEAP	Defined file name is not the last file entered
66	LRNDIR?	Incoming directory learn block is too long
67	LRNDIR?	Incoming directory learn block is too small
68	SCALE	Scale factor is out of range
69	General	More than one query within a message record

Comments

Refer to Example 3-5 for an error checking program that will report command and execution errors when they occur. This program can be included as part of any new program as a debugging aid.

FORMAT
(Waveform Memory Numbering Format)



Description

The FORMAT command selects the numbering convention for data transferred to/from waveform memory.

Each element in waveform memory is represented by a 12-bit computer word. The Synthesizer's internal digital-to-analog converter (DAC) converts this word to one of 4096 different amplitude levels. The FORMAT command determines how the Synthesizer interprets the data. FORMAT SIGN values range from −2048 to 2047. FORMAT UNSIGN values range from 0 to 4095.

When queried (?), FORMAT returns its current status, either SIGN or UNSIGN, in an ASCII string.

Example

To select signed format for data transfers to and from waveform memory:

```
OUTPUT 719; "FORMAT SIGN"
```

Comments

Only data to or from waveform memory can have either numbering system. Data to or from the sequencer memory is always signed.

Sign selects two's complement form and unsign selects straight binary weighting.

Power-up and program command *RST set FORMAT to UNSIGN.

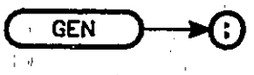
Related Sections

EXDWAV
EXDWM
*RST
WAVE
WMEM

*PER
(Parallel Poll Enable)

GEN

(Generate a Sine Wave)



Description The GEN command generates a 12.5 MHz sine wave. This command simulates the execution of the following commands in the order listed:

```
*RST
SINPQ TEMP,100,1000
PACKET TEMP,1,AUTO
GO
```

Example To generate a 12.5 MHz sine wave

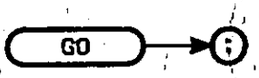
```
OUTPUT 719; "GEN"
```

Comments The sine wave is stored in waveform memory in a file named TEMP.

Related Sections

GO
PACKET
*RST
SINPQ
WAVE

GO



Description The GO command starts the sequencer running at the first packet in sequencer memory.

Example To start the sequencer running:

```
OUTPUT 719;"GO"
```

Comments A sequence must be loaded in sequencer memory prior to issuing the GO command. Otherwise, execution error 51 is generated.

The GO command aborts any pending query.

The GO command has no effect on the OUTPUT command. If the OUTPUT command is set to OFF when the Synthesizer receives the GO command, the output from the RF OUTPUT connector will be 0 Vdc.

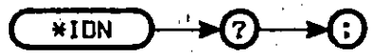
The following commands, when executed, stop the sequencer. The GO command must be issued to restart the sequencer.

BLKMOVE	EXDWM	SEQUEN
CLKDIV	GEN	SEQUEN?
CONST	LRNDIR	SINPQ
DIR? SEQ	PACKET	WAVE
DIR? WAV	PACLIT	WAVE?
EXDSEQ	PACLIT?	WAVEAP
EXDWAV	PURGE	WMEM
	SCALE	WMEM?

Related Sections

EXERR
PACKET
PACLIT
STOP

*IDN
(Identification)



Description *IDN? identifies the Synthesizer's model number and firmware version. In response to the *IDN? query, the Synthesizer returns the following string:

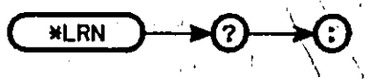
HP,8770A,0,X.XX <CR> <LF with EOI>

HP is the manufacturer, 8770A is the model number, 0 is always 0 and X.XX is the firmware version number. CR is carriage return, LF is line feed and EOI is end-or-identify.

Example To determine the instrument's firmware version:

```
10 OUTPUT 719; "*IDN?"
20 ENTER 719; A$
30 PRINT "INSTRUMENT IS"; A$
40 END
```

*LRN
(Learn Mode)



Description

In response to a *LRN? query, the Synthesizer sends out an ASCII string containing eight instrument settings. These settings can be stored in a string variable in computer memory or on a disc. When this string is returned to the Synthesizer, the instrument settings change to the returned values.

The following eight instrument settings are captured by learn mode:

- ATTEN
- CLKDIV
- CLKSEL
- MARKADD
- FORMAT
- OUTPUT
- EDV
- EDAR

Data is sent in an ASCII character string. The format of the string is shown below.

ATTEN <dB>;CLKDIV <integer>;CLKSEL <INT/EXT>;MARKADD <address>; FORMAT <SIGN/UNSIGN>;OUTPUT <ON/OFF>;EDV <HIGH/LOW>; EDAR <HIGH/LOW> <CR> <LF with EOI>

Example

To save the current Synthesizer settings for future use:

```
10 DIM A$[130]
20 OUTPUT 719;"*LRN?"
30 ENTER 719; A$
40 END
```

To return the settings to the Synthesizer:

```
OUTPUT 719; A$
```

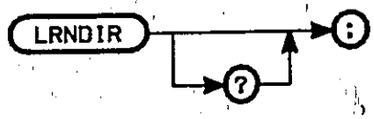
Comments

No waveform or sequencer memory data is sent in the learn string.

Related Sections

ATTEN
EDAR
EDV
CLKDIV
CLKSEL
FORMAT
MARKADD
OUTPUT

LRNDIR
(Learn Directory)



Description

LRNDIR?, in conjunction with WMEM?, SEQUEN? and *LRN?, provides a way to save all the information in the Synthesizer's memory.

In response to the LRNDIR? query, the Synthesizer outputs a message containing its current waveform memory file directory. The message is in the form of A-block data with a fixed length of 14 028 data bytes. (Be sure to dimension an array for 14 032 bytes because #, A, and 2 length bytes are also sent in the message.)

Information obtained from the LRNDIR? command is returned to the instrument by using the LRNDIR command.

Example

Example 3-3, Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory, uses LRNDIR. Refer to the subroutine "COPY DIRECTORY FROM AWS TO DISC (Dir _ to _ disc)" for an example of LRNDIR?. Refer to the subroutine "COPY DIRECTORY FROM DISC TO AWS (Dir _ to _ aws)" for an example of LRNDIR.

Comments

If data was stored in waveform memory by address instead of by file name, there is no file directory. Therefore, LRNDIR? is not needed to save the contents of the Synthesizer's memory. Only WMEM? (for waveform memory data), SEQUEN? (for sequencer data) and *LRN (for hardware settings) need be used.

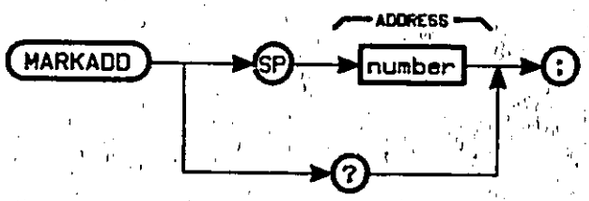
To read the file directory, use the query DIR? WAVE.

Related Sections

DIR? WAVE
GO
*LRN
SEQUEN
WMEM

MARKADD

(Set Address Marker by Address)



Item	Description	Range/Restrictions
Number	Represents an address in waveform memory	0 to 131 071

Description

The MARKADD command sets the address marker to a specified address in waveform memory. Whenever the specified waveform memory address occurs, a TTL level pulse is output on the rear panel EQUAL ADDRESS connector. The pulse is high for a minimum of 8 x (1/sampling clock).

When queried (?), MARKADD returns the waveform memory address of the marker setting.

Example

To create a ramp of 1024 elements and set the marker at midpoint on the ramp (address 512):

```
10  INTEGER A(1:1024)
20  OUTPUT 719;"*RST"
30  FOR I=1 TO 1024
40    A(I)=I
50  NEXT I
60  OUTPUT 719;"WMEM 0,";A(*)
70  OUTPUT 719;"PACLIT 0, 1024, 64, AUTO"
80  OUTPUT 719;"MARKADD 512"
90  OUTPUT 719;"GO"
100 END
```

Line 20: Resets Synthesizer to default parameters.
Lines 30—50: Creates data array.
Line 60: Loads the data array into waveform memory, starting from address 0
Line 70: Creates a sequence of one packet.
Line 80: Sets address marker to address 512.
Line 90: Starts sequencer running.

Comments

Only addresses that are multiples of 8 are valid. Other addresses are truncated to the nearest multiple of 8. For example, the command string OUTPUT 719; "MARKADD 130" sets the marker to 128, the nearest multiple of 8. The query MARKADD? returns the truncated value, 128.

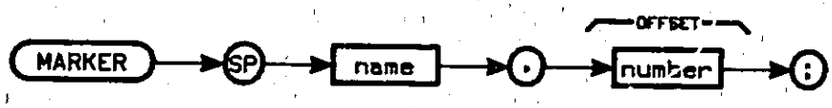
Power-up and program command *RST set MARKADD to 0.

Related Sections

PACLIT WMEM
*RST

MARKER

(Set Address Marker by File Name)



Item	Description	Range/Restrictions
Name	Waveform memory file name	6 characters maximum
Number	Represents added offset from the starting address of the file	0 to the length of the named file. Must be a multiple of 8.

Description

The MARKER command sets the address marker to the start address of a named waveform memory file, plus offset. The added offset must be a multiple of eight. Offsets that are not multiples of eight are truncated to the nearest multiple of eight.

Whenever the starting address plus offset of the specified waveform memory file occurs, a TTL level pulse is output on the EQUAL ADDRESS rear panel connector. The pulse is high for a minimum of 8 x (1/sampling clock).

Example

To create a sine wave and set the address marker at the midpoint of the sine wave:

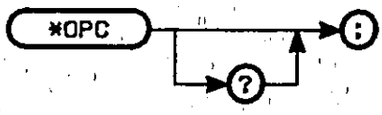
```
10 OUTPUT 719;"*RST"
20 OUTPUT 719;"SINPQ FREQ1, 1, 1024"
30 OUTPUT 719;"PACKET FREQ1, 64, AUTO"
40 OUTPUT 719;"MARKER FREQ1, 512"
50 OUTPUT 719;"GO"
60 END
```

Line 10: Resets Synthesizer to default parameters.
Line 20: Creates sine wave consisting of 1 cycle of 1024 elements and stores the data in waveform memory file FREQ1.
Line 30: Creates sequence of one packet.
Line 40: Sets address marker to 512 elements from the starting address of waveform memory file FREQ1.
Line 50: Starts sequencer running.

Related Sections

PACKET
WAVE

*OPC
(Operation Complete)



Description

The *OPC command sets the operation complete bit (bit 1) in the Event Status Register true.

When queried (?), *OPC always returns a "1" and sets bit 1 in the Event Status Register true.

Example

To generate an SRQ to indicate when the Synthesizer has executed a string of commands:

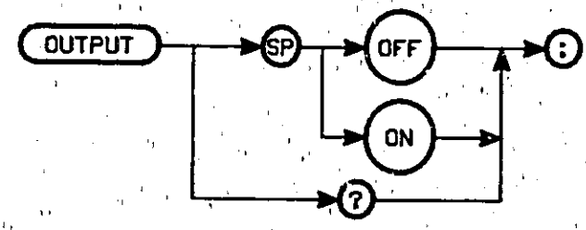
```
10 OUTPUT 719;"*ESE 1;*SRE 32"
20 OUTPUT 719;"GEN;CLKDIV 1;ATTEN 0;*OPC"
30 END
```

Line 10: Sets Event Status Register and Status Register enable fields.
Line 20: Sends *OPC as the last command in a string of commands to indicate when the Synthesizer has executed the commands.

Related Sections

*ESE
*ESR
*SRE
*STB

OUTPUT



Description The OUTPUT command turns the Synthesizer's RF output either on or off. Turning the RF output off puts the output to 0 Vdc. Turning the output off, however, does not stop the sequencer.

If the sequencer is not running, the output is at 0 Vdc, regardless of the whether the output is on or off.

When queried (?), OUTPUT returns an ASCII string containing the current setting, either ON or OFF.

Example To turn the RF output off:

```
OUTPUT 719; "OUTPUT OFF"
```

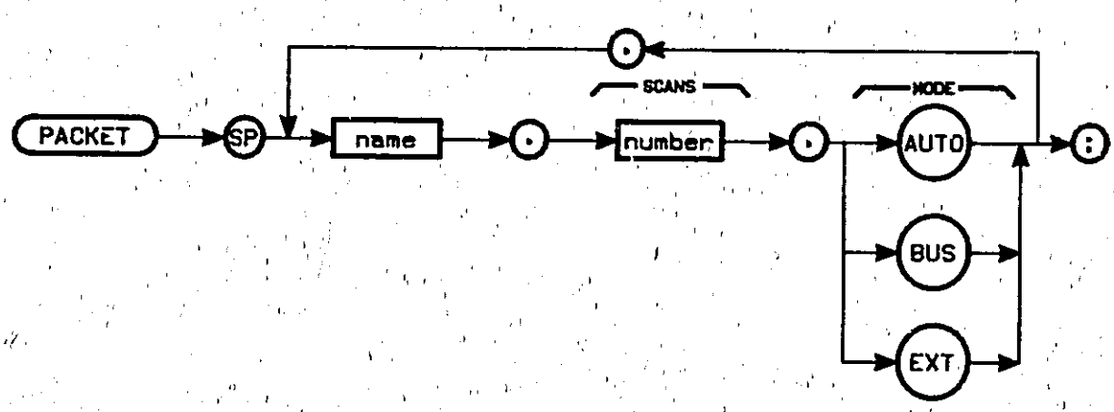
Comments When OUTPUT OFF is selected, the RF output is not floating.

Power-up and program command *RST set the OUTPUT to ON.

Related Sections
ATTEN
*RST
STOP

PACKET

(Create Packet Using File Names)



Item	Description	Range/Restrictions
Name	Defines waveform memory file name that the sequencer should access for this packet.	6 characters maximum
Number	Defines the number of passes through the waveform memory file.	1 to 65 536 for AUTO mode; 0 to 65 535 for EXT and BUS modes

Description

The PACKET command defines one or more packets for the current sequence. This new packet is appended to any other packets that may have been defined since a "PURGE SEQ" or "PURGE BOTH" command was received by the Synthesizer. A maximum of 2048 packets can be defined.

There are three ways to exit a packet and advance to the next packet in a sequence: automatically (AUTO), external trigger (EXT) or HP-IB trigger (BUS). For AUTO, the sequencer continues to the next packet in the sequence after the specified number of scans has occurred. For EXT trigger, a user-provided trigger exits the packet. Two packet advance TTL control lines are provided on the rear panel: READY OUTPUT and TRIGGER INPUT. When the READY OUTPUT goes TTL high, it informs the user that the TRIGGER INPUT is active. If the TRIGGER INPUT is then pulsed high for at least 25 ns the sequencer exits the packet at the end of the current scan. (TRIGGER INPUT is positive-edge sensitive.) For BUS trigger, the packet exits at the end of the current scan only when the HP-IB command "ADVSEQ" has been received by the Synthesizer.

For both the EXT and BUS triggers, the number of scans is indefinite. However, the scan parameter must be present in the HP-IB command statement.

Example

To create two packets:

```
10 OUTPUT 719;"*RST"
20 OUTPUT 719;"SINPQ FREQ1, 1, 1024; SINPQ FREQ2, 2, 1024"
30 OUTPUT 719;"PACKET FREQ1, 0, BUS, FREQ2, 10, AUTO"
40 OUTPUT 719;"GO"
50 END
```

PACKET (cont'd)

(Create Packet Using File Names)

Example (cont'd)

Line 10: Resets Synthesizer to default parameters.
Line 20: Creates two wave segments, FREQ1 and FREQ1, and stores the data in waveform memory.
Line 30: Creates sequence of two packets. In the first packet, wave segment FREQ1 is scanned indefinitely until the Synthesizer receives the ADVSEQ command. The sequencer then advances to the next packet. In the second packet, FREQ2 is scanned 10 times and then the sequencer automatically returns to the first packet in the sequence.
Line 40: Starts the sequencer running.

Comments

A packet can only be built if the named waveform memory file already exists. Be sure all waveform data is loaded before the packets are defined.

PACKET and PACLIT commands should not be intermixed in a sequence.

Reminders:

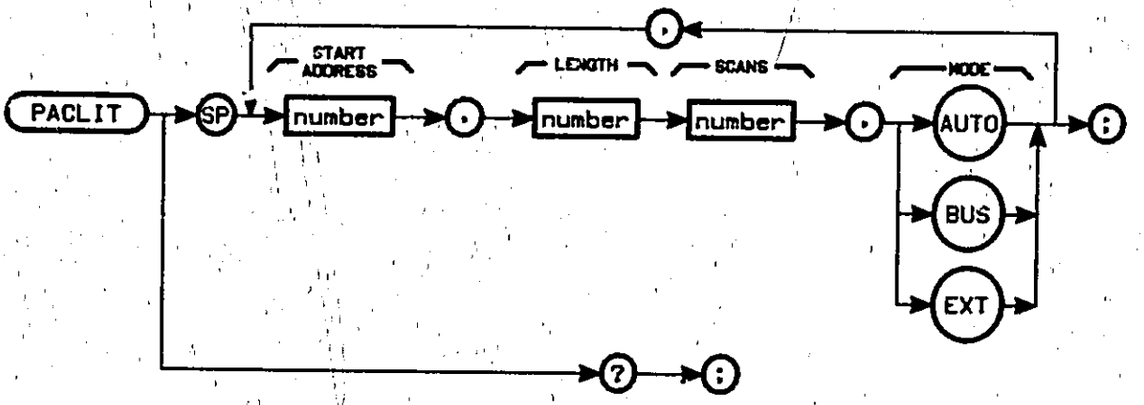
- Minimum length of a named wave segment must be 56 elements.
- Wave segment length must be a multiple of 8.
- Minimum packet length (wave segment length x scans) must be 344 elements.

Related Sections

ADVSEQ
DIR? SEQ
GO
PACLIT
WAVE

PACLIT

(Create Packet Using Addresses)



Item	Description	Range/Restrictions
Number	Represents starting address in waveform memory that sequencer should access for this packet.	0 to 131 064. Must be multiple of 8.
Number	Represents length of data (number of elements) in waveform memory that sequencer should packet.	56 to 131 072. Must be a multiple of 8
Number	Represents number of passes through the specified data in waveform memory.	1 to 65 536 for AUTO mode; 0 to 65 536 for EXT and BUS modes.

Description

The PACLIT command defines one or more packets for the current sequence. This new packet is appended to any other packets that may have been defined since a "PURGE SEQ" or "PURGE BOTH" command was received by the Synthesizer. A maximum of 2048 packets can be defined.

There are three ways to exit a packet and advance to the next packet in a sequence: automatically (AUTO), external trigger (EXT) or HP-IB trigger (BUS). For AUTO, the sequencer continues to the next packet in the list after the specified number of scans hae occurred. For EXT trigger, a user-provided trigger exits the packet. Two packet advance TTL control lines are provided on the rear panel: READY OUTPUT and TRIGGER INPUT. When the READY OUTPUT goes TTL high, it informs the user that the TRIGGER INPUT is active. If the TRIGGER INPUT is then pulsed high for at least 25 ns the sequencer exits the packet at the end of the current scan. (TRIGGER INPUT is positive-edge sensitive.) For BUS trigger, the packet exits at the end of the current scan only when the HP-IB command "ADVSEQ" has been received by the Synthesizer.

When queried (?), PACLIT returns four parameters that define the existing sequence: start address, length of packet, number of scans, and mode for packet advance. The returned values are formatted in an ASCII string and are returned in the order they were received.

PACLIT (cont'd)
(Create Packet Using Addresses)

Example

To create a packet using address locations to define the wave segment:

```
10 INTEGER A(1:1024)
20 OUTPUT 719;"*RST"
30 FOR I=1 TO 1024
40   A(I)=I
50 NEXT I
60 OUTPUT 719;"WMEM 0,"; A(*)
70 OUTPUT 719;"PACLIT 0, 1024, 64, AUTO"
80 OUTPUT 719;"GO"
90 END
```

Line 20: Resets Synthesizer to default parameters.
Lines 30—50: Creates data array.
Line 60: Loads the data array into waveform memory, starting from address 0.
Line 70: Creates packet. A wave segment beginning at address 0, 1024 elements long, is scanned 64 times before the sequencer automatically advances to the next packet in the sequence.
Line 80: Starts sequencer running.

Comments

The PACLIT command literally defines the waveform memory wave segment using a start address and length whereas the PACKET command uses file names. The PACLIT command is useful when data was loaded into waveform memory using a start address rather than a file name.

PACKET and PACLIT commands should not be intermixed in a sequence.

For both the EXT and BUS triggers, the number of scans is indefinite. However, the scan parameter must be present in the HP-IB command statement.

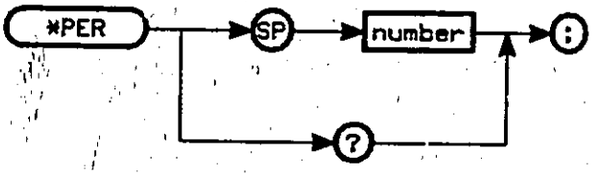
Reminders:

- Wave segment length must have at least 56 elements and be be a multiple of 8.
- Minimum packet length (wave segment length x scans) must be 344 elements.

Related Sections

ADVSEQ
GO
PACKET
WMEM

*PER
(Parallel Poll Enable)



Item	Description	Range/Restrictions
Number	Represents value of bits in the Status Register being enabled.	0 to 255

Description

The *PER command enables bits in the Status Register, when true, to cause the Synthesizer to respond affirmatively to a parallel poll. The logic sense of the Synthesizer's response to a parallel poll is determined by the system controller.

Bits in the Parallel Poll enable field are logically ANDed with bits in the Status Register. If the result is one, the Synthesizer will respond affirmatively on its assigned bit when parallel polled.

The Parallel Poll enable field and the decimal value of each bit are shown below.

Parallel Poll Enable Field

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	Require Service	Events	Message Available	0	0	Operating Conditions	Hardware Errors
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

To determine the value to enter for the *PER command, add the decimal value of each bit that is being enabled.

When queried (?), *PER returns the current setting of the parallel poll enable field.

Example

To configure the Synthesizer to respond to a parallel poll with positive-true logic on HP-IB data line DIO3 when bit 7 (Require Service) in the Status Register is true:

```
10 PARALLEL POLL CONFIGURE 719;10
20 OUTPUT 719; "*PER 64"
30 END
```

*PER (cont'd)
(Parallel Poll Enable)

Example (cont'd)

Line 10: Configures the logic sense and data line on which Synthesizer will respond to a parallel poll.
Line 20: Sets Parallel Poll enable field.

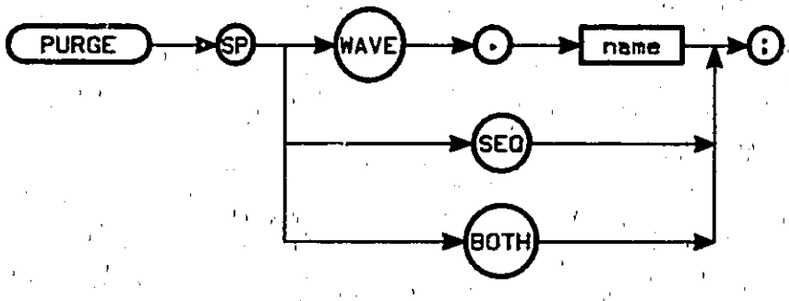
Comments

At power-up the Parallel Poll enable field is set to zero.

Related Sections

B1EN
B2EN
*ESE
Programming the Status Register
*SRE

PURGE



Item	Description	Range/Restrictions
Name	Represents a named waveform memory file.	6 characters maximum

Description The PURGE command can purge data in sequencer memory, both sequencer memory and the entire waveform memory or a specified waveform memory file.

Example To purge a waveform memory file named SINX:

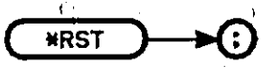
```
OUTPUT 719; "PURGE WAVE, SINX"
```

Comments When purging waveform memory files using the PURGE WAVE <file name> command, the new free space can be used for new data files.

PURGE BOTH is executed at power-up and by programming command *RST.

Related Sections
EXDSEQ
EXDWAV
EXDWM
GO
PACKET
PACLIT
*RST
WAVE
WMEM

*RST
(Reset)



Description The *RST command resets the Synthesizer to the default parameters shown in Table 3-16. Note that the Synthesizer executes a PURGE BOTH command prior to setting the default values.

Table 3-16. Reset Conditions

Parameter	Default Value
ATTEN	0 dB
B1EVT	0
B2EVT	0
CLKDIV	1
CLKSEL	INT
CMDERR	0
EDAR	LOW
EDV	LOW
*ESR	0
EXDABT	
EXERR	0
FORMAT	UNSIGN
MARKADD	0
MUXSEL	255 (See Section VIII, Service)
OUTPUT	ON (The sequencer is not running. Output=0V)
PURGE	BOTH
*STB	0

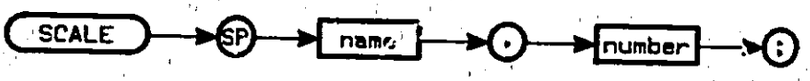
Example To set the Synthesizer to a known state:

```
OUTPUT 719; "*RST"
```

Related Sections

ATTEN
B1EVT
B2EVT
CLKDIV
CLKSEL
CMDERR
EDAR
EDV
*ESR
EXDABT
EXERR
FORMAT
MARKADD
OUTPUT
PURGE
*STB

SCALE



Item	Description	Range/Restrictions
Name	Name of waveform memory file to be scaled.	6 characters maximum
Number	Represents scale factor. A floating point number.	1.00000 to 0.01000. Any digits below the 10^{-5} position are ignored.

Description

The SCALE command scales a named wave segment by a scale factor. The scale factor is the percentage by which the wave segment is to be reduced. The scale factor can range from 1.0 to 0.01. The wave segment can only be reduced in power; it cannot be scaled up.

Example

To generate a sine wave and then reduce it by 50%:

```
10 OUTPUT 719; "GEN"
20 OUTPUT 719; "SCALE TEMP, .5"
30 OUTPUT 719; "GO"
40 END
```

Line 10: Generates a 12.5 MHz sine wave, stores waveform data in waveform memory file TEMP, and starts sequencer.
Line 20: Scales TEMP by 50%.
Line 30: Starts sequencer so that scaled waveform can be viewed on an external piece of test equipment, such as an oscilloscope.

Comments

To reduce a wave segment by dB rather than percent, use the following equation to calculate the scale factor:

$$\text{Scale Factor} = 10^{-x/20}$$

where x is the value in dB by which the wave segment is to be reduced.

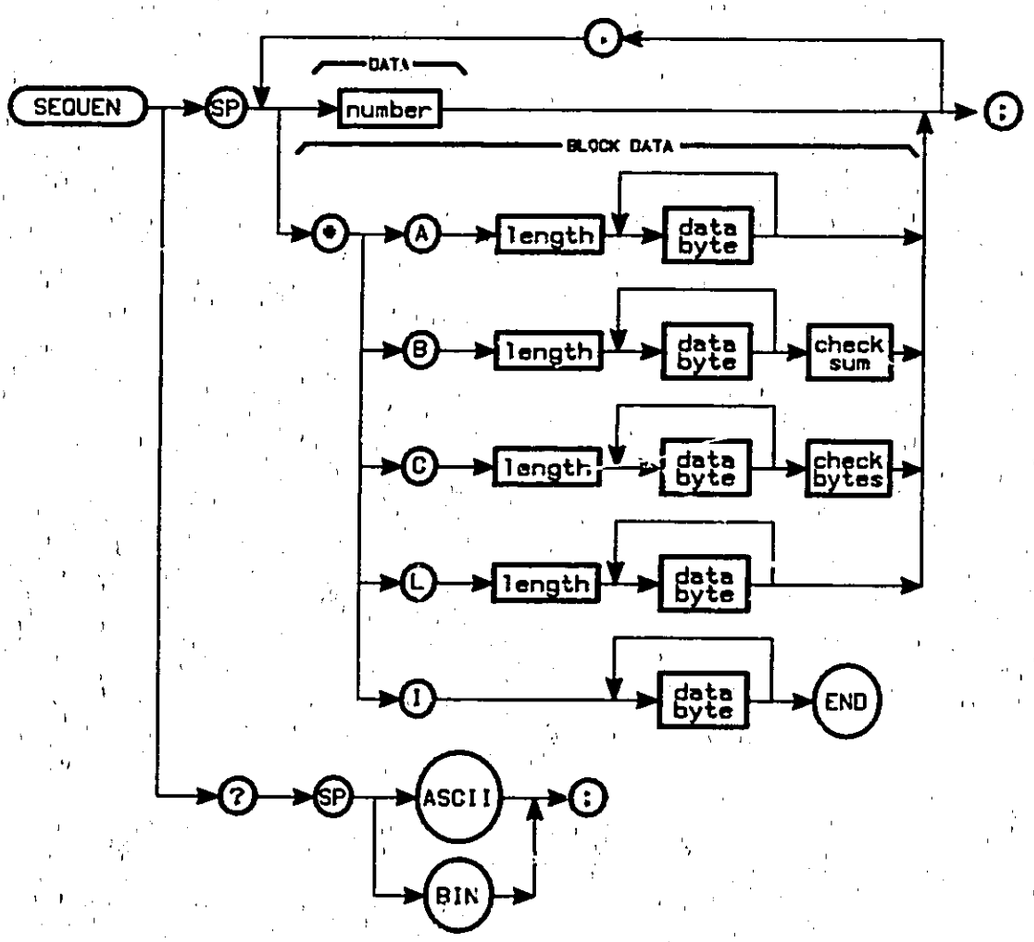
It is suggested that the SCALE command not be used in place of the output attenuator to reduce the signal power. SCALE is intended to be used for precision attenuation in differential amplitude applications. For example, to reduce a unity amplitude sine wave by 0.01 dB, scale by the factor 0.99884. To reduce the same wave by 0.1 dB, scale it by 0.98855.

The precision by which a wave segment can be scaled is a function of its amplitude. A sine wave at −40 dBm will not scale as precisely as one at 0 dBm. For waves between +10 and −10 dBm the precision of the scale is within 0.01 dB from ideal.

Related Sections

ATTEN
GO
WAVE

SEQUEN
(Load Sequencer Memory)



SEQUEN (cont'd)
(Load Sequencer Memory)

Item	Description	Range/Restrictions
Number	ASCII, decimal number. Represents a waveform data element.	−32 768 to 32 767. Data items must be separated from each other by a comma.
Length	16-bit (for A, B, and C blocks) or 32 bit (for L blocks) unsigned binary integer. Specifies number of data bytes to follow. Includes checksum for B-block data field and check bytes for C-block data field.	
Data Byte	8-bit byte representing sequencer data.	
Checksum	This single byte is the negative of the modulo 256 check sum of the data bytes. It does not include the length bytes.	
Check Bytes	Two bytes containing the remainder from division of the data byte stream by the CRC16 Forward polynomial.	

Description

The SEQUEN command loads sequencer memory with data. This command is used in conjunction with the SEQUEN? query. The sequencer data is formatted to contain all the necessary parameters to define the sequence. We recommend that instead of building the data array, obtain the data by using the SEQUEN? command.

SEQUEN? reads ASCII decimal or 16-bit binary data from the Synthesizer's sequencer memory. The sequencer memory contains four formatted, 16-bit words per packet. The four words contain a formatted version of the start address, length, scan, and packet advance mode variables.

The current sequence consists of a number of packets. The returned data string will be four times the number of packets in the current sequence in length. For example, if five packets are defined in the current sequence, 20 16-bit words will be returned if the sequencer memory is read in binary. Likewise, 20 decimal values will be returned if the sequencer memory is read in ASCII.

Example

Example 3-3, Copying Synthesizer Memory to Disc and Copying Disc to Synthesizer Memory, uses SEQUEN and SEQUEN?. Refer to the subroutine "COPY SEQUENCER DATA TO DISC (Seq_to_disc)" for an example of SEQUEN?. Refer to the subroutine "COPY SEQUENCER DATA FROM DISC TO AWS (Seq_to_aws)" for an example of SEQUEN.

Comments

The FORMAT command has no effect on data in sequencer memory.

Each packet requires 8 bytes of information. In block binary form, 8 bytes per packet must be sent. In ASCII decimal integer form, four 16-bit integers must be sent per packet.

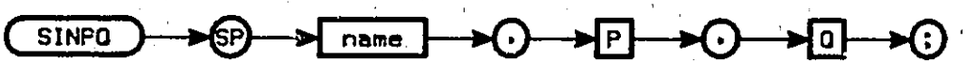
Refer to Data Input under HP-IB Remote Operation in this section for a discussion of block binary data transfers.

Related Sections

GO
PACKET
PACLIT

SINPQ

(Compute Sine Wave)



Item	Description	Range/Restrictions
Name	Waveform memory file name that conatins the sine data.	6 characters maximum
P	Represents number of cycles.	.0 to Q/2.5. Integer only.
Q	Represents number of elements	64 to 131 064. Must be a multiple of 8.

Description

The SINPQ command computes a 12-bit sine wave. This wave will be stored in waveform memory as a named file. The user specifies the cycles (P) and total number of elements (Q). The frequency of the sine wave is Fclock (P/Q), where Fclock is the effective sampling rate of the DAC data input.

The actual sine function computed and stored in waveform memory is

INT[2047 x SIN(2Pi(P/Q) x I)]+2048

where I ranges from 0 to Q−1 and Pi=3.1415926

This yields a peak-to-peak sine wave with −1 to +1 volts range at the RF output.

This function only computes and stores the 12-bit sine data. The user must define a sequence that uses the sine data file name.

Example

To generate a sine wave with one cycle of 1024 elements and to store the data in file SINX:

```
10 OUTPUT 719; "SINPQ SINX, 1, 1024"
20 OUTPUT 719; "PACKET SINX, 24, AUTO;GO"
30 END
```

Line 10: Creates and stores sine wave data in waveform memory file SINX.
Line 20: Creates packet using SINX and starts sequencer running.

Comments

The values of P and Q are used to obtain the rational value (P/Q) exactly. As Q increases, more values of P are legal value and more frequencies are available. Using a smaller Q, however, saves waveform memory. The frequency step resolution of the sine wave is not constant because both P and Q are variables. If a Qmax is selected then the number of different sine frequencies available may be closely approximated by

$$\frac{Qmax^2}{3}$$

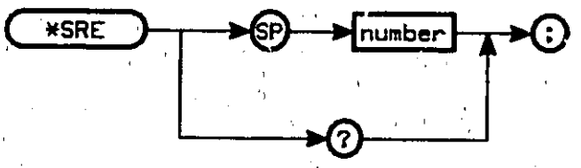
as Q: 64 to Qmax and P: 1 to Qmax/2.5

This function computes at a rate of 25 000 sine points per second.

Related Sections

GO
PACKET

*SRE
(Status Register Enable)



Item	Description	Range/Restrictions
Number	Represents value of bits in Status Register being enabled.	0 to 255

Description

The *SRE command sets the enable field for the Status Register.

Bits in the Status Register are logically ANDed with the corresponding bits in the enable field. If the result is one, bit 7 is set true and a Service Request (SRQ) is generated.

When queried (?), *SRE returns the current setting of the Status Register enable field.

The decimal value of each bit in the Status Register enable field is shown below.

Status Register Enable Field

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	Require Service	Events	Message Available	0	0	Operating Conditions	Hardware Errors
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Example

To enable hardware errors to generate an SRQ (that is, set bit 7 true):

```
OUTPUT 719;"*SRE 1"
```

Comments

Bits that have not been enabled by the *SRE command cannot generate a service request, regardless of whether or not they are true.

The Status Register enable field is set to zero at power-up.

Refer to Example 3-5 for an error checking program where command and execution errors generate an SRQ when they occur.

Related Sections

B1EN
B1EVT
B2EN
B2EVT
*ESE
*ESR
Programming the Status Register
*STB

*STB

(Read Status Register)

Description

*STB? queries the Synthesizer for the current value of the Status Register.

The Synthesizer returns a value between 0 and 255, representing the value of bits in the Status Register that are true.

The decimal value of each bit in the Status Register is shown below.

Status Register

Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1
0	Require Service	Events	Message Available	0	0	Operating Conditions	Hardware Errors
Value=128	Value=64	Value=32	Value=16	Value=8	Value=4	Value=2	Value=1

Example

To read the Status Register:

```
10 OUTPUT 719; "*STB?"
20 ENTER 719; A
30 PRINT "STATUS REGISTER ="; A
40 END
```

Comments

The *STB? command reads the Status Register but does not clear any bits, if they are set.

Power-up and program command *CLS set the Status Register to zero.

Related Sections

B1EN
B1EVT
B2EN
B2EVT
*CLS
*ESE
*ESR
Programming the Status Register
*SRE

STOP
(Stop Sequencer)

Description The STOP command stops the sequencer at the current packet if the sequencer was running. The RF output goes to 0 Vdc.

Example To stop the sequencer:

```
OUTPUT 719;"STOP"
```

Comments Both the STOP and OUTPUT OFF commands set the RF output to 0 Vdc. However, the STOP command stops the sequencer, whereas the OUTPUT OFF command does not.

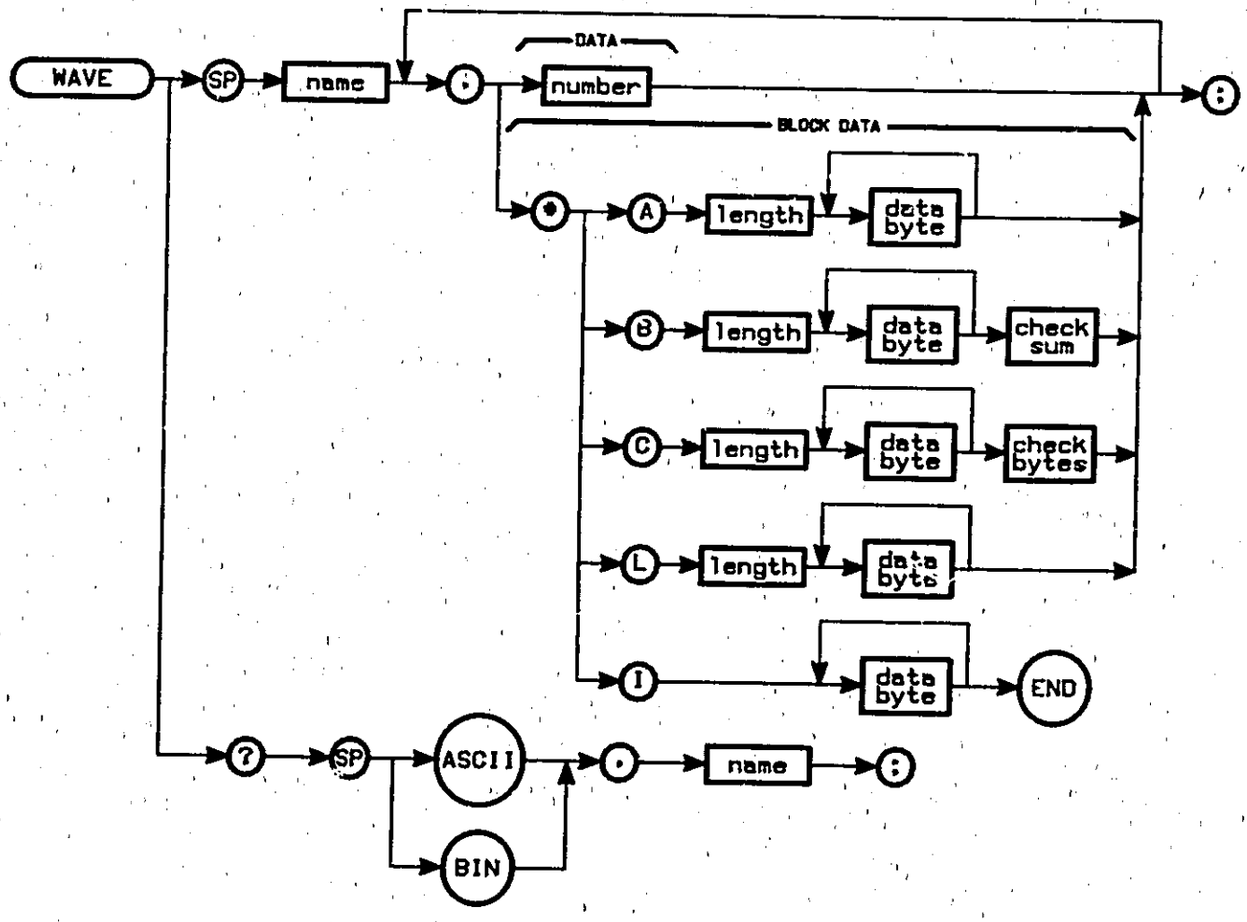
The following commands, when executed, stop the sequencer. The GO command must be issued to restart the sequencer.

BLKMOVE	EXDWM	SEQUEN
CLKDIV	GEN	SEQUEN?
CONST	LRNDIR	SINPQ
DIR? SEQ	PACKET	WAVE
DIR? WAV	PACLIT	WAVE?
EXDSEQ	PACLIT?	WAVEAP
EXDWAV	PURGE	WMEM
	SCALE	WMEM?

Related Sections
GO
OUTPUT
PACKET
PACLIT

WAVE

(Load Waveform Memory Using File Names)



WAVE (cont'd)

(Load Waveform Memory Using File Names)

Item	Description	Range/Restrictions
Name	Waveform memory file name where data is to be placed.	6 characters maximum
Number	ASCII, decimal number. Represents a waveform data element.	0 to 4095, unsigned format; −2048 to 2047, signed format. Numbers must be separated from each other by a comma
Length	16-bit (for A, B and C blocks) or 32 bit (for L blocks) unsigned binary integer. Specifies number of data bytes to follow. Includes checksum for B-block data field and check bytes for C-block data field.	
Data Byte	16-bit byte of data representing a waveform data element.	
Checksum	This single byte contains the negative of modulo 256 check sum of the data bytes. It does not include the length bytes.	
Check Bytes	Two bytes containing the remainder from division of the data byte stream by the CRC16 Forward polynomial.	

WAVE (cont'd)

(Load Waveform Memory Using File Names)

Description

The WAVE command loads waveform memory with data. The data is placed into the specified file name.

The length of the data string defines the length of the file. Before the file can be used in the PACKET command as part of a sequence, its length must have at least 56 elements and be a multiple of eight.

The data can be either ASCII decimal or block binary.

When queried (?), WAVE returns the contents of a specified waveform memory file. The data can be read in either ASCII decimal or 16-bit binary form. If the ASCII option is selected, the returned data values will have comma separators. If the binary option is selected, the first two bytes returned are #I (represented by binary characters) followed by 16-bit binary integers, representing waveform data elements. The last byte is sent with EOI. Thus, if this were read into a 16-bit integer array the first number would be 9033, which represents #I. The 9033 should be removed or ignored if the array is treated as data. The 9033 should remain the first number in the array if the array is being sent back to the Synthesizer.

Example

To load ASCII data into a waveform memory file named RAMP and then use this data in a sequence:

```
10 INTEGER A (1:1024), I
20 FOR I=1 TO 1024
30   A(I)=I
40 NEXT I
50 OUTPUT 719; "PURGE BOTH"
60 OUTPUT 719; "WAVE RAMP,"; A(*)
70 OUTPUT 719; "PACKET RAMP 1, AUTO;GO"
80 END
```

Line 10: Dimensions integer array A and declares integer variable I.
Lines 20—40: Creates data array.
Line 50: Purges both waveform and sequencer memory.
Line 60: Creates waveform memory file RAMP and loads it with data array.
Line 70: Creates sequence of 1 packet using wave segment RAMP and starts sequencer.

To load the array into waveform memory as binary data, replace line 60 with the following:

```
OUTPUT 719 USING "#, K, 1024(W)"; "WAVE RAMP, #I", A (*), END
```

Comments

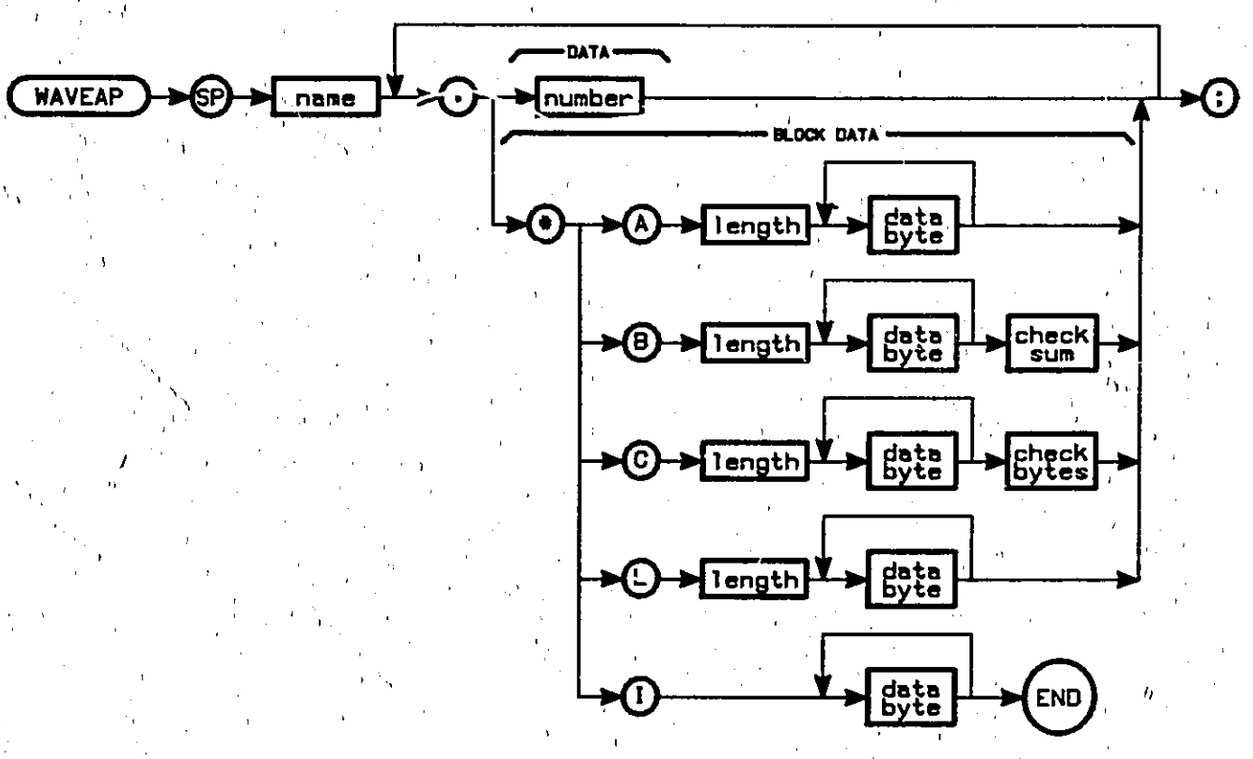
Block binary transfers load faster than ASCII transfers.

The numbering convention for data transferred to and from waveform memory is determined by the FORMAT command.

Related Sections

FORMAT
GO
PACKET
WAVEAP
WMEM

WAVEAP
(Append Waveform Data)



Item	Description	Range/Restrictions
Name	Waveform memory file name where data is to be placed. Must be the last file that was loaded.	6 characters maximum
Number	ASCII, decimal number. Represents a waveform data element.	0 to 4095, unsigned format; −2048 to 2047, signed format. Numbers must be separated from each other by a comma
Length	16-bit (for A, B and C blocks) or 32 bit (for L blocks) unsigned binary integer. Specifies number of data bytes to follow. Includes checksum for B-block data field and check bytes for C-block data field.	
Data Byte	16-bit byte of data representing a waveform data element.	
Checksum	This single byte contains the negative of modulo 256 check sum of the data bytes. It does not include the length bytes.	
Check Bytes	Two bytes containing the remainder from division of the data byte stream by the CRC16 Forward polynomial.	

WAVEAP (cont'd)
(Append Waveform Data)

Description

The WAVEAP command appends waveform data to the last named file that was loaded into waveform memory.

Example

To create a waveform where a sine wave is loaded into waveform memory and then additional data is appended to the wave segment:

```
10 INTEGER A (1:2048), I
20 FOR I=1 TO 2048
30    A(I)=I
40 NEXT I
50 OUTPUT 719; "PURGE SEQ"
60 OUTPUT 719; "SINPQ FREQ, 10, 1000"
70 OUTPUT 719; "WAVEAP FREQ,"; A(*)"
80 OUTPUT 719; "PACKET FREQ, 1, AUTO;GO"
90 END
```

Line 10: Dimensions integer array A and declares integer variable I.
Lines 20—40: Creates data array.
Line 50: Purges sequencer memory.
Line 60: Generates a sine wave and stores the data in a file named FREQ.
Line 70: Appends data array to FREQ
Line 80: Creates a sequence and starts sequencer running.

Comments

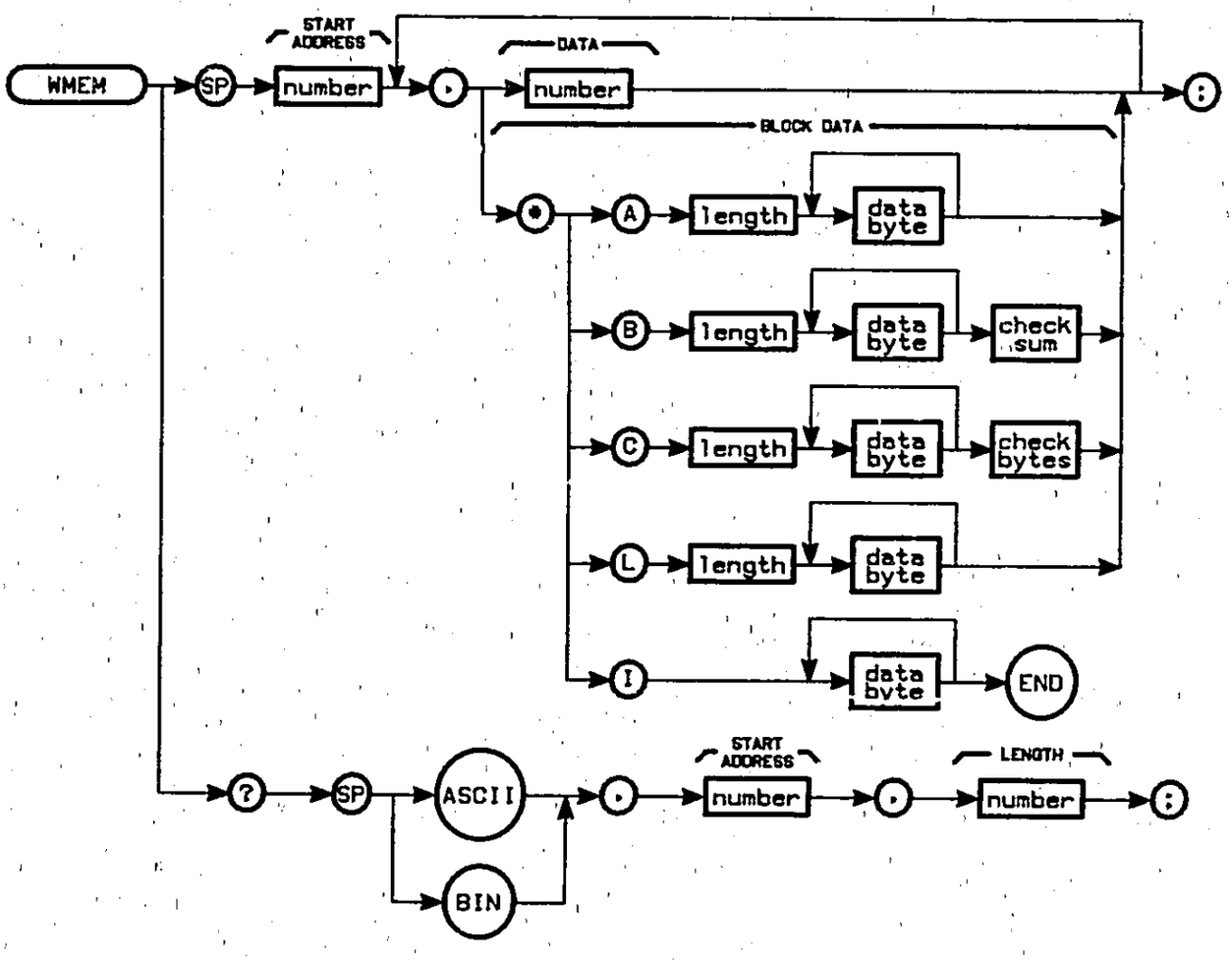
This command is useful whenever a data download is limited in length, for example, due to array dimension limits.

The numbering convention for data transferred to and from waveform memory is determined by the FORMAT command.

Related Sections

FORMAT
GO
PACKET

WMEM
(Load Waveform Memory Using Addresses)



WMEM

(Load Waveform Memory Using Addresses)

Item	Description	Range/Restrictions
Name	Waveform memory file name where data is to be placed.	6 characters maximum
Number	ASCII, decimal number. Represents a waveform data element.	0 to 4095, unsigned format; −2048 to 2047, signed format. Numbers must be separated from each other by a comma
Length	16-bit (for A, B and C blocks) or 32 bit (for L blocks) unsigned binary integer. Specifies number of data bytes to follow. Includes checksum for B-block data field and check bytes for C-block data field.	
Data Byte	16-bit byte of data representing a waveform data element.	
Checksum	This single byte contains the negative of modulo 256 check sum of the data bytes. It does not include the length bytes.	
Check Bytes	Two bytes containing the remainder from division of the data byte stream by the CRC16 Forward polynomial.	

Description

The WMEM command loads waveform memory with data. The data is placed in an unnamed record. The data can be either ASCII decimal or block binary.

When queried (?), WMEM returns the data in waveform memory. Data is read from the starting address for the length of words. The Synthesizer can send this data in either ASCII decimal or 16-bit binary. If the ASCII option is selected, the returned data values will have comma separators. If the binary option is selected, the first two bytes returned are #I (represented by binary characters) followed by 16-bit binary integers, representing waveform data elements. The last byte is sent with EOI. Thus, if this were read into a 16-bit integer array the first number would be 9033, which represents #I. The 9033 should be removed or ignored if the array is treated as data. The 9033 should remain the first number in the array if the array is being sent back to the Synthesizer.

Example

To load ASCII data into waveform memory and then use this data in a sequence:

```
10 INTEGER A (1:1024), I
20 FOR I=1 TO 1024
30   A(I)=I
40 NEXT I
50 OUTPUT 719; "PURGE BOTH"
60 OUTPUT 719; "WMEM 0,"; A(*)
70 OUTPUT 719; "PACLIT 0, 1024, 64, AUTO; GO"
80 END
```

WMEM (cont'd)

(Load Waveform Memory Using Addresses)

Example (cont'd)

Line 10: Dimensions integer array A and declares integer variable I.
Lines 20—40: Creates data array.
Line 50: Purges both waveform and sequencer memory.
Line 60: Loads the data array into waveform memory, starting from address 0.
Line 70: Creates a sequence of one packet and starts sequencer.

To enter the data into waveform memory in binary rather than ASCII, replace line 60 with the following:

```
OUTPUT 719 USING "#, K, 1024(W)"; "WMEM 0, #I", A(*), END
```

Comments

This command should be used when the user wants to work directly with waveform memory and not refer to it by file names.

For either the binary or the ASCII option, the returned data values can be signed or unsigned. Refer to the FORMAT command.

Block binary transfers load faster than ASCII transfers.

Related Sections

FORMAT
GO
PACLIT
WAVE

SECTION IV
PERFORMANCE TESTS

4-1. INTRODUCTION

The procedures in this section test the electrical performance of the Arbitrary Waveform Synthesizer using the specifications of Table 1-1 as performance standards. All tests can be performed without access to the interior of the instrument. Where equipment and time is limited, such as On-Site testing, certain tests and equipment can be eliminated. The performance tests are separated into three different categories:

Abbreviated Performance Test

- **Operational Verification** — this test provides a quick check of the major warranted specifications. It requires less time and equipment than the full performance tests. This test should be performed after normal repairs, for incoming inspection or used after On-Site repairs.

Scheduled Calibration

- **Performance Test** — this test provides assurance that the instrument meets all its warranted specifications and performs its non-warranted functional checks. A Performance Test Record is provided to record the test results. This test requires more time and equipment than Operation Verification in order to achieve its higher confidence level, and hence may not be available On-Site. For On-Site availability contact your nearest Hewlett-Packard Sales Office.

- **Automated Performance Test** — automated performance tests can save test time and provide consistent test procedures from instrument to instrument. Automated performance test software and manuals for specific Hewlett-Packard computers are ordered separately from your nearest Hewlett-Packard Sales Office. For your convenience, a tab has been provided in the Operating and Service manual to add the Automated Performance Test procedures.

4-2. EQUIPMENT REQUIRED

Equipment required for the Performance Tests or Operation Verification is listed in Table 1-3, Recommended Test Equipment. Any equipment that satisfies the critical specifications given in the table may be substituted for the recommended models, except where noted. All of the tests require the use of an HP-IB compatible controller to control the Synthesizer. Program code listings in HP BASIC are provided. Some of the programs, however, may have to be modified if a controller other than the one recommended is used.

4-3. PERFORMANCE TEST RECORD

Results of Operation Verification and the Performance Tests may be recorded in Tables 4-2 and 4-3 respectively. These tables are located at the end of this section and list all of the tested specifications and their acceptable limits. Results recorded at incoming inspection can be used for comparison in periodic maintenance, troubleshooting and after repairs or adjustments.

4-4. CALIBRATION CYCLE

This instrument requires periodic verification of performance. Depending on the use and environmental conditions, the instrument should be checked using the performance tests at least once a year. In the case where a Hewlett-Packard automated performance test is available, the automated version may be substituted for the manual performance test in this manual.

4-5. ABBREVIATED PERFORMANCE TEST (OPERATION VERIFICATION)

This test provides a quick check of the major warranted specifications. It requires less time and equipment than the full performance tests. This test should be performed after normal repairs, for incoming inspection or used after On-Site repairs. It will provide a 90% confidence level that the instrument meets its specifications and functional checks. Perform the tests in the order given and record the data in Table 4-2, Operation Verification Test Record or in the data spaces provided within the test procedures.

NOTE

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

ABBREVIATED PERFORMANCE TEST (OPERATION VERIFICATION) (cont'd)

The Synthesizer must have a 15 minute warm-up period to allow the internal 10 MHz reference oscillator to stabilize.

It is assumed that the person performing the tests understands how to use the specified test equipment. Equipment settings, other than those for the Synthesizer, are stated in general terms. It is also assumed that the person performing the tests will supply whatever cables, connectors and adapters are necessary.

Preliminary Set-Up

1. Connect an HP-IB compatible controller to the Synthesizer's rear panel HP-IB connector. Load the test program, shown in Figure 4-1, into the controller. This program is used to set different frequencies and output levels needed during the performance tests.
2. After loading the program, store it on cassette or disc for future use.

NOTE

This program listing is in HP technical BASIC. If you have any other controller, you will need to change the program to your controller's operating language.

When transposing the program to other controller languages, the symbol "!" represents the "REM" statement in BASIC. Remarks follow the "!" symbol. Remarks need not be entered for the program to run and are for documentation only.

3. The program will allow you to control the Synthesizer to output several different sine waves with the following frequencies and levels:

Test Program Selections	
Sine wave Frequencies	Output Levels
100 kHz 1 MHz 10 MHz 25 MHz 30 MHz 50 MHz	With the Synthesizer attenuator set to 0 dB, output of all sine waves will typically be +10 dBm ±1.0 dB. Attenuation can then be selected in the range of 0 to 110 dB in 10 dB steps.

Since the Synthesizer has no manual controls, this program will make it relatively easy to set the Synthesizer to any of the test conditions needed during the performance tests.

```
10 ! (REM) HP 8770A TEST PROGRAM
20 ! (REM) FILE NAME"AWS"
30 ! ********************************************************
40 ! (REM) CLEAR THE DISPLAY AND DISPLAY ENTER MESSAGE
50 CLEAR @ DISP @ DISP "ENTER A 1 OR 2 THEN PRESS ENTER"
60 DISP @ DISP "1 = SET FREQUENCY" @ DISP "2 = SET ATTENUATION"
70 DISP @ DISP "CHOOSE ONE";
80 INPUT A
90 ! ********************************************************
100 ! (REM) IF USER ENTERS 1 THEN GOTO FREQUENCY ROUTINE
110 ! (REM) IF 2 THEN GOTO ATTENUATION ROUTINE
120 IF A=1 THEN GOTO 270
130 ! ********************************************************
140 ! CHANGE ATTENUATION ROUTINE
145 CLEAR
150 DISP @ DISP "ENTER ATTENUATION DESIRED THEN  PRESS ENTER"
160 DISP @ DISP "YOUR CHOICES ARE 0, 10, 20, UP  TO 110 in 10 DB STEPS"
170 DISP "ENTER 200 TO EXIT"
180 DISP @ DISP "WHICH DO YOU CHOOSE";
190 !
200 INPUT B !
210 IF B=200 THEN GOTO 50
220 IF B<0 OR B>110 THEN B=INT(B/10)*10
230 A$="ATTEN "&VAL$(B)
240 OUTPUT 719 ;A$
245 DISP "ATTENUATION IN dB SET FOR ";B
250 GOTO 150
260 ! ********************************************************
270 ! (REM) ROUTINE TO SET THE FREQ
275 CLEAR
280 DISP @ DISP "ENTER .1, 1, 10, 25, 30, OR 50  THEN PRESS ENTER"
290 DISP "ENTER 0 TO EXIT"
300 DISP @ DISP "CHOOSE ONE";
310 INPUT C
320 !
330 ! (REM) EVALUATE "C" AND GOTO STATEMENT TO SET CORRECT FREQUENCY
340 IF C=.1 THEN GOTO 450
350 IF C=1 THEN GOTO 470
360 IF C=10 THEN GOTO 490
370 IF C=25 THEN GOTO 510
380 IF C=30 THEN GOTO 530
390 IF C=50 THEN GOTO 550
400 IF C=0 THEN GOTO 50
410 ! ********************************************************
420 ! (REM) USING THE SYNTHESIZER 'SINPQ' COMMAND A SINE WAVE IS PRODUCED
430 ! (REM) EACH STATEMENT PRODUCES A DIFFERENT FREQUENCY SINE WAVE
440 !
450 OUTPUT 719 ;"SINPQ SINA,8,10000" @ GOTO 560
460 !
470 OUTPUT 719 ;"SINPQ SINA,8,1000" @ GOTO 560
480 !
490 OUTPUT 719 ;"SINPQ SINA,64,800" @ GOTO 560
500 !
510 OUTPUT 719 ;"SINPQ SINA,160,800" @ GOTO 560
520 !
530 OUTPUT 719 ;"SINPQ SINA,192,800" @ GOTO 560
540 !
550 OUTPUT 719 ;"SINPQ SINA,320,800"
560 OUTPUT 719 ;"PACKET SINA,1,AUTO" @ OUTPUT 719 ;"GO"
565 DISP "FREQUENCY IN MHz SET TO ";C
570 GOTO 280
580 !
590 END
```

Figure 4-1. Test Program Listing

4-6. OPERATION VERIFICATION

Parameters Checked	Performance Limits	Conditions
Power-Up Diagnostic	Pass/Fail	On-Power-Up
10 MHz REFERENCE OUTPUT		
Output Level	>1 volt peak-to-peak, > 0 dBm	
RF OUTPUT LEVEL		
Output Voltage	>1 volt peak-to-peak,	
Peak Output Power		
Standard: 50Ω Output	+10 dBm ± 0.25 dB	Into 50Ω
Option 002: 75Ω Output	+8.24 dBm ± 0.25 dB	Into 75Ω
Harmonics		
Second	<−50 dBc	
Third	<−50 dBc	
Fourth	<−50 dBc	
Spurious Signals	<−50 dBc	
Attenuator Switching Function for 0—80 dB Positions	Each attenuator pad is checked for its approximate specified value.	Referenced at attenuator setting of 0 dB
Rear Panel Input/Output Functional Checks	Each output is checked for proper operation	

Equipment

Oscilloscope	HP 1980B
Spectrum Analyzer	HP 8566B
Power Meter	HP 436A
Controller	HP 9000 Series 200 Model 236
Series 200 BASIC Language	HP 98613A
Frequency Counter	HP 5343A
Power Sensor	HP 8482A
75Ω to 50Ω Adapter	HP 11687A

Procedure

Power-Up Diagnostic

1. Check that the Synthesizer has been properly installed using the procedures in Section II, Installation, of this manual. Ensure that the proper line voltage and fuse have been selected and installed.

2. Plug in the line (mains) power to the Synthesizer and set the LINE switch to the ON position. Observe the front panel LEDs for the first 15 seconds after turn-on. Internal diagnostics are run during this period. The LEDs should display the following pattern:
   - All LEDs flash on for approximately 1 second.
   - All LEDs turn off for about 3 seconds.
   - Each LED is turned on one at a time in succession.

OPERATION VERIFICATION

Procedure (cont'd)

Power-Up Diagnostic (cont'd)

- All LEDs are flashed on simultaneously.
- All LEDs except IDLE are turned off.

Power-Up Diagnostic _____ (√)

3. Completion of the above pattern indicates that the Synthesizer passed all of its internal diagnostics. If any other pattern occurs and/or the FAULT LED stays on, a failure has occurred. If a failure has occurred now or does occur in any of the following tests, go to Block Diagram 1 in Section VIII and begin troubleshooting or refer servicing to a qualified technician.

10 MHz Reference Output Level

4. Connect the rear panel output, 10 MHz REFERENCE OUTPUT 1, to an oscilloscope. Set the oscilloscope to DC coupled. The level of the sine wave displayed should be greater than 1.0 volt peak-to-peak. Record the peak-to-peak level, then disconnect the oscilloscope.

10 MHz Output Level _____ volt(s) p-p

NOTE

For the next step, ensure that the power meter has been zeroed, calibrated and that the CAL FACTOR has been set for a frequency of 10 MHz.

5. Connect a power meter to the 10 MHz REFERENCE OUTPUT 1. The output power should be >0 dBm. Record the output power then disconnect the power meter.

10 MHz Output Power _____ dBm

6. Remove the small gray jumper cable, on the rear of the instrument, from the connector labeled 10 MHz REFERENCE OUTPUT 2. Connect an oscilloscope to the 10 MHz REFERENCE OUTPUT 2 connector. The output should be >1 volt peak-to-peak. Record the peak-to-peak level then disconnect the oscilloscope.

10 MHz Output Level _____ volt(s) p-p

7. Connect a power meter to the 10 MHz REFERENCE OUTPUT 2 connector. The power level should be >0 dBm. Record the output power. Disconnect the power meter. Reconnect the cable between 10 MHz REFERENCE OUTPUT 2 and the 10 MHz REFERENCE INPUT.

10 MHz Output Power _____ dBm

RF Output Level

8. Ensure that the Synthesizer is connected to an HP-IB compatible controller and that the Synthesizer address is set to "719," (refer to Section II, Installation). Connect an oscilloscope to the rear panel RF OUTPUT connector on the Synthesizer. Set the oscilloscope to DC coupled, 1 volt/division and a time base of 0.05 μs.

9. Run the test program shown in Figure 4-1. When the question appears "Enter 1 for Frequency, 2 for Attenuation," enter a 1 then press EXECUTE on controller.

OPERATION VERIFICATION

Procedure (cont'd)

RF Output Level (cont'd)

You will now be able to select a frequency. Enter "10" then press EXECUTE. This will program the Synthesizer to output a 10 MHz sine wave.

10. Now enter "0" to exit the frequency mode. Enter "2" to select the attenuation mode and another "0" to select 0 dB of attenuation. The Synthesizer should now be outputting a 10 MHz sine wave at about +10 dBm.

    The oscilloscope should display a sine wave whose level should be greater than 1.0 volt peak-to-peak. Record the peak-to-peak level then disconnect the oscilloscope from the RF OUTPUT connector.

    RF OUTPUT Level _____ volt(s) p-p

    For the standard Synthesizer (50Ω RF OUTPUT), perform the following step. For Option 002 (75Ω RF OUTPUT), skip to step 12.

11. Connect a power meter to the RF OUTPUT connector. The measured power should be +10 dBm ±0.25 dB. Record the RF OUTPUT power level.

    RF OUTPUT Power Level _____ dBm
    (Standard Instrument)

12. Connect the 75Ω to 50Ω adapter to the RF OUTPUT connector. The 75Ω end of the adapter should be connected to the Synthesizer. Connect the 50Ω end of the adapter to the power meter. Record the power meter reading and correct it by the loss of the adapter. The specified corrected power is +8.24 dBm ±0.25 dB.

    Power Meter Reading _____ + (1.76 dB) = _____ dBm
    (adapter correction) (corrected level)

NOTE

1.76 dB is the amount of power lost through the adapter.

If the measured values are not as stated, go to Section V and perform the DAC Absolute Power and DC Offset Adjustment. After making the adjustment repeat steps 10, 11 and 12.

Harmonics and Spurious Signals

13. Disconnect the power meter from the RF OUTPUT connector. Connect a spectrum analyzer to the RF OUTPUT connector. Make the following settings on the spectrum analyzer:

    Center Frequency: 10 MHz
    Frequency Span: 100 MHz (10 MHz/div)
    Reference Level: Set the 10 MHz carrier peak to the top line of the display.

14. Tune the spectrum analyzer so that the 10 MHz carrier is on the left hand side of the display and the 2nd, 3rd and 4th harmonics are visible. The harmonics and spurious signals of the 10 MHz carrier should be as shown on the following page:

OPERATION VERIFICATION

Procedure (cont'd)

Harmonics and Spurious Signals (cont'd)

Harmonic	Level	Signal	Level
2nd (20 MHz) 3rd (30 MHz) 4th (40 MHz)	<−50 dBc <−50 dBc <−50 dBc	Spurious	<−50 dBc

Record the level of each harmonic and the level of the highest spurious signal.

Second Harmonic _____ dBc

Third Harmonic _____ dBc

Fourth Harmonic _____ dBc

Spurious Signal _____ dBc

If the spurious signals are out of specification, go to Section V and perform the Sampler Delay Adjustment. After making the adjustment, repeat the spurious signal measurement in step 14.

Attenuator Switching Function

15. Adjust the spectrum analyzer so that the 10 MHz carrier is in the center of the display. Set the signal peak on the top line of the display. This will act as the 0 dB reference for the other attenuator measurements.

16. Using the test program, set the Synthesizer to 10 dB of attenuation. The level on the spectrum analyzer should be approximately −10 dB below the 0 dB reference set in the previous step.

    Signal Was Decreased By 10 dB _____ (√)

17. Increase the Synthesizer attenuation in 10 dB steps from 20 to 80 dB while observing the spectrum analyzer. Ensure that the signal level steps down approximately 10 dB at each Synthesizer attenuator setting. This ensures that each attenuator pad is functioning properly.

NOTE

The reference level of the spectrum analyzer may have to be adjusted in order to measure the lowest signal level at 80 dB of attenuation.

Synthesizer Attenuator Setting (dB)	Attenuation Verified
20	_____ (√)
30	_____ (√)
40	_____ (√)
50	_____ (√)
60	_____ (√)
70	_____ (√)
80	_____ (√)

OPERATION VERIFICATION

Procedure (cont'd)

Rear Panel Input/Outputs

18. Perform the Rear Panel Input/Output Functional Checks in paragraph 4-12, starting at step 2 and continuing through step 14. Delete the External Data Input Check.

PERFORMANCE TESTS

4-7. PERFORMANCE TEST PROCEDURES

Perform the tests in the order given and record the data in Table 4-3, Performance Test Record or in the data spaces provided within the test procedures.

NOTE

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

The Synthesizer must have a 15 minute warm-up period to allow the internal 10 MHz reference oscillator to stabilize.

It is assumed that the person performing the tests understands how to use the specified test equipment. Equipment settings, other than those for the Synthesizer, are stated in general terms. It is also assumed that the person performing the tests will supply whatever cables, connectors and adapters are necessary.

Preliminary Set-Up

1. Connect an HP-IB compatible controller to the Synthesizer's rear panel HP-IB connector. Load the test program, shown in Figure 4-1, into the controller. This program is used to set different frequencies and output levels needed during the performance tests.

2. After loading the program, store it on cassette or disc for future use.

NOTE

This program listing is in HP technical BASIC. If you have any other controller, you will need to change the program to your controller's operating language.

When transposing the program to other controller languages, the symbol "!" represents the "REM" statement in BASIC. Remarks follow the "!" symbol. Remarks need not be entered for the program to run and are for documentation only.

PERFORMANCE TESTS

4-8. POWER-UP DIAGNOSTIC

Description This test does not require any external test equipment to determine the pass or fail results. If the test fails, the front panel FAULT LED will turn on.

Equipment None Required

Procedure

1. Plug in the line (mains) power to the Synthesizer and set the LINE switch to the ON position. Observe the front panel LEDs for the first 15 seconds after turn-on. Internal diagnostics are run during this period. The LEDs should display the following pattern:
   - All LEDs flash on for approximately 1 second.
   - All LEDs turn off for about 3 seconds.
   - Each LED is turned on one at a time in succession.
   - All LEDs are flashed on simultaneously.
   - All LEDs except IDLE are turned off.
2. Completion of the above pattern indicates that the Synthesizer passed all of its internal diagnostics. If any other pattern occurs and/or the FAULT LED stays on, a failure has occurred. If a failure has occurred now or does occur in any of the following tests, go to Block Diagram 1, Section VIII of the Operating and Service Manual, unless otherwise instructed, and begin troubleshooting or refer servicing to a qualified technician.

Power-Up Diagnostic _____ (√)

PERFORMANCE TESTS

4-9. SINE WAVE OUTPUT POWER AND ATTENUATOR ACCURACY

Specification

Electrical Specifications	Performance Limits	Conditions
Sine Wave Performance Output Power	+10 dBm ±0.25 dB (Standard; into 50Ω) +8.24 dBm ±0.25 dB (Option 002; into 75Ω)	Referenced at 10 MHz, 0 dB of attenuation
Internal Attenuator	Attenuator Setting (dB) / Accuracy (dB) 10 ±0.2 20 ±0.4 30 ±0.5 40 ±0.8 50 ±0.9 60 ±1.0 70 ±1.2 80 ±1.4 90 ±1.5 100 ±1.6 110 ±1.8	Referenced at attenuator setting of 0 dB for all frequencies from dc to 50 MHz.

Description

This test uses a power meter to check the Synthesizer's output power at 10 MHz with a 0 dB attenuator setting. This measurement point is then recorded and used as a reference for the 10, 20 and 30 dB attenuator accuracy measurements. The relative error of the specified attenuator setting from the reference is then recorded.

A spectrum analyzer is used to check the relative errors of the 40 through 110 dB attenuator settings. In this portion of the test 110 dB of attenuation is inserted in line with the Synthesizer's RF OUTPUT connector, followed by 40 dB of amplification. A 0 dB reference is set. The 'Attenuation Substitution Method" is then used to measure the Synthesizer's attenuator accuracy.

In this method the Synthesizer's attenuation is increased in 10 dB steps while the external step attenuator is reduced by 10 dB steps. For each attenuator setting the relative error from the reference line is read. This error is then recorded for each attenuator setting.

Equipment

Power Meter	HP 436A
Power Sensor	HP 8482A
Spectrum Analyzer	HP 8566B
Two 20 dB Amplifiers	HP 8447A
110 dB Step Attenuator	HP 8496A
HP-IB Compatible Controller	HP 9000 Series 200 Model 236
Series 200 BASIC Language	HP 98613A
For Option 002	
75Ω to 50Ω Adapter	HP 11687A

PERFORMANCE TESTS

SINE WAVE OUTPUT POWER AND ATTENUATOR ACCURACY (cont'd)

Procedure **Output Level and High Level Attenuation Accuracy (0—30 dB)**

1. Ensure that the Synthesizer is connected to an HP-IB compatible controller. Run the test program. When the question appears "Enter 1 for Frequency, 2 for Attenuation," enter a 1 then press EXECUTE on the controller. You will now be able to select a frequency. Enter "10" then press EXECUTE. This will program the Synthesizer to output a 10 MHz sine wave.

2. Now enter "0" to exit the frequency mode. Enter "2" to select the attenuation mode and another "0" to select 0 dB of attenuation. The Synthesizer should now be outputting a 10 MHz sine wave at about +10 dBm.

NOTE

For the next step, ensure that the power meter has been zeroed, calibrated and that the CAL FACTOR has been set for a frequency of 10 MHz.

For a standard instrument (50Ω RF OUTPUT), perform the following step. For Option 002 (75Ω RF OUTPUT), skip to step 4.

3. Connect the power meter to the Synthesizer's RF OUTPUT. The power meter should read +10 dBm ±0.25 dB. Record the power reading.

Power Out at 10 MHz _____ dBm
(Standard Instrument)

4. Connect the 75Ω to 50Ω adapter to the RF OUTPUT connector. The 75Ω end of the adapter should be connected to the Synthesizer. Connect the 50Ω end of the adapter to the power meter. Record the power meter reading and correct it by the loss of the adapter. The specified corrected power is +8.24 dBm ±0.25 dB.

Power Meter Reading _____ + (1.76 dB) = _____ dBm
(correction factor) (corrected level)

NOTE

1.76 dB is the amount of power lost through the adapter.

If the measured values are not as stated, go to Section V and perform the DAC Absolute Power and DC Offset Adjustments. After making the adjustments, repeat steps 3 and 4.

5. Set frequency to 0.1 MHz with the test program (you will have to exit the attenuation mode). Record the power meter reading or set the power meter to the dB REL (relative) mode if available on your power meter. This will establish the 0 dB reference point.

6. Change the Synthesizer's attenuator setting to 10 dB using the test program. Verify that the power meter reads −10 dB ±0.2 dB from the 0 dB reference. Record only the + or − error from the reference (for example, −0.2 dB or +0.1 dB error).

Relative Error at 10 dB of attenuation _____ dB

PERFORMANCE TESTS

SINE WAVE OUTPUT POWER AND ATTENUATOR ACCURACY (cont'd)

Procedure (cont'd)

Output Level and High Level Attenuation Accuracy (0—30 dB) (cont'd)

7. Change the Synthesizer to 20 dB of attenuation. Verify that the power meter reads −20 dB ± 0.4 dB from the 0 dB reference. Record the + or − error from the specified attenuator value.

Relative Error at 20 dB of attenuation _____ dB

8. Change the Synthesizer to 30 dB of attenuation. Verify that the error is less than ± 0.5 dB from the specified value for 30 dB. The power meter will take approximately 10 seconds to settle. Record the + or − error from the specified value.

Relative Error at 30 dB of attenuation _____ dB

Low Level Attenuation Accuracy (40—110 dB)

9. Disconnect the power sensor from the Synthesizer's RF OUTPUT. Connect the equipment as shown in Figure 4-2.

CAUTION

Before connecting the signal to the spectrum analyzer, ensure the step attenuator is set to 110 dB of attenuation. This will protect the amplifiers and spectrum analyzer inputs from overload.

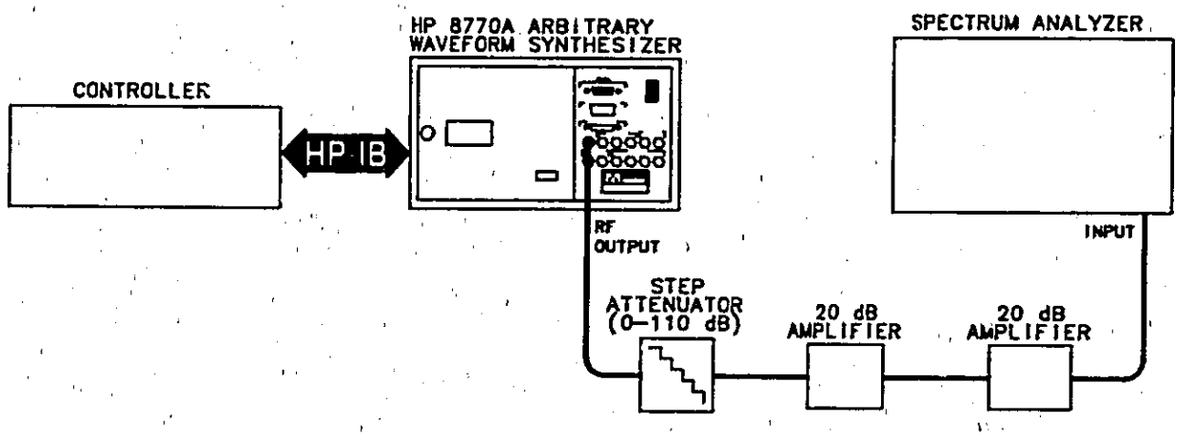


Figure 4-2. Low Level Attenuation Accuracy Test Setup

Make the following settings on the Synthesizer and spectrum analyzer:

Synthesizer	Spectrum Analyzer
Attenuation ..... 0 dB	Input Attenuation .....≤ 10 dB Carrier ..... 0.1 MHz (Place signal at center of display.) Frequency Span/BW ..... For a clear display Vertical Gain ..... 1 dB/division; (Place signal at center horizontal line.)

PERFORMANCE TESTS

SINE WAVE OUTPUT POWER AND ATTENUATOR ACCURACY (cont'd)

Procedure (cont'd)

Low Level Attenuation Accuracy (40—110 dB) (cont'd)

NOTE

For best spectrum analyzer display, set the top reference line for approximately −50 dBm, VIDEO FILTER on, INPUT ATTENUATION to 10 dB or less, and RESOLUTION BANDWIDTH to 100 kHz or less.

10. With the signal approximately at the center of the display, adjust the vertical gain vernier to set the 0.1 MHz signal exactly on the center horizontal line of the display. This will be used as the 0 dB reference for all other measurements.

11. Set the Synthesizer's attenuator to 40 dB using the test program. Set the step attenuator to 70 dB of attenuation.

CAUTION

Reducing the external step attenuator setting by more than the change in the Synthesizer's attenuator value could result in damage to the 40 dB amplifiers or spectrum analyzer. When changing attenuation values, always ensure that the Synthesizer's attenuator setting is increased first before the external step attenuator is decreased.

12. Record the plus or minus error of the carrier level from the reference. The error should be less than ±0.8 dB from the reference.

Relative Error at 40 dB of attenuation _____ dB

NOTE

The actual attenuation value rather than the + or − error can be determined for each setting, if desired, by using the following procedure:

If the signal level is above the reference line, then the actual attenuation is less than the specified value by the error amount. If the signal level is below the reference, then the attenuation is more than the specified value by the error amount.

13. Increase the Synthesizer's attenuation in 10 dB steps while decreasing the external step attenuator by the same amount. Record the + or − error from the specified value, for each attenuator setting from 50 to 110 dB, in Table 4-1:

Table 4-1. Level Accuracy Attenuation Error

Synthesizer Attenuation	External Step Attenuator	Specification	Measured ± Error from Reference
0 dB	110 dB	± 0 dB	Reference
50 dB	60 dB	±0.9 dB	_____ dB
60 dB	50 dB	±1.0 dB	_____ dB
70 dB	40 dB	±1.2 dB	_____ dB
80 dB	30 dB	±1.4 dB	_____ dB
90 dB	20 dB	±1.5 dB	_____ dB
100 dB	10 dB	±1.6 dB	_____ dB
110 dB	0 dB	±1.8 dB	_____ dB

PERFORMANCE TESTS

4-10. SINE WAVE HARMONICS, SPURIOUS AND NON-HARMONIC DISTORTION

Specification

Electrical Specifications	Performance Limits	Conditions
SINE WAVE PERFORMANCE		Harmonics and spurious referenced to +10 dBm while using the 125 MHz internal clock.
Harmonics	<−50 dBc	Carrier ≤10 MHz
	<−40 dBc	For carrier frequencies >10 MHz to 50 MHz
Spurious and Nonharmonic Distortion	<−50 dBc	Carrier frequencies ≤25 MHz
	<−40 dBc	Carrier frequencies >25 MHz to 50 MHz

Description

This group of tests uses a spectrum analyzer to check the spectral purity when the Synthesizer is outputting a sine wave. These checks are performed with an output power level of +10 dBm and frequencies of 1, 25 and 30 MHz.

Equipment

Spectrum Analyzer ..... HP 8566B
Power Meter ..... HP 436A
HP-IB Compatible Controller ..... HP 9000 Series 200 Model 236
Series 200 BASIC Language ..... HP 98613A

Procedure

1. Connect the Synthesizer's RF OUTPUT to the spectrum analyzer.
2. Set the Synthesizer to 1 MHz using the test program. Set the output level to +10 dBm (0 dB of attenuation).
3. Tune the spectrum analyzer to display the carrier (1 MHz). Set the frequency span wide enough to see the 4th harmonic of the carrier. Set the reference level of the spectrum analyzer so the carrier is at the top line of the display.
4. Check the level of the 2nd, 3rd and 4th harmonics. The harmonics should be <−50 dBc. Record the levels below.

Harmonic		
Second	_____	<−50 dBc
Third	_____	<−50 dBc
Fourth	_____	<−50 dBc

5. Change the Synthesizer output frequency to 25 MHz using the test program. Retune the spectrum analyzer to display the harmonics and spurious signals. Record the levels of the harmonics and worst case spurious signal below.

Harmonic			Spurious Signal	
Second	_____	<−40 dBc	_____	<−50 dBc
Third	_____	<−40 dBc		
Fourth	_____	<−40 dBc		

PERFORMANCE TESTS

SINE WAVE HARMONICS, SPURIOUS AND NON-HARMONIC DISTORTION (cont'd)

Procedure (cont'd)

If the spurious signals are out of specification, go to Section V and perform the Sampler Delay Adjustment. After making the adjustment, repeat the measurement in step 5.

6. Change the Synthesizer output frequency to 30 MHz using the test program. Retune the spectrum analyzer to display the harmonics and spurious signals. Record the levels of the harmonics and worst case spurious signal below.

Harmonic		Spurious Signal	
Second	_____ <−40 dBc	_____	<−40 dBc
Third	_____ <−40 dBc		
Fourth	_____ <−40 dBc		

PERFORMANCE TESTS

4-11. 10 MHz REFERENCE OSCILLATOR AGING RATE

Specification

Electrical Specifications	Performance Limits	Conditions
REFERENCE OSCILLATOR Aging Rate	$<5 \times 10^{-10}$/day	After 24 hour warm-up and an oscillator off-time of less than 24 hours.

Description

The 10 MHz REFERENCE OUTPUT is connected to the oscilloscope's vertical input. A frequency standard (with long term stability greater than 1×10^{-10} is connected to the oscilloscope's trigger input. The time required for a specific phase change is measured immediately and after a period of time. The aging rate is inversely proportional to the absolute value of the difference of the measured times.

Equipment

Frequency Standard ..... HP 5065A
Oscilloscope ..... HP 1980B
50Ω Termination ..... HP 11593A

NOTE

The 10 MHz oscillator will typically take 24 to 48 hours to reach its specified aging rate after instrument storage or shipment. In some cases, if extreme environmental conditions were encountered during storage, the reference could take up to one week to achieve its specified aging rate.

Procedure

1. Connect the Synthesizer's rear panel 10 MHz REFERENCE OUTPUT 1 to channel A of the oscilloscope. Use the 50Ω termination to load the output, (see Figure 4-3 below). Adjust the oscilloscope's time per division to 0.1 μs/division.

2. Connect the frequency standard to the oscilloscope's external trigger input.

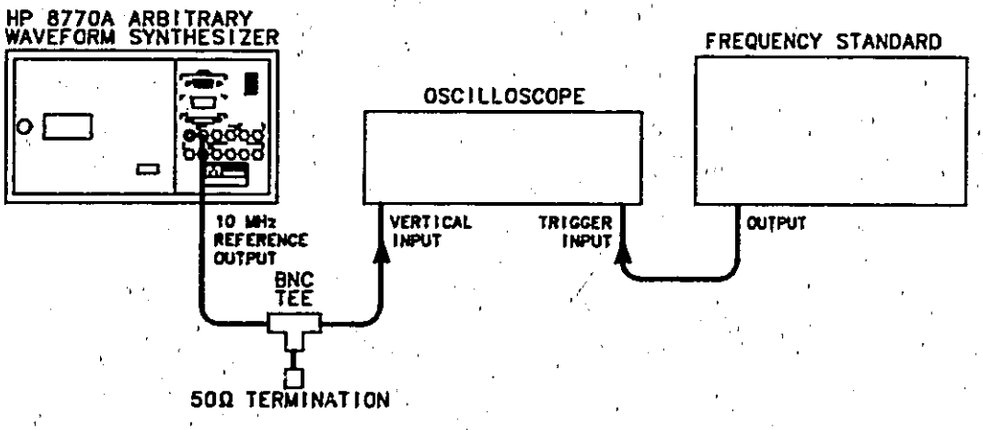


Figure 4-3. 10 MHz Reference Oscillator Aging Rate Test Setup

3. Set the oscilloscope to EXTERNAL TRIGGER and adjust it so that its sweep is synchronized with the reference frequency, (stable display).

PERFORMANCE TESTS

10 MHz REFERENCE OSCILLATOR AGING RATE (cont'd)

Procedure (cont'd)

4. Measur time required for a phase change of 360°. Record the time (T_1) in seconds.

T_1 = _____ s

5. Wait for a period of time (from 3 to 24 hours) and re-measure the phase change time. Record the period of time between measurements (T_2) in hours and the new phase change time (T_3) in seconds.

T_2 = _____ h

T_3 = _____ s

6. Calculate the aging rate from the following equation:

$$\text{Aging Rate} = \left| \left(\frac{1 \text{ cycle}}{f} \right) \left(\frac{1}{T_1} - \frac{1}{T_3} \right) \left(\frac{T}{T_2} \right) \right|$$

where: 1 cycle = the phase change reference for the time measurement (in this case, 360°)
f = Synthesizer's reference output frequency (10 MHz)
T = specified time for aging rate (24h)
T_1 = initial time measurement for a 360° (1 cycle) change
T_2 = time between measurements
T_3 = final time measurement for a 360° (1 cycle) change

for example:

if T_1 = 351s
T_2 = 3h
T_3 = 349s

then:

$$\text{Aging Rate} = \left| \left(\frac{1 \text{ cycle}}{10 \text{ MHz}} \right) \left(\frac{1}{351\text{s}} - \frac{1}{349\text{s}} \right) \left(\frac{24\text{h}}{3\text{h}} \right) \right|$$

$$= 1.306 \times 10^{-11}/\text{day}$$

7. Verify that the aging rate is less than 5×10^{-10}/day.

NOTE

If the absolute frequencies of the frequency standard and the Synthesizer's reference oscillator are extremely close, (within 1 Hz) the measurement time in steps 4 and 5 (T_1, T_2 and T_3) can be reduced by measuring the time required for a phase change of something less than 360°. Change 1 cycle in the formula to whichever of the following is appropriate: 180° = 1/2 cycle, 90° = 1/4 cycle. This can be accomplished by performing the 10 MHz Reference Oscillator Adjustment, in Section V, Adjustments, of the Operating and Service Manual.

Calculated Aging Rate _____ $< 5 \times 10^{-10}$/day

PERFORMANCE TESTS

4-12. REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS

Description

This test checks a portion of the Synthesizer's SUPPLEMENTAL CHARACTERISTICS shown in Table 1-2. The checks do not test SPECIFICATIONS of the Synthesizer. The signal parameters and waveforms shown are typical unless otherwise indicated. Judgement must be exercised in determining if the performance is far enough away from the typical value to justify a repair. The following inputs and outputs are checked for proper functioning:

10 MHz REFERENCE
- OUTPUT 1
- OUTPUT 2
- INPUT

SAMPLING CLOCKS
- INPUT
- CLK/8 OUTPUT

MARKER OUTPUTS
- SCAN START
- PACKET START
- SEQUENCE START
- EQUAL ADDRESS

PACKET ADVANCE
- READY OUTPUT
- TRIGGER INPUT

EXTERNAL DATA INPUT (Optional)

Equipment

50 MHz Oscilloscope	HP 1980B
Frequency Counter	HP 5343A
HP-IB Compatible Controller	HP 9000 Series 200 Model 236
Series 200 BASIC Language	HP 98613A
Optional for Checking the External Data Port	
HP-IB/GPIO Compatible Controller	HP 9000 Series 200 Model 236
GPIO Binary Extension File	P/O HP 98613A
GPIO Interface Bus and Cable	HP 98622A
External Data Port Cable	HP 11738A

Procedure

10 MHz Reference Input/Outputs

1. Connect the rear panel output, 10 MHz REFERENCE OUTPUT 1, to an oscilloscope. Set the oscilloscope's input to DC coupled. The level of the sine wave displayed should be >1.0 volt peak-to-peak.

2. Remove the short gray jumper cable, on the rear of the instrument, from 10 MHz REFERENCE OUTPUT 2. Disconnect the oscilloscope from 10 MHz REFERENCE OUTPUT 1 and connect to 10 MHz REFERENCE OUTPUT 2. The signal level should be >1.0 volt peak-to-peak. Leave the short jumper cable disconnected until instructed to reconnect it.

3. Connect an HP-IB compatible controller to the rear panel HP-IB connector on the Synthesizer. Ensure that the Synthesizer address is set to "719," (refer to Section II, Installation). Connect the frequency counter to the Synthesizer's RF OUTPUT connector. Execute the program command:

   OUTPUT 719; "GEN"

   The counter should read 12.500000 MHz ± 100 Hz. Record the RF OUTPUT signal frequency.

   RF OUTPUT Frequency _____ MHz

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

10 MHz Reference Input/Outputs (cont'd)

NOTE

If the Synthesizer does not read within the required tolerance, perform the Phase-Locked Loop Adjustment in Section V. The exact frequency read by the counter is dependent upon the Synthesizer's internal 125 MHz Clock plus any relative error introduced by the frequency counter's internal time base.

4. Connect 10 MHz REFERENCE OUTPUT 2 to the 10 MHz REFERENCE INPUT using the short jumper cable, while observing the frequency counter. The frequency counter reading of 12.5 MHz should change by several Hz up to several hundred Hz.

NOTE

This occurs as a result of the Synthesizer's internal freerunning 125 MHz Sampling Clock phase-locking to the Synthesizer's 10 MHz reference. If the frequency does not change, then a failure has occurred in the instrument or the free-running sampling clock frequency is adjusted within 1 Hz of the 10 MHz reference. An adjustment this close is highly improbable. The Synthesizer should be checked for a failure.

Sampling CLocks

5. Disconnect the frequency counter from the RF OUTPUT and connect the counter to the SAMPLING CLOCKS CLK/8 OUTPUT. The counter should read 15.625000 MHz ± 100 Hz.

6. Connect the 10 MHz REFERENCE OUTPUT 1 to the SAMPLING CLOCKS INPUT with a cable. With the computer, execute the command:

   OUTPUT 719; "CLKSEL EXT"

   The counter should read 1.250000 MHz ± 100 Hz. This demonstrates the use of an external sampling clock. The Synthesizer's 10 MHz reference is being used in place of the 125 MHz internal sampling clock.

7. Disconnect the counter and cables from the 10 MHz REFERENCE AND SAMPLING CLOCK connectors. Leave the short jumper cable connected between 10 MHz REFERENCE OUTPUT 2 and 10 MHz REFERENCE INPUT.

Marker Outputs

8. Clear the Synthesizer's memory by turning it off then on again. When the Synthesizer has completed its power-up diagnostics, execute the HP-IB commands:

   OUTPUT 719;"SINPQ SINA,1,512;SINPQ SINB,2,512"
   OUTPUT 719;"PACKET SINA,2,AUTO,SINB,1,AUTO;GO"
   OUTPUT 719;"MARKER SINA,256"

   These commands perform the following functions:
   a) Create SINA, 1 cycle long with 512 points.
   b) Create SINB, 2 cycles long with 512 points.
   c) Create a sequence of two scans through SINA followed by one scan of SINB.
   d) Output a marker at the 256th point of the SINA scan.

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

Marker Outputs (cont'd)

9. Connect Channel A of the two channel oscilloscope to the Synthesizer's RF OUTPUT. Connect Channel B to the MARKER OUTPUT SCAN START. Refer to Figure 4-4 for test setup.

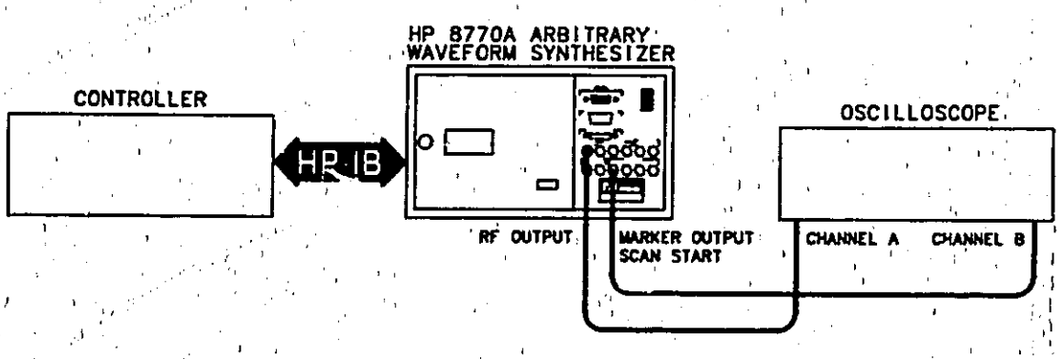


Figure 4-4. Marker Output Test Setup

10. Set the oscilloscope to the following settings:

Channel A	Channel B	Display
1 volt/div. 50Ω Input	2 volts/div. 50Ω Input	Alternate A & B Trigger — Internal; Channel B Time Base — 2 μs Trigger Level — for a stable display

The oscilloscope should display the waveform as shown in Figure 4-5. Channel B will show a SCAN START marker pulse at the start of each new scan of a wave segment. The marker will lead the start of the scan by 370 ns ± 25 ns.

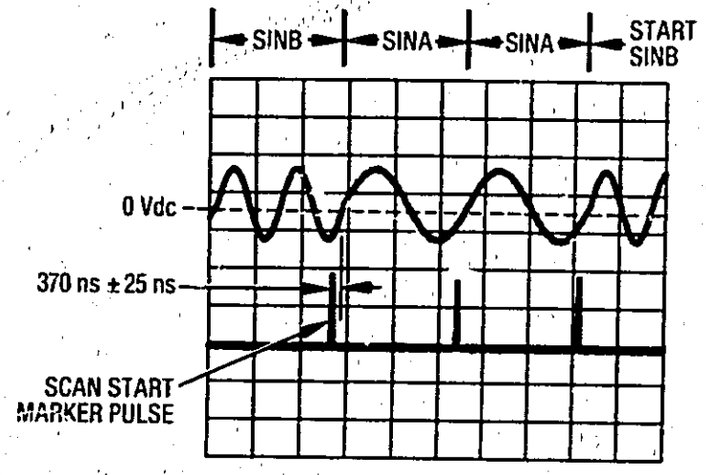


Figure 4-5. Marker Output Scan Start

11. Now check MARKER OUTPUT PACKET START, SEQUENCE START and EQUAL ADDRESS by disconnecting the oscilloscope from SCAN START and

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

Marker Outputs (cont'd)

connecting it to each of the marker outputs. The oscilloscope should display waveforms as shown in Figures 4-6 through 4-8 respectively. Disconnect the oscilloscope when the test is completed.

NOTE

When viewing the EQUAL ADDRESS marker output, adjust the trigger level starting at the most negative value. Increase the trigger value until a stable display as in Figure 4-8 is produced.

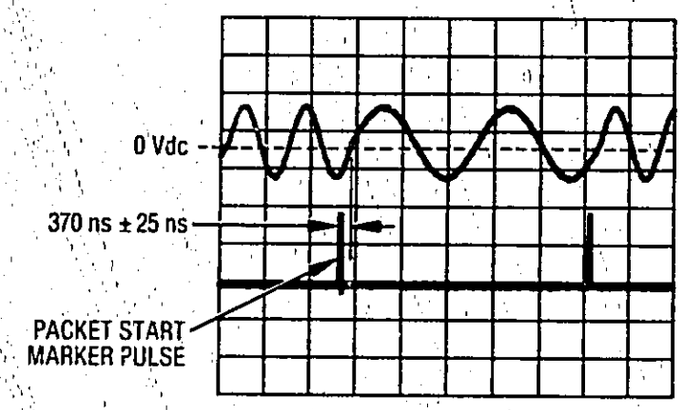


Figure 4-6. Market Output Packet Start

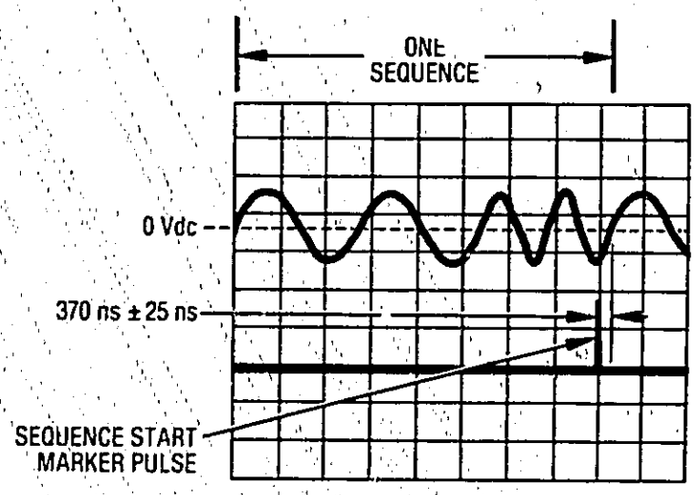


Figure 4-7. Marker Output Sequence Start

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

Marker Outputs (cont'd)

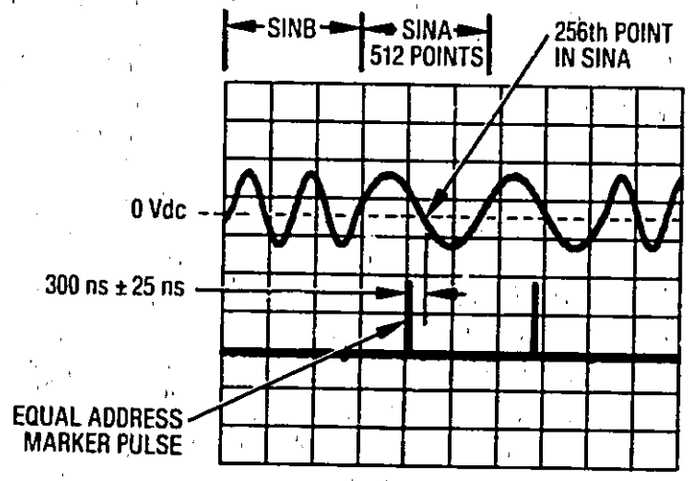


Figure 4-8. Marker Output Equal Address

Packet Advance

12. Turn off the Synthesizer, then turn it on again. Connect the PACKET ADVANCE READY OUTPUT to PACKET ADVANCE TRIGGER INPUT with a coaxial cable. Connect Channel A of the oscilloscope to the Synthesizer's RF OUTPUT. Connect Channel B to MARKER OUTPUT SEQUENCE START.

13. Execute the following HP-IB commands:

    OUTPUT 719;"SINPQ SINA,1,512;SINPQ SINB,2,512"
    OUTPUT 719;"PACKET SINA,1,EXT,SINB,1,EXT;GO"

    Readjust the oscilloscope trigger level for a stable display. The waveform displayed on the oscilloscope should be as shown in Figure 4-9.

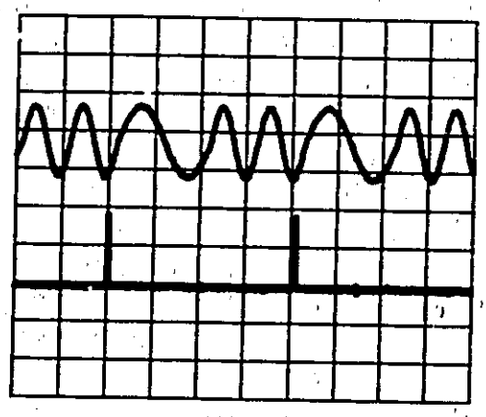


Figure 4-9. Waveform Using External Trigger Packet Advance

14. Disconnect the PACKET ADVANCE READY OUTPUT from the PACKET ADVANCE TRIGGER INPUT. Set the oscilloscope to trigger from Channel A.

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

Packet Advance (cont'd)

Readjust the trigger level for a stable display. The oscilloscope should now display a continuous sine wave, with a period of 2 µs, as shown in Figure 4-10.

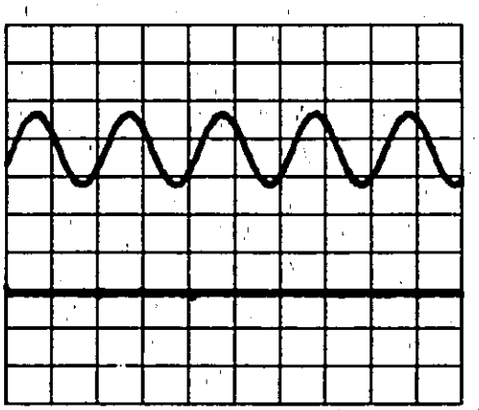


Figure 4-10. Waveform With External Trigger Removed

External Data Input

NOTE

The following test requires specialized equipment and software. It is optional and should not be performed as part of the standard Performance Tests. It should be performed whenever verification of the External Data Port is needed.

Equipment

HP-IB/GPIO Compatible Controller	HP 9000 Series 200 Model 236
Series 200 BASIC Language	HP 98613A
GPIO Extension File	P/O HP 98613A or HP 11775A
GPIO Interface	HP 98622A
Oscilloscope	HP 1980B
External Data Port Cable	HP 11738A

15. Connect the equipment as shown in Figure 4-11.

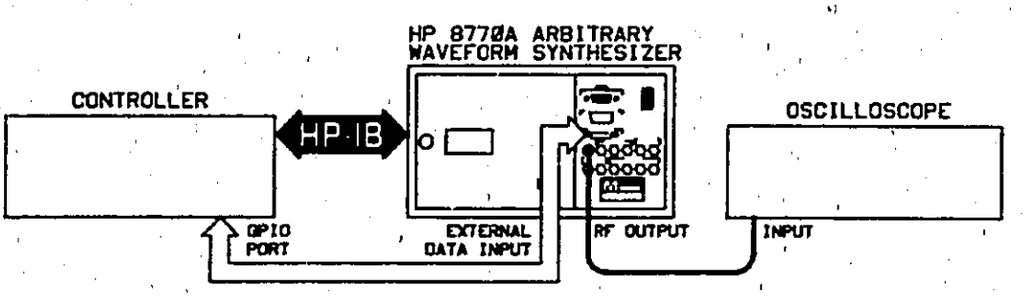


Figure 4-11. External Data Input Test Setup

PERFORMANCE TESTS

REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS (cont'd)

Procedure (cont'd)

External Data Input (cont'd)

16. Set the oscilloscope as follows: Channel A, internal trigger, time base of 2 μs/division and vertical gain of 0.2 volts/division.
17. Load the BASIC Language and GPIO Binary Extension file into the controller. Next load the following BASIC program into the controller and run the program:

```
10 INTEGER A(0:1023), I
20 ASSIGN @Gpio to 12;FORMAT OFF,WORD
30 OUTPUT 719;"EXDABT; PURGE BOTH;EXDWAV TEMP,1024"
40 FOR I = 0 TO 1023
50 A(I)=I*2
60 NEXT I
70 OUTPUT @Gpio;A(*)
80 OUTPUT 719;"PACKET TEMP,1,AUTO;GO"
90 END
```

NOTE

This program example is contained in commented form in Section III, Operation, of this manual.

18. The program will create a data array representing a linear ramp waveform. It will then download the data into the Synthesizer's waveform memory and play it back.
19. The oscilloscope should display a ramp as shown in Figure 4-12.

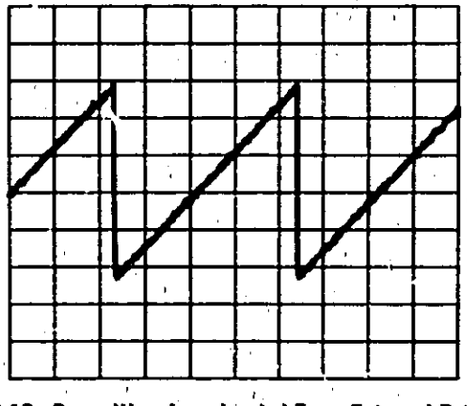


Figure 4-12. Ramp Waveform Loaded From External Data Input

Table 4-2. Operation Verification Test Record

Hewlett-Packard Company
Model HP 8770A
Arbitrary Waveform Synthesizer
Serial Number _____ Tested by _____
Date _____

Para. No.	Test	Results		
		Min.	Actual	Max.
4-6.	**Steps 1 through 3**			
	POWER-UP DIAGNOSTIC		_____ (√)	
	Steps 4 through 7			
	10 MHz REFERENCE			
	Output 1 Voltage	1 volt peak-to-peak	_____	
	Output 1 Power	0 dBm	_____	
	Output 2 Voltage	1 volt peak-to-peak	_____	
	Output 2 Power	0 dBm	_____	
	Steps 8 through 12			
	RF OUTPUT LEVEL			
	Output Voltage	1 volt peak-to-peak	_____	
	Output Power			
	Standard Instrument (50Ω Output)	9.75 dBm	_____	10.25 dBm
	Option 002 Instrument (75Ω Output)	7.99 dBm	_____ (corrected)	8.49 dBm
	Steps 13 and 14			
	HARMONICS			
	Second		_____	50 dBc
	Third		_____	50 dBc
	Fourth		_____	50 dBc
	SPURIOUS SIGNALS		_____	50 dBc
	Steps 15 through 17			
	ATTENUATOR ACCURACY			
	10 through 80 dB		_____ (√)	
	Step 18			
	REAR PANEL INPUT/ OUTPUT FUNCTIONAL CHECKS		_____ (√)	

Table 4-3. Performance Test Record (1 of 2)

Hewlett-Packard Company
Model HP 8770A
Arbitrary Waveform Synthesizer
Serial Number _____

Tested by _____

Date _____

Para. No.	Test	Results		
		Min.	Actual	Max.
4-8.	**POWER-UP DIAGNOSTIC**		_____ (√)	
4-9.	**SINE WAVE OUTPUT POWER AND ATTENUATOR ACCURACY**			
	Output Power			
	Standard Instrument (50Ω Output)	9.75 dBm	_____	10.25 dBm
	Option 002 Instrument (75Ω Output)	7.99 dBm	_____ (corrected)	8.49 dBm
	Attenuator Relative Accuracy			
	Attenuator Setting (dB)			
	10	−0.2 dB	_____	+0.2 dB
	20	−0.4 dB	_____	+0.4 dB
	30	−0.5 dB	_____	+0.5 dB
	40	−0.8 dB	_____	+0.8 dB
	50	−0.9 dB	_____	+0.9 dB
	60	−1.0 dB	_____	+1.0 dB
	70	−1.2 dB	_____	+1.2 dB
	80	−1.4 dB	_____	+1.4 dB
	90	−1.5 dB	_____	+1.5 dB
	100	−1.6 dB	_____	+1.6 dB
	110	−1.8 dB	_____	+1.8 dB
4-10.	**SINE WAVE HARMONICS, SPURIOUS AND NONHARMONIC DISTORTION**			
	Harmonics (Carrier 1 MHz)			
	Second		_____	−50 dBc
	Third		_____	−50 dBc
	Fourth		_____	−50 dBc
	Harmonics (Carrier 25 MHz)			
	Second		_____	−40 dBc
	Third		_____	−40 dBc
	Fourth		_____	−40 dBc
	Spurious and Non-Harmonic Distortion (Carrier 25 MHz)		_____	−50 dBc

Table 4-3. Performance Test Record (2 of 2)

Hewlett-Packard Company
Model HP 8770A
Arbitrary Waveform Synthesizer
Serial Number _____

Tested by _____
Date _____

Para. No.	Test	Results		
		Min.	Actual	Max.
4-10.	**SINE WAVE HARMONICS, SPURIOUS AND NONHARMONIC DISTORTION (cont'd)**			
	Harmonics (Carrier 30 MHz)			
	Second		_____	−40 dBc
	Third		_____	−40 dBc
	Fourth		_____	−40 dBc
	Spurious and Non-Harmonic Distortion (Carrier 30 MHz)		_____	−40 dBc
4-11.	**10 MHz REFERENCE OSCILLATOR AGING RATE**			
	Aging Rate		_____	$<5\times10^{-10}$/day
4-12.	**REAR PANEL INPUT/OUTPUT FUNCTIONAL CHECKS**			
	10 MHz Reference		_____ (√)	
	Sampling Clocks		_____ (√)	
	Marker Outputs		_____ (√)	
	Packet Advance		_____ (√)	

4-13. AUTOMATED PERFORMANCE TESTS

Automated Performance Tests can save test time and provide consistent test procedures from instrument to instrument. Automated performance test software and manuals for specific Hewlett-Packard computers are ordered separately from your nearest Hewlett-Packard Sales Office for a nominal fee. The software and manuals, however, are not available at the time this manual was printed.

For your convenience, a tab has been provided in this manual to add the Automated Performance Test procedures.

APPENDIX A
MANUAL CHANGES

An instrument manufactured after the printing of this manual may have a serial number prefix that is not listed on the manual title page. In this case, your manual is provided with updating information to make it as current as possible. This updating information may consist of replacement or additional pages that should be incorporated into the manual to bring it up to date.

If you operate or service instruments with several different serial prefixes, you may wish to keep pages that are replaced with the updating information. Appendix A provides a place to file these pages.