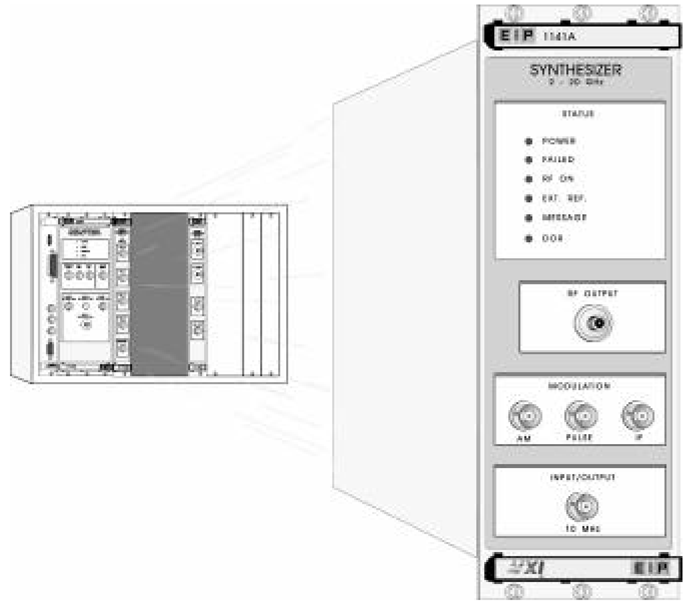


Models 1140A, 1141A, & 1142A VXIbus Synthesized Microwave Frequency Generators

Operation Manual

1140A CCN 8202-03
1141A CCN 9605-06
1142A CCN 8106-07

Manual Assy Part Number: 5585106-02
Manual Text Part Number: 5580106-02
Printed in USA

Warranty

Phase Matrix, Inc. warrants this product to be free from defects in material and workmanship for one year from the date of delivery. Damage due to accident, abuse, or improper signal level is not covered by the warranty. Removal, defacement, or alteration of any serial or inspection label, marking or seal may void the warranty. Phase Matrix, Inc. will repair or replace, at its option, any components of this product which prove to be defective during the warranty period, provided the entire unit is returned COLLECT to Phase Matrix, Inc. or an authorized repair facility. Please visit our web site at: www.phasematrix.com for up-to-date return information. In warranty units will be returned freight prepaid; out of warranty units will be returned freight COLLECT. No other warranty other than above is expressed or implied.

Certification

Phase Matrix, Inc. certifies this instrument to be in conformance with the specifications noted herein at time of shipment from the factory. Phase Matrix, Inc. further certifies that its calibration measurements are traceable to the National Institute of Standards and Technology (NIST).

Manual Change Information

As Phase Matrix, Inc. continually improves and updates its products, changes to the material covered by the manual will occur. When a part or assembly in a Phase Matrix, Inc. instrument is change to the extent that it is no longer interchangeable with the earlier part, the configuration control number (CCN) of the instrument, shown on the title page of the manual, will change, and a new edition of the manual will be published.

To maintain the technical accuracy of the manual, it may be necessary to provide new or additional information with the manual. In these cases, the manual is shipped with a Manual update. Please be sure to incorporate the information as instructed in the Manual update.

SAFETY

The Phase Matrix, Inc. Models 1140A / 1140A / 1142A are designed and tested according to international safety requirements, but as with all electronic equipment, certain precautions must be observed. This manual contains information, cautions, and warnings that must be followed to prevent the possibility of personal injury and / or damage to the instrument.

SAFETY AND HAZARD SYMBOLS

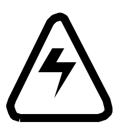
WARNING

A WARNING denotes a hazard to personnel. It calls attention to a procedure or practice, which, if not correctly performed or adhered to, could result in personal injury.

CAUTION

A CAUTION denotes a hazard to the equipment. It calls attention to an operating procedure or practice, which, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the product.

This is a general warning that appears whenever care is necessary to prevent damage to the equipment.



Dangerous Voltage

Toxic Substance

Static-Sensitive Component

Fire Hazard

OVERALL SAFETY CONSIDERATIONS

WARNING

Before this instrument is switched on, it's protective earth terminals *must* be connected to the AC power cord's protective conductor. The main plug *must* only be inserted in a socket/outlet that has a protective earth contact. The protective action must not be negated by using an extension cord (power cable) or adapter that does not have a protective earth (grounding) conductor.

WARNING

Use only fuses of the type specified with the required current and voltage ratings. Never use repaired fuses or short-circuited fuse holders, as doing so causes shock and/or fire hazard.

WARNING

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

WARNING

All protective earth terminals, extension cords, autotransformers, and other devices connected to this instrument *must* be connected to a socket/outlet that has a protective earth contact. Any interruption of the protection causes a potential shock hazard that can result in personal injury.

WARNING

The power supply is energized whenever AC power is connected to this instrument. Disconnect the AC power cord before removing the covers to prevent electrical shock. Internal adjustments or servicing that must be done with the AC power cord connected must be performed only by qualified personnel.

WARNING

Since the power supply filter capacitors may remain charged after the AC power cord is disconnected from the equipment, disconnecting the power cord does not ensure that there is no electrical shock hazard.

WARNING

Some of the components used in this instrument contain resins and other chemicals that give off toxic fumes if burned. Be sure to dispose of these items properly.

WARNING

Beryllia (beryllium oxide) is used in the construction of the YTF assembly. This material, if handled incorrectly, can pose a health hazard. *NEVER* disassemble the microwave counter assembly.

CAUTION

Static sensitive components are used in the YTF Assembly. These components can be damaged if handled incorrectly.

CAUTION

Before connecting power to the instrument, ensure that the correct fuse is installed and the voltage-selection switch on the instrument's rear panel is set properly. Refer to INSTALLATION Section 2, *Installation*.

CAUTION

Excessive signal levels can damage this instrument. To prevent damage, do not exceed the specified damage level. Refer to the instrument specifications in Section 1 of this manual.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

Section 4 - Programming (Continued)

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF ILLUSTRATIONS

This Page Intentionally Left Blank

DECLARATION OF CONFORMITY

Application Of Council Directive 89/336/EEC

Standards to which Conformity is Declared:

EMC: EN50011
EN50082-1

Standards to which Compliance is Declared:

Safety: IEC 1010-1 (1990)

Manufacturer's Name:	EIP/Phase Matrix, Inc.
Manufacturer's Address:	109 Bonaventura Dr. San Jose, CA 95134
Type of Equipment:	VXI Frequency Generator
Model Name(s):	1140A/1141A/1142A
Tested By:	Rockford Engineering Services, Inc. 9959 Calaveras Road Sunol, CA 94586 USA
Project Engineer:	Mr. Bruce Gordon and Leo Hernandez
Reviewer:	Mr. Michael Gbadebo, P.E.

I, the undersigned, hereby declare that the equipment specified above conforms to Directives and Standards listed.

For: **Phase Matrix, Inc.**

Name: Mark Espinosa
Title: QA Manager

Signature: Mark Espinosa

Date: 11/01/2004

SECTION 1
GENERAL INFORMATION

DESCRIPTION

The EIP Models 1140A, 1141A, and 1142A VXIbus Synthesized Microwave Frequency Generators are message-based VXI modules capable of generating microwave signals. The 1140A has a frequency range of 0.01 to 20 GHz with a dynamic range of +10 to -90 dBm. The 1141A has a frequency range of 2 GHz to 20 GHz with a dynamic range of +10 to -90 dBm. The 1142A has a frequency range of 4 GHz to 12.4 GHz with a dynamic range of +10 to -100 dBm. Other than interface address switches, the instruments have no manual controls. The instruments are normally controlled via a computer using SCPI type commands. The instruments also provide, as standard, a variety of external modulation inputs including: AM, FM, Pulse, and complex modulation. The 1140A, 1141A, and 1142A are VXIbus "C" size, 3-wide plug-in modules that require a VXIbus mainframe for operation.

OPERATING CONDITIONS

The EIP 1140A, 1141A, and 1142A synthesizers are designed to operate at temperatures from 0 to 50 °C at a relative humidity not to exceed 95% (75% over 25 °C; 45% over 40 °C). The synthesizers will perform to specifications at altitudes not exceeding 10,000 ft (3050 m). They are fungus resistant. The module housings are not designed to provide protection from severe mechanical shock or liquids and are intended for normal VXIbus use in an environmentally clean area.

The 1140A, 1141A, and 1142A synthesizers meet the requirements of MIL-T-28800D, Type III, Class 7, Style G, Color R with the following modifications and exceptions:

1. The non-operating temperature requirement is limited to the range of -40 to +70 °C.
2. The operating and non-operating altitude requirements are not invoked.
3. The EMI requirement is modified as follows:
   a. For frequencies ≥1 GHz, RE02 of MIL-STD-461C applies.
   b. For frequencies <1 GHz, VXIbus System Specifications Revision 1.3/1.4 applies.
4. The warm up time is 15 minutes at 25 °C ambient temperature.

STORAGE

To prevent possible damage to the synthesizers, they must be stored in an antistatic bag or enclosure and in an environment that is protected from moisture, dust, and other contaminants. Do not expose the instruments to temperatures below -40 °C or above 70 °C, altitudes above 40,000 ft (12,000 m), nor vibration exceeding 2 g.

1140A SPECIFICATIONS

GENERAL

Operating Temperature Range	0 to 50 °C
Non-operating Temperature Range	-40 to 70 °C
Relative Humidity	0 to 95%, non-condensing
EMI	
Below 1 GHz	Complies with VXIbus Revision 1.3/1.4 specifications
Above 1 GHz	Complies with RE02 of MIL-STD-461C
Warm-up Time	15 minutes at 25 °C ambient temperature
Weight	<18 lbs

VXIbus

Compatibility	Full compliance with VXIbus Specification for message-based instruments
Module Size	C-size, 3 slots wide
Device Type	Message-based instrument
Protocol	Word Serial
Address/Data Modes Supported	A16/D16
Local Bus	Not used
ECLTRG Utilization	Available for triggerable functions
TTLTRG Utilization	Available for triggerable functions
CLK10 Utilization	Not used
Cooling	0.1 mm H_2O @ 5 liters/sec for <25 °C internal temperature rise
Power Dissipation	<160 watts
Current Requirements	

Voltage (VDC)	+5	+12	+24	-2	-5.2	-12	-24
I_{PEAK} (Amperes)	2.0	2.8	2.9	0.2	0.2	0.8	2.0

FREQUENCY

Range	0.01 to 20 GHz
Resolution	1 Hz
Accuracy	Same as timebase
Internal Timebase	
Frequency	10 MHz
Aging Rate	<1 x 10^{-9}/day at 25 °C after 72 hours warm-up
Temperature Stability	<1 x 10^{-7} change over 0 to 50 °C
Switching Time	<50 ms to within 500 Hz (Triggered List Mode, 0.01 to 20 GHz step)

1140A SPECIFICATIONS (Continued)

PROGRAMMING

Compatibility	Conforms to SCPI Version 1993.0
Sweep Mode	Triggered List/Sweep

SPECTRAL PURITY
(at +10 dBm CW output 10 MHz to 18 GHz, +5 dBm 18 to 20 GHz, Complex Modulation OFF)

	10 MHz to 2 GHz	2 to 4.84 GHz	4.84 to 10 GHz	10 to 20 GHz
Subharmonic Spurious	None	None	None	<-60 dBc
Harmonic Spurious	<-20 dBc	<-25 dBc	<-25 dBc	<-25 dBc
Power Line Related Spurious	<-45 dBc	<-51 dBc	<-45 dBc	<-39 dBc

Non-Harmonically Related Spurious

	0.01 to 2 GHz		2 to 4.84 GHz		4.84 to 10 GHz		10 to 20 GHz	
Offset Frequency	Level	Typical	Level	Typical	Level	Typical	Level	Typical
<100 kHz	<-40 dBc	<-50 dBc	<-46 dBc	<-56 dBc	<-40 dBc	<-50 dBc	<-34 dBc	<-44 dBc
100 kHz to <1 MHz	<-50 dBc	<-60 dBc	<-56 dBc	<-66 dBc	<-50 dBc	<-60 dBc	<-44 dBc	<-54 dBc
1 to 100 MHz	<-60 dBc	<-70 dBc	<-60 dBc	<-70 dBc	<-60 dBc	<-70 dBc	<-60 dBc	<-63 dBc
>100 MHz	<-60 dBc①	<-70 dBc	<-70 dBc	<-75 dBc	<-70 dBc	<-75 dBc	<-60 dBc	<-75 dBc

Residual Modulation (50 Hz to 15 kHz bandwidth)

	10 MHz to 2 GHz	2 to 4.84 GHz	4.84 to 10 GHz	10 to 20 GHz
FM	<150 Hz rms	<75 Hz rms	<150 Hz rms	<300 Hz rms
AM	<0.1% peak	<0.1% peak	<0.1% peak	<0.1% peak

Single-sideband Phase Noise (dbc/Hz)②

Frequency	Offset from Carrier 30 Hz	100 Hz	1 kHz	10 kHz	100 kHz	Frequency	Offset from Carrier 30 Hz	100 Hz	1 kHz	10 kHz	100 kHz
10 MHz	-81	-84	-84	-85	-87	11 GHz	-76	-78	-80	-79	-81
1 GHz	-78	-81	-82	-83	-84	12 GHz	-75	-78	-79	-79	-81
2 GHz	-85	-86	-89	-92	-93	13 GHz	-74	-77	-78	-79	-81
3 GHz	-83	-85	-88	-91	-93	14 GHz	-72	-77	-77	-78	-80
4 GHz	-83	-85	-84	-86	-87	15 GHz	-72	-76	-77	-78	-79
5 GHz	-83	-84	-84	-85	-87	16 GHz	-72	-75	-76	-77	-78
6 GHz	-81	-84	-84	-85	-87	17 GHz	-72	-75	-76	-76	-78
7 GHz	-78	-82	-83	-84	-86	18 GHz	-71	-73	-76	-76	-78
8 GHz	-78	-81	-82	-83	-84	19 GHz	-71	-73	-76	-75	-76
9 GHz	-77	-79	-80	-82	-84	20 GHz	-71	-73	-76	-74	-74
10 GHz	-77	-79	-80	-80	-80						

① 1.8 to 2 GHz, -55 dBc

② Typical performance is >7 dB lower than specified.

1140A SPECIFICATIONS (Continued)

RF OUTPUT	
Range (Leveled)	
10 MHz to 2 GHz	+10 to -80 dBm
2 to 10 GHz	+13 to -90 dBm
10 to 18 GHz	+10 to -90 dBm
18 to 20 GHz	+5 to -90 dBm
Resolution	0.1 dB
Power Accuracy (In CW mode with attenuator and ALC coupled)	
10 MHz to 2 GHz	± 2.0 dB, >-50 dBm; ± 3.0 dB, ≤-50 dBm
2 to 20 GHz	± 1.0 dB, >-50 dBm; ± 2.0 dB, -50 to -80 dBm; ± 2.0 dB typical, <-80 dBm
Output Level Switching Time (Triggered List mode)	With attenuator change: <100 ms typical Without attenuator change: <5 ms/dB typical
Source Impedance	50 Ω nominal
VSWR	<2.0:1 (0 dB attenuation) typical
Connector	APC 3.5 female
Reverse Power Tolerance	1 watt continuous

10 MHz INPUT/OUTPUT	
Frequency	10 MHz
Level	0 dBm ± 3 dB
Impedance	50 Ω nominal
Connector	BNC female

PULSE MODULATION (external) (at +10 dBm output power)	
Pulse Repetition Frequency	DC to 10 MHz
Minimum Pulse Width	50 ns
On/Off Ratio	>80 dB
Rise/Fall Time	<15 ns, 10% to 90%
Pulse Overshoot, Ringing	<10% for PRF's <1 MHz
Pulse Width Compression	<10 ns at 50% points (<5 ns typical)
Video Feedthrough	<20 mV p-p (<10 mV p-p typical)
Delay Time	<55 ns, 50% TTL to 50% RF (<30 ns typical)
Peak-to-CW Level Accuracy	<0.5 dB change (>50 ns pulse widths excluding leading edge overshoot/ringing)
Input Level	TTL compatible
Input Level Tolerance	-0.5 ≤Vin ≤ +7.0 Vdc continuous
Polarity	RF output is ON with a TTL logic "1" input
Connector	BNC female

1140A SPECIFICATIONS (Continued)

AMPLITUDE MODULATION (external)	
Rate	DC to 100 kHz (3 dB bandwidth, typical)
Depth	0% to 90% minimum
Distortion	<5% (50% depth, 1 kHz rate)
Sensitivity	Programmable from 0% to 100%. 2.0 V p-p input gives full-scale modulation.
Modulation Index Accuracy	± 10% (50% depth, 1 kHz rate, 2.0 V p-p modulating input)
Modulation Overdrive Threshold	± 2 Vdc ± 10%
Average Power Output	$-20 \log\left(1 + \frac{\text{Modulation Index}}{100}\right)$ ± 2 dB relative to set CW level with AM OFF
Input Impedance	10 kΩ ± 10%
Input Level Tolerance	± 20 Vdc continuous
Connector	BNC female

IF INPUT (Complex Modulation)	
Input Frequency	300 MHz to 1 GHz, programmable
Input Level	-6 dBm nominal
Instantaneous 3 dB Bandwidth	>50 MHz typical (ALC OFF)

Spurious Output (+10 dBm output level, ALC ON, -6 dBm input level), typical

IF Input	Level
300 to <700 MHz	<-30 dBc
700 MHz to 1 GHz	<-60 dBc

Input Impedance	50 Ω nominal
Input VSWR	<2.0:1 typical
Connector	BNC female

Note: Specifications subject to change without notice.

1141A SPECIFICATIONS

GENERAL

Operating Temperature Range	0 to 50 °C
Non-operating Temperature Range	-40 to 70 °C
Relative Humidity	0 to 95%, non-condensing
EMI	
Below 1 GHz	Complies with VXIbus Revision 1.3/1.4 specifications
Above 1 GHz	Complies with RE02 of MIL-STD-461C
Warm-up Time	15 minutes at 25 °C ambient temperature
Weight	<18 lbs

VXIbus

Compatibility	Full compliance with VXIbus Specification for message-based instruments
Module Size	C-size, 3 slots wide
Device Type	Message-based instrument
Protocol	Word Serial
Address/Data Modes Supported	A16/D16
Local Bus	Not used
ECLTRG Utilization	Available for triggerable functions
TTLTRG Utilization	Available for triggerable functions
CLK10 Utilization	Not used
Cooling	0.1 mm H_2O @ 5 liters/sec for <25 °C internal temperature rise
Power Dissipation	<155 watts

Current Requirements

Voltage (VDC)	+5	+12	+24	-2	-5.2	-12	-24
I_{PEAK} (Amperes)	2.0	2.8	2.5	0.2	0.2	0.8	1.7

FREQUENCY

Range	2 to 20 GHz
Resolution	1 Hz
Accuracy	Same as timebase
Internal Timebase	
Frequency	10 MHz
Aging Rate	$<1 \times 10^{-9}$/day at 25 °C after 72 hours warm-up
Temperature Stability	$<1 \times 10^{-7}$ change over 0 to 50 °C
Switching Time	<50 ms to within 500 Hz (Triggered List Mode, 2 to 20 GHz step)

1141A SPECIFICATIONS (Continued)

PROGRAMMING

Compatibility: Conforms to SCPI Version 1993.0

Sweep Mode: Triggered List/Sweep

SPECTRAL PURITY (at +10 dBm CW output level, Complex Modulation OFF)

	2 to 4.84 GHz	4.84 to 10 GHz	10 to 20 GHz
Subharmonic Spurious	None	None	<-60 dBc
Harmonic Spurious	<-25 dBc	<-25 dBc	<-25 dBc
Power Line Related Spurious	<-51 dBc	<-45 dBc	<-39 dBc

Non-Harmonically Related Spurious

	2 to 4.84 GHz		4.84 to 10 GHz		10 to 20 GHz	
Offset Frequency	Level	Typical	Level	Typical	Level	Typical
<100 kHz	<-46 dBc	<-56 dBc	<-40 dBc	<-50 dBc	<-34 dBc	<-44 dBc
100 kHz to <1 MHz	<-56 dBc	<-66 dBc	<-50 dBc	<-60 dBc	<-44 dBc	<-54 dBc
1 to 100 MHz	<-60 dBc	<-70 dBc	<-60 dBc	<-70 dBc	<-60 dBc	<-63 dBc
>100 MHz	<-70 dBc	<-75 dBc	<-70 dBc	<-75 dBc	<-60 dBc	<-75 dBc

Residual Modulation (50 Hz to 15 kHz bandwidth)

	2 to 4.84 GHz	4.84 to 10 GHz	10 to 20 GHz
FM	<75 Hz rms	<150 Hz rms	<300 Hz rms
AM	<0.1% peak	<0.1% peak	<0.1% peak

Single-sideband Phase Noise (dbc/Hz)*

Frequency	Offset from Carrier					Frequency	Offset from Carrier				
GHz	30 Hz	100 Hz	1 kHz	10 kHz	100 kHz	GHz	30 Hz	100 Hz	1 kHz	10 kHz	100 kHz
2.0	-85	-86	-89	-92	-93	12.0	-75	-78	-79	-79	-81
3.0	-83	-85	-88	-91	-93	13.0	-74	-77	-78	-79	-81
4.0	-83	-85	-84	-86	-87	14.0	-72	-77	-77	-78	-80
5.0	-83	-84	-84	-85	-87	15.0	-72	-76	-77	-78	-79
6.0	-81	-84	-84	-85	-87	16.0	-72	-75	-76	-77	-78
7.0	-78	-82	-83	-84	-86	17.0	-72	-75	-76	-76	-78
8.0	-78	-81	-82	-83	-84	18.0	-71	-73	-76	-76	-78
9.0	-77	-79	-80	-82	-84	19.0	-71	-73	-76	-75	-76
10.0	-77	-79	-80	-80	-80	20.0	-71	-73	-76	-74	-74
11.0	-76	-78	-80	-79	-81						

* Typical performance is >7 dB lower than specified.

1141A SPECIFICATIONS (Continued)

RF OUTPUT	
Range (Leveled) 2 to 10 GHz 10 to 20 GHz	 +13 to -90 dBm +10 to -90 dBm
Resolution	0.1 dB
Power Accuracy (In CW mode with attenuator and ALC coupled)	± 1.0 dB, >-50 dBm ± 2.0 dB, -50 to -80 dBm ± 2.0 dB typical, <-80 dBm
Output Level Switching Time (Triggered List mode)	With attenuator change: <100 ms typical Without attenuator change: <5 ms/dB typical
Source Impedance	50 Ω nominal
VSWR	<2.0:1 (0 dB attenuation) typical
Connector	APC 3.5 female
Reverse Power Tolerance	1 watt continuous

10 MHz INPUT/OUTPUT	
Frequency	10 MHz
Level	0 dBm ± 3 dB
Impedance	50 Ω nominal
Connector	BNC female

PULSE MODULATION (external) (at +10 dBm output power)	
Pulse Repetition Frequency	DC to 10 MHz
Minimum Pulse Width	50 ns
On/Off Ratio	>80 dB
Rise/Fall Time	<15 ns, 10% to 90%
Pulse Overshoot, Ringing	<10% for PRF's <1 MHz
Pulse Width Compression	<10 ns at 50% points (<5 ns typical)
Video Feedthrough	<20 mV p-p (<10 mV p-p typical)
Delay Time	<55 ns, 50% TTL to 50% RF (<30 ns typical)
Peak-to-CW Level Accuracy	<0.5 dB change (>50 ns pulse widths excluding leading edge overshoot/ringing)
Input Level	TTL compatible
Input Level Tolerance	-0.5 ≤Vin ≤ +7.0 Vdc continuous
Polarity	RF output is ON with a TTL logic "1" input
Connector	BNC female

1141A SPECIFICATIONS (Continued)

AMPLITUDE MODULATION (external)	
Rate	DC to 100 kHz (3 dB bandwidth, typical)
Depth	0% to 90% minimum
Distortion	<5% (50% depth, 1 kHz rate)
Sensitivity	Programmable from 0% to 100%. 2.0 V p-p input gives full-scale modulation.
Modulation Index Accuracy	± 10% (50% depth, 1 kHz rate, 2.0 V p-p modulating input)
Modulation Overdrive Threshold	± 2 Vdc ± 10%
Average Power Output	$-20 \log\left(1 + \frac{\text{Modulation Index}}{100}\right)$ ± 2 dB relative to set CW level with AM OFF
Input Impedance	10 kΩ ± 10%
Input Level Tolerance	± 20 Vdc continuous
Connector	BNC female

IF INPUT (Complex Modulation)	
Input Frequency	300 MHz to 1 GHz, programmable
Input Level	-6 dBm nominal
Instantaneous 3 dB Bandwidth	>50 MHz typical (ALC OFF)

Spurious Output (+10 dBm output level, ALC ON, -6 dBm input level), typical

IF Input	Level
300 to <700 MHz	<-30 dBc
700 MHz to 1 GHz	<-60 dBc

Input Impedance	50 Ω nominal
Input VSWR	<2.0:1 typical
Connector	BNC female

Note: Specifications subject to change without notice.

1142A SPECIFICATIONS

GENERAL

Operating Temperature Range	0 to 50 °C
Non-operating Temperature Range	-40 to 70 °C
Relative Humidity	0 to 95%, non-condensing
EMI	
Below 1 GHz	Complies with VXIbus Revision 1.3/1.4 specifications
Above 1 GHz	Complies with RE02 of MIL-STD-461C
Warm-up Time	15 minutes at 25 °C ambient temperature
Weight	<18 lbs

VXIbus

Compatibility	Full compliance with VXIbus Revision 1.3/1.4 specifications for message-based instruments
Module Size	C-size, 3 slots wide
Device Type	Message-based instrument
Protocol	Word Serial
Address/Data Modes Supported	A16/D16
Local Bus	Not used
ECLTRG Utilization	Available for triggerable functions
TTLTRG Utilization	Available for triggerable functions
CLK10 Utilization	Not used
Cooling	0.1 mm H_2O @ 5 liters/sec for <15 °C internal temperature rise
Power Dissipation	<126 watts
Current Requirements	

Voltage (VDC)	+5	+12	+24	-2	-5.2	-12	-24
I_{PEAK} (Amperes)	2.0	2.2	1.7	0.2	0.2	0.8	1.6

FREQUENCY

Range	4 to 12.4 GHz
Resolution	1 Hz
Accuracy	Same as timebase
Internal Timebase	
Frequency	10 MHz
Aging Rate	<1 x 10^{-9}/day at 25 °C after 72 hours warm-up
Temperature Stability	<1 x 10^{-7} change over 0 to 50 °C
Switching Time	<40 ms to within 500 Hz (Triggered List Mode, 4 to 12.4 GHz step)

1142A SPECIFICATIONS (Continued)

PROGRAMMING

Compatibility — Conforms to SCPI Version 1993.0

Sweep Mode — Triggered List/Sweep

SPECTRAL PURITY (at +10 dBm CW output level, Complex Modulation OFF)

Subharmonic Spurious — None

Harmonic Spurious — <-40 dBc

Power Line Related Spurious — <-45 dBc

Non-Harmonically Related Spurious

Offset Frequency	Level	Typical
<100 kHz	<-40 dBc	<-50 dBc
100 kHz to <1 MHz	<-50 dBc	<-60 dBc
1 to 100 MHz	<-60 dBc	<-70 dBc
>100 MHz	<-70 dBc	<-75 dBc

Residual Modulation (50 Hz to 15 kHz bandwidth) — FM: <150 Hz rms, AM: <0.1% peak

Single-sideband Phase Noise (dbc/Hz)*

Frequency	Offset from Carrier				
GHz	30 Hz	100 Hz	1 kHz	10 kHz	100 kHz
4.0	-83	-87	-89	-86	-87
5.0	-83	-86	-89	-85	-87
6.0	-81	-85	-88	-85	-87
7.0	-78	-83	-87	-84	-86
8.0	-78	-82	-86	-83	-84
9.0	-77	-79	-84	-82	-84
10.0	-77	-79	-84	-80	-80
11.0	-76	-79	-81	-78	-79
12.4	-75	-78	-81	-78	-79

* Typical performance is >7 dB lower than specified.

RF OUTPUT

Range (Leveled) — +10 to -100 dBm

Resolution — 0.1 dB

Power Accuracy (In CW mode with attenuator and ALC coupled) — ± 1.0 dB, >-50 dBm; ± 2.0 dB, -50 to -80 dBm; ± 2.0 dB typical, <-80 dBm

Output Level Switching Time (Triggered List mode) — With attenuator change: <100 ms, typical; Without attenuator change: <5 ms/dB, typical

Source Impedance — 50 Ω nominal

VSWR — <2.0:1 (0 dB attenuation) typical

Connector — Type N female

Reverse Power Tolerance — 1 watt continuous

1142A SPECIFICATIONS (Continued)

10 MHz INPUT/OUTPUT	
Frequency	10 MHz
Level	0 dBm ± 3 dB (0.7 V p-p ± 0.2 V p-p)
Impedance	50 Ω nominal
Connector	BNC female

PULSE MODULATION (external) (at +10 dBm output power)	
Pulse Repetition Frequency	DC to 10 MHz
Minimum Pulse Width	<50 ns
On/Off Ratio	>80 dB
Rise/Fall Time	<15 ns, 10% to 90%
Pulse Overshoot, Ringing	<10% for PRF's <1 MHz
Pulse Width Compression	<10 ns at 50% points (<5 ns typical)
Video Feedthrough	<20 mV p-p (<10 mV p-p typical)
Delay Time	<55 ns, 50% TTL to 50% RF (<30 ns typical)
Peak-to-CW Level Accuracy	<0.5 dB change (>50 ns pulse widths excluding leading edge overshoot/ringing)
Input Level	TTL compatible
Input Level Tolerance	-0.5 ≤Vin ≤ +7.0 Vdc continuous
Polarity	RF output is ON with a TTL logic "1" input
Connector	BNC female

AMPLITUDE MODULATION (external)	
Rate	DC to 100 kHz (3 dB bandwidth, typical)
Depth	0% to 90% minimum
Distortion	<5% (50% depth, 1 kHz rate)
Sensitivity	Programmable from 0% to 100%. 2.0 V p-p input gives full-scale modulation.
Modulation Index Accuracy	± 10% (50% depth, 1 kHz rate, 2.0 V p-p modulating input)
Modulation Overdrive Threshold	± 2 Vdc ± 10%
Average Power Output	$-20 \log\left(1 + \frac{\text{Modulation Index}}{100}\right) \pm 2$ dB relative to set CW level with AM OFF
Input Impedance	10 kΩ ± 10%
Input Level Tolerance	± 20 Vdc continuous
Connector	BNC female

1142A SPECIFICATIONS (Continued)

IF INPUT (Complex Modulation)

Input Frequency	300 MHz to 1 GHz, programmable, in the 4 to 12 GHz range
Input Level	-6 dBm nominal
Instantaneous 3 dB Bandwidth	>50 MHz typical (ALC OFF)

Spurious Output (+10 dBm output level, ALC ON, -6 dBm input level), typical

IF Input	Level
300 to <700 MHz	<-30 dBc
700 MHz to 1 GHz	<-60 dBc

Input Impedance	50 Ω nominal
Input VSWR	<2.0:1 typical
Connector	BNC female

Note: Specifications subject to change without notice.

This Page Intentionally Left Blank

SECTION 2
INSTALLATION

UNPACKING

The EIP 1140A, 1141A, and 1142A VXIbus Synthesized Microwave Frequency Generators arrive ready for operation. Carefully inspect the shipping carton for any sign of visible or concealed damage. If the carton or instrument is damaged, immediately notify shipper's agent.

Remove the packing carton and supports, being careful not to scar or damage the instrument. Make a complete visual inspection of the synthesizer, checking for any damage or missing components. Report any problems to EIP immediately.

SETTING THE LOGICAL ADDRESS

Before installing the synthesizer in the VXIbus mainframe, verify that the logical address is between 1 and 254 (decimal). The factory default setting for the logical address of the synthesizer is 14 hexadecimal (20 decimal). The logical address of the synthesizer is set using the two rotary-type hexadecimal switches located on the bottom of the module (see Figure 2-1). To set the logical address, dial each switch to the hexadecimal value desired. For example, to set a logical address of 17 hexadecimal (23 decimal), use a small flathead screwdriver (or similar tool), and rotate the MSB switch to 1 and the LSB switch to 7. The logical address desired must be a value between decimal 1 and 254. Logical address 0 is reserved for Slot 0 devices. Logical address 255 is reserved for dynamically configured devices. The EIP 1140A, 1141A, and 1142A do not support dynamic configuration.

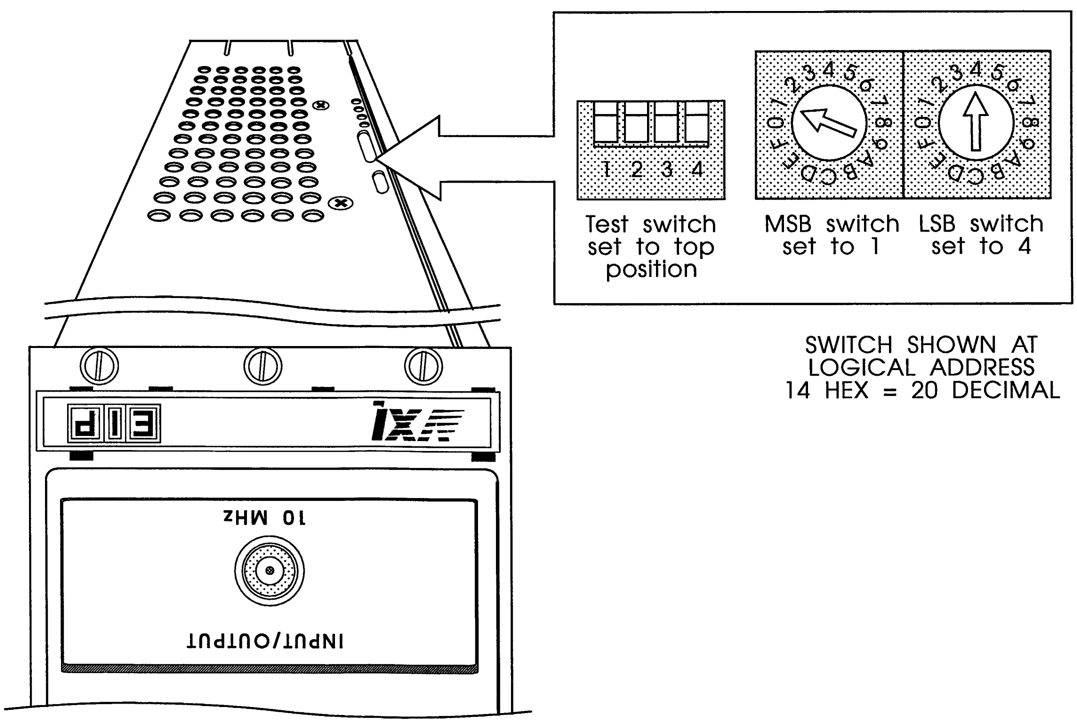


Figure 2-1. Logical Address Switch Locations.

POWER AND COOLING

The EIP 1140A, 1141A, and 1142A synthesizers operate over an ambient temperature range from 0 to 50 °C and consume up to 126 watts (see Specifications in Section 1). When configuring your VXIbus system, make sure that the chassis has sufficient power and cooling capacity for the synthesizer. Refer to chassis specifications and cooling capacity curves.

INSTALLATION

The 1140A, 1141A, and 1142A are VXIbus modules designed to be installed in a VXIbus mainframe. Prior to installing the synthesizer in a VXIbus mainframe, verify that all VXI defined voltages are present and within limits, and make sure the mainframe is capable of supplying the required current (see Specifications in Section 1).

CAUTION

Prior to installing the 1140A, 1141A, or 1142A in a VXIbus mainframe, verify that all the VXI defined voltages are present and that the mainframe is capable of supplying the required current.

CAUTION

Do not plug the synthesizer into VXIbus mainframe with power applied.

The synthesizer is a 3-slot, C-size module that can be installed into any slot of a VXIbus mainframe except slot 0. Slot 0 is reserved to the resource manager. To install the synthesizer into the VXIbus mainframe, first turn mainframe power off. Next, place the synthesizer card edges into the front mainframe guides (top and bottom). Gently slide the synthesizer towards the rear of the mainframe until the connectors just mate with the backplane. Firmly seat module with backplane connectors making sure the front panel is flush with the front of the card cage. Tighten down the retaining screws to ensure the module remains fully seated.

CAUTION

Do not use retaining screws to seat module.

INCOMING OPERATIONAL CHECK

A VXIbus mainframe along with a slot 0 resource manager and an instrument controller are required to verify that the synthesizer is operational. With the synthesizer installed in a VXIbus mainframe, two types of tests can be performed to verify proper operation.

Whenever a VXIbus mainframe is energized, the resource manager queries each device checking for proper operation. The synthesizer also performs a power-on self-test and lights the failed LED if any problems are detected.

The instrument can also be queried as to its operational status using a controller by issuing the self-test query ***TST?** to the instrument. The instrument will return a 0 if all tests pass.

IN CASE OF PROBLEMS

In the event that a problem does occur, there are a few things to check prior to returning the instrument for repair.

1. Verify logical address setting on the instrument.
2. Verify that all the VXI specified voltages are present.
3. The synthesizer monitors the VXIbus AC fail line. If this line is asserted, the synthesizer will not function.
4. If the unit has never worked in the particular system, the problem may be due to the system and not an instrument fault. In this case, call EIP at the phone number listed on the cover page of the manual and ask for Customer Support.

SERVICE INFORMATION

PERIODIC MAINTENANCE

No periodic preventive maintenance is required. However, to maintain accuracy, it is recommended that the synthesizer be recalibrated every 12 months or whenever a problem is suspected. The specific calibration interval depends upon the accuracy required.

SYNTHESIZER IDENTIFICATION

This synthesizer is identified by three sets of numbers: the model number (EIP 1140A, 1141A, or 1142A), serial number, and a configuration control number (CCN). These numbers are located on a label affixed to the top of the module and must be included in any correspondence regarding your synthesizer.

FACTORY SERVICE

If the synthesizer is being returned to EIP for service or repair, be sure to include the following information with the shipment.

- Name and address of owner.
- Model, complete serial number, and CCN of the synthesizer.
- A complete description of the problem. The main thing is to provide enough information so that the problem can be verified, i.e., Under what conditions did the problem occur? Did the unit work and then fail? What other equipment was connected to the synthesizer?
- Name and telephone number of someone familiar with the problem who may be contacted by EIP if any further information is required.
- Shipping address to which the synthesizer is to be returned. Include any special shipping instructions. Pack the synthesizer for shipping as detailed in Shipping Instructions.

SHIPPING INSTRUCTIONS

Place the synthesizer in an antistatic bag or enclosure, wrap in heavy plastic or kraft paper, and repack in the original container, if available. If the original container cannot be used, pack in a heavy (275 pound test) double-walled carton with approximately four inches of packing material between the synthesizer and the inner carton. Seal carton with strong filament tape or strapping. Mark the carton to indicate that it contains a fragile electronic instrument. Ship to the EIP address on the title page of the manual.

This Page Intentionally Left Blank

SECTION 3
FRONT PANEL CONNECTORS AND INDICATORS

INTRODUCTION

This section describes the functions of the front panel LED's and connectors.

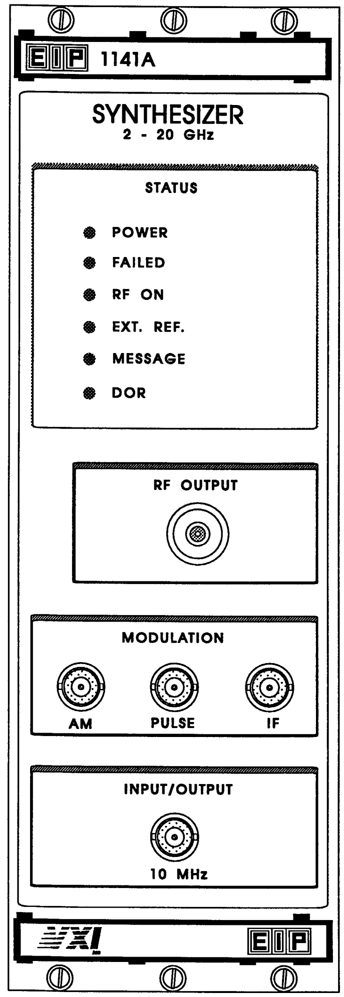


Figure 3-1. Front Panel (1141A Shown).

FRONT PANEL STATUS LEDs

- POWER - This LED lights to indicate that power is being supplied to the synthesizer. The synthesizer senses the voltage on all lines that are used and checks for sufficient voltage. If the required voltages are not present, the LED will not light.

- FAILED - This LED lights to indicate that the synthesizer has failed, or is in the process of executing, its self-test. Failures are typically the result of internal component failure or inadequate power supply current. The FAILED LED follows the condition of the VXIbus SYSFAIL line. If the unit has failed, the LED will remain lit even if SYSFAIL is inhibited by the commander.

- RF ON - This LED lights to indicate that microwave power is applied to the front panel RF output connector.

- EXT. REF. - This LED lights to indicate that the synthesizer is set up to input an external 10 MHz frequency reference from the front panel connector.

- MESSAGE - This LED lights to indicate that the synthesizer is either sending or receiving messages or data over the VXIbus. The MESSAGE LED is also illuminated when a commander accesses the synthesizer's VXIbus registers.

- DOR - This LED lights to indicate that the synthesizer is ready to output data.

FRONT PANEL CONNECTORS

- RF OUTPUT - Main output for the synthesizer. The 1140A and 1141A have an APC 3.5 female connector with a nominal 50 Ω impedance. The 1142A has a type N connector with a nominal 50 Ω impedance.

- MODULATION

  AM - Amplitude modulation input. BNC female connector with 10 kΩ input impedance with a frequency range of dc to 100 kHz. Optimum input level is 2 V p-p. Maximum input tolerance is ±20 Vdc continuous.

  PULSE - Pulse modulation input. BNC female connector. TTL compatible input with an input impedance of less than 1 standard TTL load. RF output is ON with TTL logical "1" input. Frequency range is dc to 10 MHz. Maximum input tolerance is -0.5 to +7 Vdc.

  IF - Complex modulation input. BNC female connector with a nominal input impedance of 50 Ω. Optimum input level is -6 dBm. Frequency range is 300 MHz to 1 GHz, programmable.

- INPUT/OUTPUT 10 MHz - Programmable as reference input or output. BNC female connector with a nominal input impedance of 50 Ω. Optimum input level is 0 dBm ±3 dB. Output level is 0 dBm ±3 dB. Maximum input tolerance is +10 dBm continuous.

SECTION 4
PROGRAMMING

INTRODUCTION

The programming interface for the EIP 1140A, 1141A, and 1142A was designed to conform to the requirements SCPI 1993.0. The SCPI interface was selected because it standardizes command syntax and style which simplifies the task of programming across a wide range of instrumentation. As with any programming language, exact command syntax must be used. Unrecognized commands will be ignored and an error will be recorded in memory. It is good programming technique to routinely check the error queue in the synthesizer while developing software.

The actual SCPI commands are made from individual keywords. The keywords can be a mixture of upper and lower case characters. Each keyword has both a long and a short form. The 1140A, 1141A, and 1142A will accept only the exact long or the exact short form of the keyword. Sending commands which include keywords which are not the exact long or short version will generate an error. The short form keyword is an abbreviation of the long form and is usually the first four characters of the long form. An exception to this is when the long form consists of more than four characters and the fourth character is a vowel. In such cases, the vowel is dropped and the short form becomes the first three characters of the long form. Throughout the programming section, the short form is capitalized and printed in **boldface** type to promote familiarity with the shortened form.

Using the long form is optional for user program readability. For example, FREQuency and FREQ are both recognized equivalently.

A <number> can be sent in any of the defined IEEE formats (NR1, NR2, NR3). For example, 12000, 12000.00, 001.2e4, and .12000E+5 are all recognized equivalently.

When a command has a query form with character response data, the short form value in upper case is always returned. For example, INT is returned for INTernal and DEF is returned for DEFault.

CONVENTIONS

GENERAL NOTATIONS

< > Angle brackets enclose parameter descriptions.

[ ] Square brackets enclose one or more optional parameters.

{ } Braces enclose one or more parameters that may be included one or more times.

| A vertical bar stands for "or" and is used to separate alternative parameter options. For example, ON|OFF is the same as ON or OFF.

::= Stands for "Is defined as". For example, <a>::=<b><c> means that <b><c> can replace <a> wherever it occurs.

→ Arrows show the path to follow when constructing a synthesizer command.

☐ Boxes enclose required elements of synthesizer commands, such as punctuation.

sp Boxes with sp inside implies whitespace.

Encloses keywords with their acceptable short form in capital letters.

Encloses the command arguments.

Encloses the default command arguments.

DATA FORMAT NOTATIONS

<arg>::=a generic command argument consisting of one or more of the other data formats.

<bNR1>::=boolean values in <NR1> format; numeric 1 or 0.

<boolean>::=ON|OFF. Can also be represented by “1” or “0”, where 1 means ON and 0 means OFF, on the command line. Boolean parameters are always returned as 1 or 0 in <NR1> format.

<char>::=<CHARACTER PROGRAM DATA>. <CHARACTER PROGRAM DATA> are: CW, FIXed, UP, Down, INTernal, and EXTernal.

<integer>::=unsigned integer without a decimal point, i.e., implied radix point.

<NR1>::=signed integer without a decimal point, i.e., implied radix point.

<NR2>::=signed integer with an explicit decimal point.

<NR3>:: =scaled explicit decimal point numeric value together with an exponent notation.

<NRf>::=NR1>|<NR2>|<NR3>

<nv>::=numeric value, which includes <NRf> and several forms of <CHARACTER PROGRAM DATA> such as MINimum, MAXimum, DEFault, UP, and DOWN. Not all of these are available for every command.

<setting>::=the current setting of the input parameter.

Suffixes can be used after numeric values. Examples are HZ (default), KHZ, MHZ, and GHZ for frequency; DBM for power; and % or PCT for AM:depth.

COMMON COMMANDS, IEEE 488.2

***CLS** (Clear Status Command)

→ *CLS →

This command clears all event registers in the status structure. Empties Output and Error queues. Refer to the Status subsystem section of the manual for more information.

***ESE** (Standard Event Status Enable Command)

Default: 0
Range: 0 through 255

→ *ESE — sp — <NRf> →

This command sets the bits in the Standard Event Status Enable register. Refer to the Status subsystem section of the manual for more information.

<NRf> must evaluate to an integer between 0 and 255.

Bit settings are as follows:

Bit 0 - Operation complete
Bit 1 - Not used (always 0)
Bit 2 - Query error
Bit 3 - Not used (always 0)
Bit 4 - Command execution error
Bit 5 - Command parsing error
Bit 6 - Not used (always 0)
Bit 7 - Unit has been reset

***ESE?** (Standard Event Status Enable Query)

Range: 0 through 255

→ *ESE — ? →

This query returns the value of the Standard Event Status Enable register. Refer to the Status subsystem section of the manual for more information.

***ESR?** (Standard Event Status Register Query)

Range: 0 through 255

→ *ESR — ? →

This query returns the value of the Standard Event Status register. Clears the value after being read. Refer to the Status subsystem section of the manual for more information.

***IDN?** (Identification Query)

→ *IDN → ? →

This query returns the unique identification of the device in a specified format:

<Manufacturer>,<Serial #>,<Firmware revision>.

Example:

EIP MICROWAVE,1142A,00122,V1.1

***OPC** (Operation Complete Command)

→ *OPC →

This command causes the Operation Complete bit (bit 0) in the Standard Event Status register to be set when all pending operations have been finished.

***OPC?** (Operation Complete Query)

→ *OPC → ? →

This query returns an ASCII character 1 when all pending operations have been finished.

***RCL** (Recall Command)

Range: 0

→ *RCL → sp → <NR1> →

This command reinstates a previously saved set of synthesizers settings. This command is used in conjunction with the ***SAV** command. The synthesizer provides one register (register 0) for storing synthesizers setup information, therefore only 0 should be used as an argument for this command.

For example, to recall the synthesizer setting stored in register 0, send the command:

***RCL 0**

Related Commands:

***SAV** (Save Command)

***RST** (Reset Command)

→ *RST →

This command sets all user programmable variables to default values.

It does not affect VXI interface, output queue, status register, standard event enable register, or calibration data.

***SAV** (Save Command)

Range: 0

→ *SAV — sp — <NR1> →

This command causes the current synthesizer setup parameters to be saved in non-volatile memory in a setup register designated register 0. The synthesizer provides one register (register 0) for saving synthesizers setup information, therefore only 0 should be used as an argument for this command.

For example, to save the current synthesizer setup parameters in register 0, send the command:

***SAV 0**

Related Commands:

***RCL** (Recall Command)

***SRE** (Service Request Enable Command)

Range: 0 through 255
Default: 0

→ *SRE — sp — <NRf> →

This command sets the Service Request Enable register (Status Byte Enable) with bit 6 ignored. Refer to the Status subsystem section of the manual for more information.

***SRE?** (Service Request Enable Query)

Range: 0 through 255

→ *SRE — ? →

This query returns the current bit values of the Service Request Enable register. Bit 6 is always 0. Refer to the Status subsystem section of the manual for more information.

***STB?** (Read Status Byte Query)

Range: 0 through 255

→ *STB — ? →

This query returns the status byte. Master Summary Status is returned in bit 6.

***TRG** (Trigger Command)

→ *TRG →

This command triggers the synthesizer if **TRIG:SOUR** is BUS and the synthesizer is waiting for a trigger.

***TST?** (Self-test Query)

→ *TST → ? →

This query causes the synthesizer to run an internal self-test and returns the unit back to default settings. If all tests pass, a 0 will be returned. If the internal self-test fails, a number corresponding to the fault detected will be returned. To determine the fault(s) detected, convert the string returned to its numeric value and then to the corresponding binary bit pattern. The fault(s) are determined by the bits set (1's) in the binary bit pattern per the following list.

Weight	
1	bit 0: RAM fault.
2	bit 1: Stored Power calibration memory checksum fault.
4	bit 2: System voltage fault.
8	bit 3: Stored YIG calibration data fault.
16	bit 4: Stored power calibration data fault.
32	bit 5: YIG Tuned Oscillator (YTO) fault.

For example, if the ***TST** query returns a 9, it indicates a self-test failure. A "9" corresponds to a binary bit pattern of 01001. This would indicate RAM fault and a problem with the YIG calibration data.

If a system voltage fault is detected, the **:STAT:QUES:VOLT?** query can be used to determine which voltage is causing the fault. Refer to the Status subsystem of the manual for more information.

If a YIG or power calibration data fault is detected, the **:STAT:QUES:CAL:COND?** query can be used to get more information on the nature of the fault. Refer to the Status subsystem of the manual for more information.

Self-test failures indicate the unit is non-functional and needs to be repaired. Refer to Section 2 of the manual for Service information.

***WAI** (Wait-to-Continue Command)

→ *WAI →

This command prevents the unit from executing any further commands or queries until all previously sent commands are finished.

INSTRUMENT DEPENDENT COMMANDS

Instrument dependent commands for the synthesizer are based on conventions of the SCPI standard. These commands follow a hierarchical structure similar to the file structure found on most computers. The SCPI standard refers to the command structure as the command tree. The keyword closest to the top is called the root. In building specific commands, the root keyword appears first and is joined to the ascending keywords in the path to reach the lower levels. The first layer of the command tree for the synthesizer is shown below. The keywords listed under the root are known as subsystems.

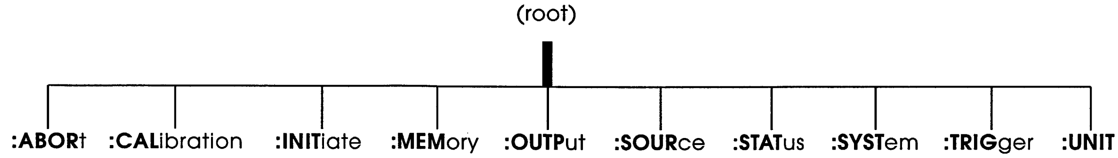


Figure 4-1. Root Command Tree.

In developing specific commands for the synthesizer, the first element of the command would be one of the sub-system keywords shown above. They would be joined to one of the commands in the ascending path of the particular subsystems.

For example, suppose you are developing a command to turn on the RF output power at the front panel of the synthesizer. Since this operation deals with an output from the synthesizer, it would be listed under the Output Subsystem. The complete command tree for the Output Subsystem is shown below.

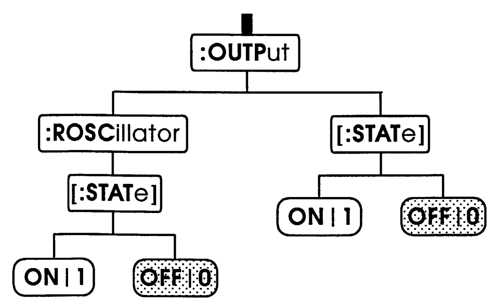


Figure 4-2. Sample Subsystem Command Tree.

To build the command to turn on the RF output power from the synthesizer, you would begin with the OUTPut subsystem keyword and append keywords shown in the path, using the colon, to develop the specific command. For example, to turn on the RF output power at the front panel of the synthesizer, you would send the following command:

:OUTPut:STATe ON

The word "ON" in the command is called the argument of the command. White space must appear between the command and its argument. To connect the internal reference oscillator to the front panel output connector, you would send the command:

:OUTPut:ROSCillator:STATe ON

The colons shown preceding each of the keywords are not actually part of the command. They tell the synthesizer firmware that the keyword following the colon is one level down in the tree structure from the keyword preceding the colon. The colon preceding the OUTPut keyword is not actually required when sending a single command because the OUTPut keyword appears at the root level of the command tree structure. However, in general, the colon is shown preceding all keywords to prevent confusion when multiple commands are joined together in a single message.

At times it is desirable to combine more than one command in a single message. This is accomplished using a semicolon to separate each command. The semicolon is not actually part of the command. It tells the synthesizer firmware that the command following the semicolon is a new command. For example, to connect a sample of the internal timebase to the front panel reference output connector and to turn on the RF at the front panel RF output connector, send the message:

:OUTPut**:ROSC**illator**:STAT**e **ON;:OUTP**ut**:STAT**e **ON**

This is a good example of the function of the colon preceding the OUTPut keyword. If it was left out, as shown below, an error would be generated.

:OUTPut**:ROSC**illator**:STAT**e **ON;OUTP**ut**:STAT**e **ON**

When the command parser (the synthesizer firmware which decodes the commands) gets to the semicolon, it knows the keyword following is a new command, but since no colon precedes the OUTPut keyword, the command parser expects the OUTPut keyword to be at the current level in the command tree. When the command parser discovers that no OUTPut keyword exists at the current level, a syntax error would be generated and the command would be ignored. For this reason, it is generally a good practice to precede even subsystem keywords with the colon.

COMMAND PUNCTUATION SUMMARY

COLON - A colon flags the synthesizer firmware to move down one level in the current path.

SEMICOLON - A semicolon flags the synthesizer firmware that a new command will follow.

WHITE SPACE - White space (one or more space characters) is normally ignored, but keywords cannot contain embedded white space. Also, white space is required between keywords and their arguments.

COMMA - A comma is used to separate multiple command arguments.

ABORT SUBSYSTEM

Figure 4-3. Abort Subsystem Command Tree.

The Abort Subsystem provides control over the Trigger Subsystem. It consists of only one command which is used to force each of the four trigger sequences back to the Idle state upon completion of current :SOURce:LIST or :SWEep<n>:STEP.

Example:

To command the synthesizer to return to the Idle state immediately upon completion of current :SOURce:LIST or :SWEep<n>:STEP, send the command

:ABORt

Related Commands:

See Trigger Subsystem.

CALIBRATION SUBSYSTEM

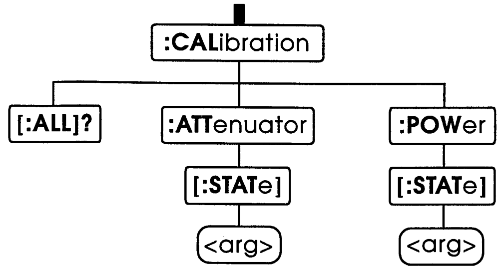


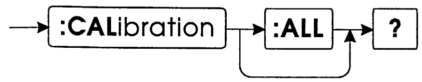
Figure 4-4. Calibration Subsystem Command Tree.

The Calibration Subsystem includes routines for developing and controlling synthesizer calibration tables. Commands in this subsystem are used to perform routine calibration on the YIG-tuned oscillator and YIG-tuned filter. Commands are also included to enable and disable the calibration tables for the power leveling loop and microwave attenuator.

It is recommended that calibration of the YIG-tuned oscillator and YIG-tuned filter be performed when the synthesizer is first installed in the mainframe or whenever it is removed and reinstalled.

The synthesizer will default to using the calibration tables for both the leveling loop and the microwave attenuator. During normal operation, these tables should not be disabled.

CALibration:ALL?



This is a fully automatic calibration routine which requires no special setup, but DO NOT run the calibration routine until the synthesizer has been allowed to warm up for at least 15 minutes. It is recommended that this calibration be performed when the synthesizer is first installed in the mainframe or whenever it is removed and reinstalled. During this routine, correction tables are created for the YIG-tuned filter and YIG-tuned oscillator. If the calibration is successfully completed, a 0 will be returned. If the calibration routine fails, a number will be returned indicating the cause of the failure. To determine why the calibration failed, convert the returned string to its numeric value and then to the corresponding binary bit pattern. The reason for failure can be determined by the bits set (1's) in the binary bit pattern per the following list.

Weight

1 bit 0: YTO calibration frequencies missing (EEPROM fault). This failure will prevent the routine from being attempted.

2 bit 1: Invalid RAM YIG calibration constants. Indicates that calibration was attempted, but data returned was invalid.

4 bit 2: Invalid stored YIG calibration constants. Indicates that calibration was completed, but writing data failed.

8 bit 3: Power-up YIG initialization not completed. This failure will prevent the routine from being attempted.

16 bit 4: System voltage fault detected. This failure will prevent routine from being attempted.

For example, if the **:CAL:ALL?** query returns 1, it indicates that calibration failed. A "1" corresponds to a binary bit pattern of 0001. This would indicate that YTO calibration frequencies were missing from the EEPROM.

Calibration failures indicate the unit is non-functional and needs to be repaired. Refer to Section 2 of the manual for Service information.

Example:

To start self-calibration, send the command

:CAL:ALL?

CAUTION

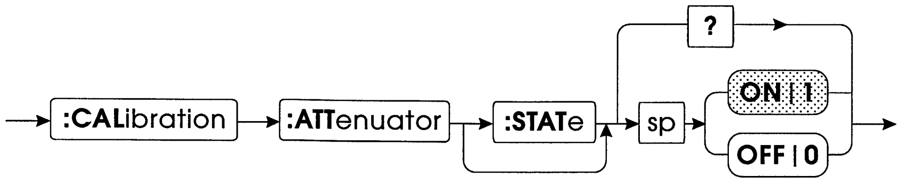
Do not attempt to calibrate until unit has been allowed to warm up for a full 15 minutes.

NOTE

It takes approximately one minute to perform this calibration.

CALibration:ATTenuator:STATe

Default: ON



The state of this command determines whether the attenuator calibration table will be applied. Turning off the attenuator calibration has no effect on the leveling loop calibration table. By default, the attenuator calibration table is enabled; it is only disabled during calibration and some other special circumstances.

Examples:

1. To enable the attenuator calibration table, send the command

   :CALibration**:ATT**enuator**:STAT**e **ON**

   or

   :CALibration**:ATT**enuator**:STAT**e **1**

2. To disable the attenuator calibration table, send the command

   :CALibration**:ATT**enuator **OFF**

   or

   :CALibration**:ATT**enuator **0**

3. To query the instrument for the state of the attenuator calibration table, send the command

   :CALibration**:ATT**enuator**?**

   After receiving the query command, a read will return the current state of the attenuator power calibration table in the <bNR1> format. For example, if the calibration table is enabled and the synthesizer is queried for the current state, it should return

   1

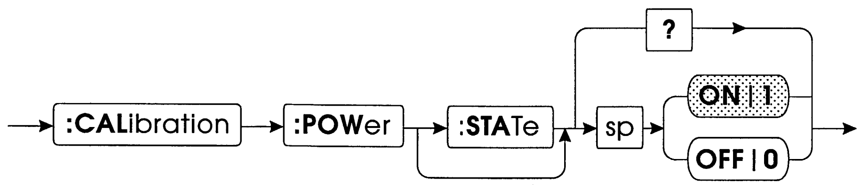
   indicating the attenuator calibration table is enabled. If a 0 is returned then the attenuator calibration table is disabled.

Related Command:

:CALibration**:POW**er**:STAT**e

CALibration:POWer:STATe

Default: ON



The state of this command determines whether power calibration tables will be applied. The power calibration tables consist of a table for the power leveling loop and a table for the internal attenuator. Turning off the power calibration disables both of these tables. The use of the calibration tables defaults to on (enabled); they are only turned off during calibration and some other special circumstances.

Examples:

1. To use the stored calibration tables, send the command

   :CALibration**:POW**er**:STAT**e **ON**

   or

   :CALibration**:POW**er**:STAT**e **1**

2. To disable the calibration tables, send the command

   :CALibration**:POW**er **OFF**

   or

   :CALibration**:POW**er **0**

3. To query the instrument for the state of the calibration tables, send the command

   :CALibration**:POW**er**?**

   After receiving the query command, a read will return the current state of the power calibration tables in the <bNR1> format. For example, if the calibration tables are enabled and the synthesizer is queried for the current state, it should return

   1

   indicating the power calibration tables are enabled. If a 0 is returned then the power calibration tables are disabled.

Related Command:

:CALibration**:ATT**enuator**:STAT**e

NOTE

Disabling the power calibration tables will cause bit 1 in the Questionable Power event register to be set.

INITIATE SUBSYSTEM

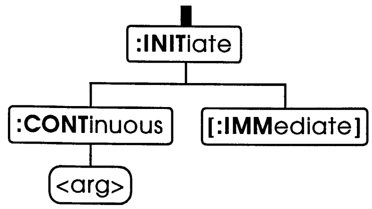


Figure 4-5. Initiate Subsystem Command Tree.

The synthesizer uses a three layer trigger model consisting of Idle, Armed, and Triggered states. The Idle state is the default state and the least active; in this state, the synthesizer is simply waiting to be commanded to the Armed state. The Armed state is where trigger detection occurs. Once triggered, the synthesizer moves from the Armed state to the Triggered state and initiates the programmed action, such as changing the output frequency.

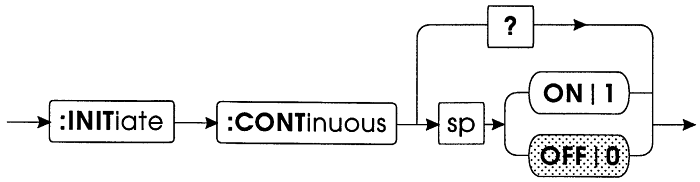
The Initiate Subsystem includes commands to control the movement from the Idle state to the Armed state. For example, the Initiate subsystem commands are used to select what causes the synthesizer to move from the Idle state to the Armed state.

Related Commands:

See Trigger Subsystem.

INITiate:CONTinuous

Default: OFF



The synthesizer uses a three layer trigger model consisting of Idle Armed, and Triggered states. This command controls the movement of the synthesizer from the Idle state to the Armed state. It defaults to off, but when turned on, the synthesizer will bypass the Idle state and move directly into the Armed state to wait for the trigger to occur. When turned on, this command effectively removes the Idle state from the trigger model causing the trigger system to continuously restart.

Examples:

1. To command the synthesizer to automatically move through the Idle state to the Armed state, send the command

   :INITiate**:CONT**inuous **ON**

2. To command the synthesizer back to the default mode, send the command

   :INITiate**:CONT**inuous **OFF**

3. To query the synthesizer for the current state, send the command

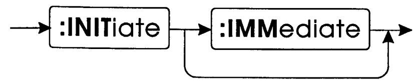
   :INITiate**:CONT**inuous**?**

   After receiving the query command, a read will return the state in the <bNR1> format. For example, if it is currently in the default state and the synthesizer is queried for the current state, it should return

   0

   indicating the synthesizer is set up to remain in the Idle state until specifically commanded to the Armed state.

INITiate:IMMediate



This command immediately moves the synthesizer out of the Idle state and into the Armed state. If the synthesizer is not idle or if **:INIT:CONT** is ON, then it is not valid and the synthesizer will return error -213.

Example:

To command the synthesizer to move immediately from the Idle state to the Armed state, send the command

:INITiate**:IMM**ediate

or just

:INIT

Related Commands:

See Trigger Subsystem.

MEMORY SUBSYSTEM

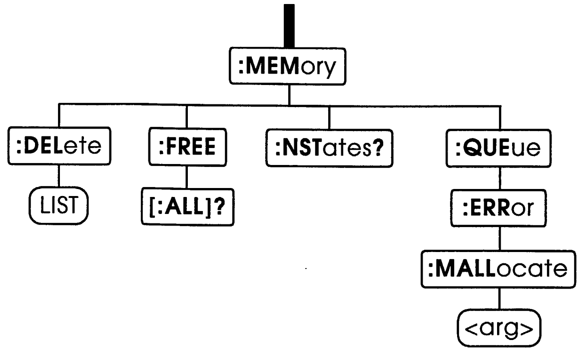


Figure 4-6. Memory Subsystem Command Tree.

The EIP 1140A, 1141A, and 1142A synthesizers include internal memory used to store a synthesizer setup, Triggered List data, and error messages. The Memory Subsystem contains commands to control how this memory is allocated. During program development, it may be more useful to allocate a large portion of the memory for storing error messages. Once a program is debugged, the memory could be reallocated to increase the number of possible Trigger List data points.

Related Commands:

See Trigger List.

MEMory:DELete LIST

→ [**:MEM**ory] → [**:DEL**ete] → [sp] → (**LIST**) →

The synthesizer provides a Triggered List feature which allows it to step through a list of up to 201 points. Each of these points may differ in frequency, power, and/or modulation. The **:MEM**ory**:DEL**ete **LIST** command deletes any existing Triggered List data and deallocates the memory previously allocated to the list. This command should be executed prior to entering new Triggered List data.

Example:

To delete any existing Triggered List data and deallocate the memory, send the command

:MEMory**:DEL**ete **LIST**

This command does not support a query. Sending the command as a query will generate an error message.

Related Command:

:MEMory **FREE:ALL?**

MEMory:FREE:ALL?

This query command causes the synthesizer to return the number of RAM memory bytes available for storing a synthesizer setup, Trigger List data, and error messages.

Example:

To query the synthesizer the number of bytes of RAM currently available, send the command

:MEMory**:FREE:ALL?**

After receiving the query command, a read will return the number of bytes available in the <NR1> format. For example, a typical response to the **:MEM:FREE:ALL?** query is

4461

indicating that 4461 bytes of RAM are currently available.

Related Command:

:MEMory**:DEL**ete**:LIST**

MEMory:NSTates?

→ [:MEMory] → [:NSTates] → [?]

This query command causes the synthesizer to return the number of synthesizer setup states available. The command is not normally used because the synthesizer provides only one location, or state, for storing synthesizer setup. It is included to maintain compatibility with the SCPI standard.

Example:

To query the synthesizer for the number of synthesizer setup states available, send the command

:MEMory**:NST**ates**?**

After receiving the query command, a read will return the number one in the <NR1> format.

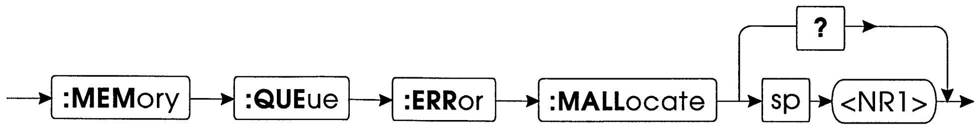
Related Commands:

***SAV**

***RCL**

MEMory:QUEue:ERRor:MALLocate

Default: 30
Range: 10 to 500



This command sets the depth of the SCPI error queue in number of messages. Each message takes one byte of RAM memory. The changes in the depth of the error queue take effect only after a synthesizer reset.

Examples:

1. To set the error queue to 15 messages, send the command

   :MEMory**:QUE**ue**:ERR**or**:MALL**ocate **15**

2. To set the error queue to 8 messages, send the command

   :MEMory**:QUE**ue**:ERR**or**:MALL**ocate **8**

3. To query the synthesizer for the number of messages in the error queue, send the command

   :MEMory**:QUE**ue**:ERR**or**:MALL**ocate**?**

   After receiving the query command, a read will return the depth of the error queue in the <NR1> format. For example, a typical response to this query is

   30

   indicating the SCPI error queue will hold up to 30 messages.

Related Command:

:MEMory**:FREE?**

OUTPUT SUBSYSTEM

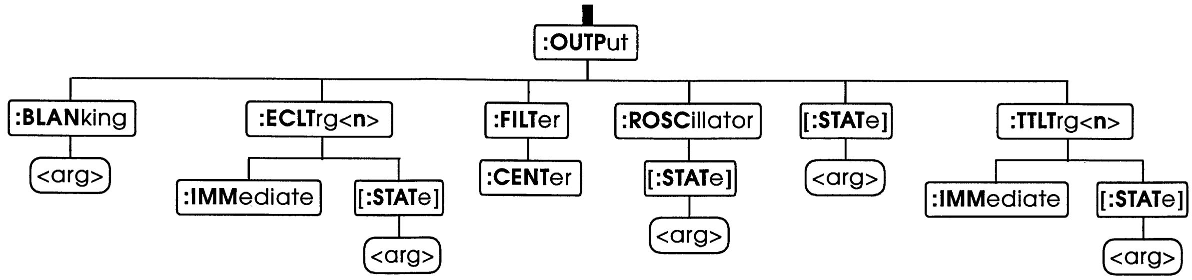
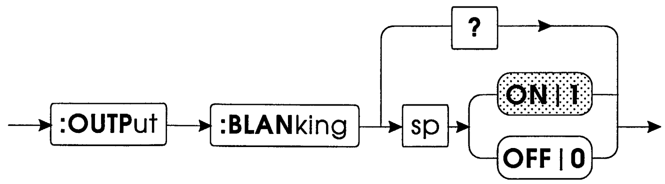


Figure 4-7. Output Subsystem Command Tree.

The Output Subsystem controls the output ports on the synthesizer. At power on, all outputs default to off and must be turned on individually.

OUTPut:BLANking

Default: ON



The RF output from the synthesizer is normally turned off (blanked) during frequency and power changes. The **:OUTP**ut**:BLAN**king command is used to control output blanking.

Examples:

1. To disable output signal blanking, forcing the synthesizer to leave the output power on during frequency and power changes, send the command

   :OUTPut**:BLAN**king **OFF**

2. To enable output signal blanking, send the command

   :OUTPut**:BLAN**king **ON**

3. To query the synthesizer for the current state of output blanking, send the command

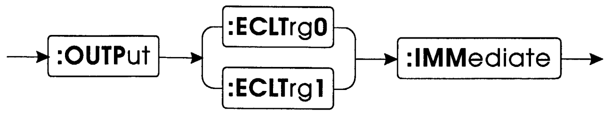
   :OUTPut**:BLAN**king**?**

   After receiving the query command, a read will return the state of output blanking in the <bNR1> format. For example, if blanking is enabled and the synthesizer is queried for the current state of blanking it should return

   1

   indicating that blanking is enabled (ON).

OUTPut:ECLTrg<n>:IMMediate



This command causes the synthesizer to immediately output an ECL pulse, approximately 400 ns wide, to the VXI backplane on the selected ECL Trigger line (ECLTrg0 or ECLTrg1).

Examples:

1. To immediately output an ECL trigger pulse on ECLTrg0, send the command

   :OUTPut**:ECLT**rg**0:IMM**ediate

2. To immediately output an ECL trigger pulse on ECLTrg1, send the command

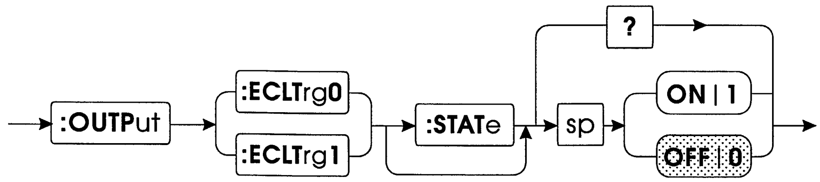
   :OUTPut**:ECLT**rg**1:IMM**ediate

Related Command:

:OUTPut**:ECLT**rg<**n**>**:STAT**e

OUTPut:ECLTrg<n>:STATe

Default: OFF



This command enables one of the two ECL triggers (0...1). The synchronous protocol is implemented on the synthesizer. The synchronous trigger protocol is a single line broadcast trigger that does not require any acknowledgment from the acceptors. The synthesizer can be programmed to initiate a trigger upon command or upon completion of any signal change, such as a change in frequency or power.

Examples:

1. To enable ECLTrg0, send the command

   :OUTPut:ECLTrg0:STATe ON

   or

   :OUTP:ECLTrg0 ON

2. To enable ECLTrg1, send the command

   :OUTPut:ECLTrg1:STATe ON

3. To query the synthesizer for the current state of ECL trigger 1, send the command

   :OUTPut:ECLTrg1:STATe?

   After receiving the query command, a read will return the state of the selected ECL trigger line in the <bNR1> format. For example, if ECLTrg was previously enabled and the synthesizer is queried for the current state of ECLTrg1, it should return

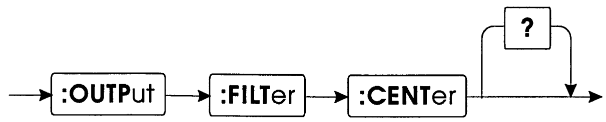
   1

   indicating that ECLTrg1 is enabled (ON). This does not mean the trigger line is active, only that it is enabled.

Related Command:

:OUTPut:ECLTrg<n>:IMMediate

OUTPut:FILTer:CENTer



This command causes the YIG-tuned bandpass filter (YTF) inside the synthesizer to center on the output signal. The YIG-tuned bandpass filter (YTF) is used to eliminate (filter out) all signals other than the primary output signal. This filter has a nominal 3 dB bandwidth of 50 MHz and is normally centered on the output signal based solely on the calibration data which correlates the center frequency of the YTF to the tuning signal. Typically, this approach is more than adequate for centering the YTF on the output signal. However, if more precise centering is required, the filter center command is used. After receiving this command, the YTF is moved around the signal until the exact peak is found, and the filter is centered on the peak.

To control the center frequency of the YTF, the microprocessor, inside the synthesizer, sends a digital word to a digital-to-analog converter (DAC). The output from the DAC directly controls the center frequency of the YTF. After receiving the filter center command, sending the query form of the command causes the synthesizer to output the frequency shift that occurred as a result of the filter centering routine.

Examples:

1. To command the synthesizer to manually center the YTF on the output signal, send the command

   :OUTPut**:FILT**er**:CENT**er

2. To find out how far the filter was moved in response to the filter center command, send the query command

   :OUTPut**:FILT**er**:CENT**er**?**

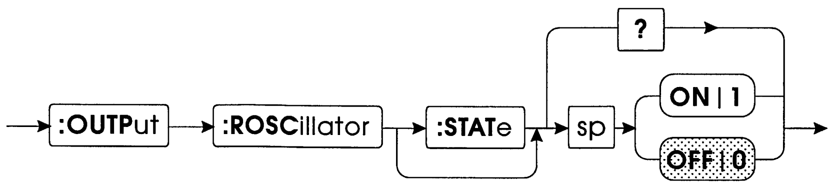
   After receiving the query command, a read will return the frequency shift in the <NR1> format. For example, if the filter center command caused the center frequency of the filter to increase by 1 MHz then the query should return

   1000000

   indicating a positive shift of 1 MHz in the center frequency of the YTF. If the new center frequency is less than the original center frequency, then the frequency shift will be negative.

OUTPut:ROSCillator:STATe

Default: OFF



This command controls the direction of the front panel INPUT/OUTPUT 10 MHz connector. Turning it on configures it to be an output; turning it off configures it to be an input.

Examples:

1. To make the front panel 10 MHz connector an output connector, send the command

   :OUTPut**:ROSC**illator**:STAT**e **ON**

2. To make the front panel 10 MHz connector an input connector, send the command

   :OUTPut**:ROSC**illator **OFF**

3. To query the synthesizer for the status of the the front panel 10 MHz connector, send the command

   :OUTPut**:ROSC**illator**?**

   After receiving the query command, a read will return the state of the reference oscillator in the <bNR1> format. For example, if it was previously set to ON and the synthesizer is queried for the current state, it should return

   1

   indicating the 10 MHz INPUT/OUTPUT connector on the front panel is configured as an output.

NOTE:

Before configuring the 10 MHz front panel connector as an output (**:OUTP:ROSC ON**), make sure the reference oscillator source is set to internal (**:SOUR:ROSC:SOUR INT**). If this is not done, a settings conflict error will occur.

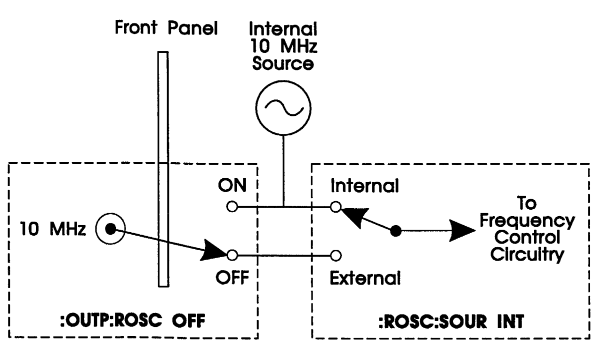
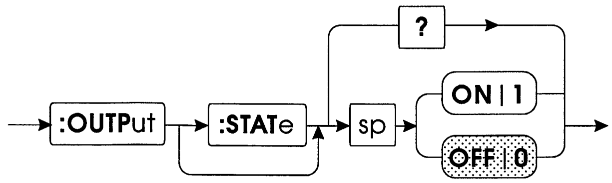


Figure 4-8. Reference Oscillator Selection Circuit.

OUTPut:STATe

Default: OFF



This command controls the state of the RF OUTPUT connector on the front panel. When enabled (ON), RF power is applied to the front panel connector.

Examples:

1. To enable the RF OUTPUT connector on the front panel, send the command

   :OUTPut**:STAT**e **ON**

   or

   :OUTP ON

2. To turn off the RF power at the front panel, send the command

   :OUTPut **:STAT**e **OFF**

   or

   :OUTP OFF

3. To query the synthesizer for the current state of the RF output, send the command

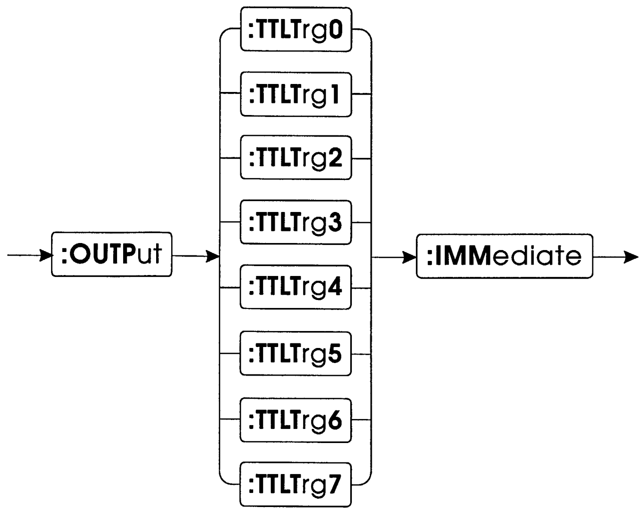
   :OUTPut**:STAT**e**?**

   After receiving the query command, a read will return the state of the front panel RF OUTPUT connector in the <bNR1> format. For example, if RF output was previously enabled and the synthesizer is queried for the current state, it should return

   1

   indicating the RF output is enabled (ON).

OUTPut:TTLTrg<n>:IMMediate



This command causes the synthesizer to immediately output an active high TTL pulse, approximately 400 ns wide, to the VXI backplane on one of the 8 TTL trigger lines (TTLTrg0...TTLTrg7). Before sending this command, the desired TTL trigger must be enabled using the **:OUTP**ut**:TTLT**rg<**n**>**:STAT**e command.

Examples:

1. To immediately output a TTL trigger pulse on TTLTrg0, send the command

   :OUTPut**:TTLT**rg**0:IMM**ediate

2. To immediately output a TTL trigger pulse on TTLTrg7, send the command

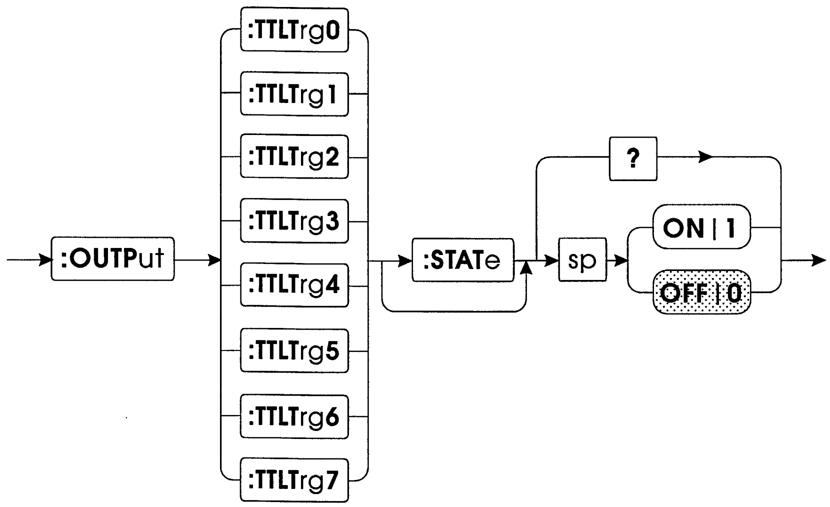
   :OUTPut**:TTLT**rg**7:IMM**ediate

Related Command:

:OUTPut**:TTLT**rg<**n**>**:STAT**e

OUTPut:TTLTrg<n>:STATe

Default: OFF



This command enables one of the eight TTL triggers (TTLTrg0...TTLTrg7). The synchronous protocol is implemented on the synthesizer. The synchronous trigger protocol is a single line broadcast trigger that does not require an acknowledge from any acceptors. The synthesizer can be programmed to initiate a trigger upon command or upon completion of any signal change, such as a change in frequency or power.

Examples:

1. To enable TTLTrg0, send the command

   :OUTPut:TTLTrg0:STATe ON

   or

   :OUTP:TTLTrg0 ON

2. To enable TTLTrg1, send the command

   :OUTPut:TTLTrg1:STATe ON

3. To query the synthesizer for the current state of TTL trigger 1, send the command

   :OUTPut:TTLTrg1:STATe?

   After receiving the query command, a read will return the state of the selected TTL trigger line in the <bNR1> format. For example, if TTLTrg was previously enabled and the synthesizer is queried for the current state of TTLTrg1, it should return

   1

   indicating that TTLTrg1 is enabled (ON). This does not mean the trigger line is active, only that it is enabled.

Related Command:

:OUTPut:TTLTrg<n>:IMMediate

SOURCE SUBSYSTEM

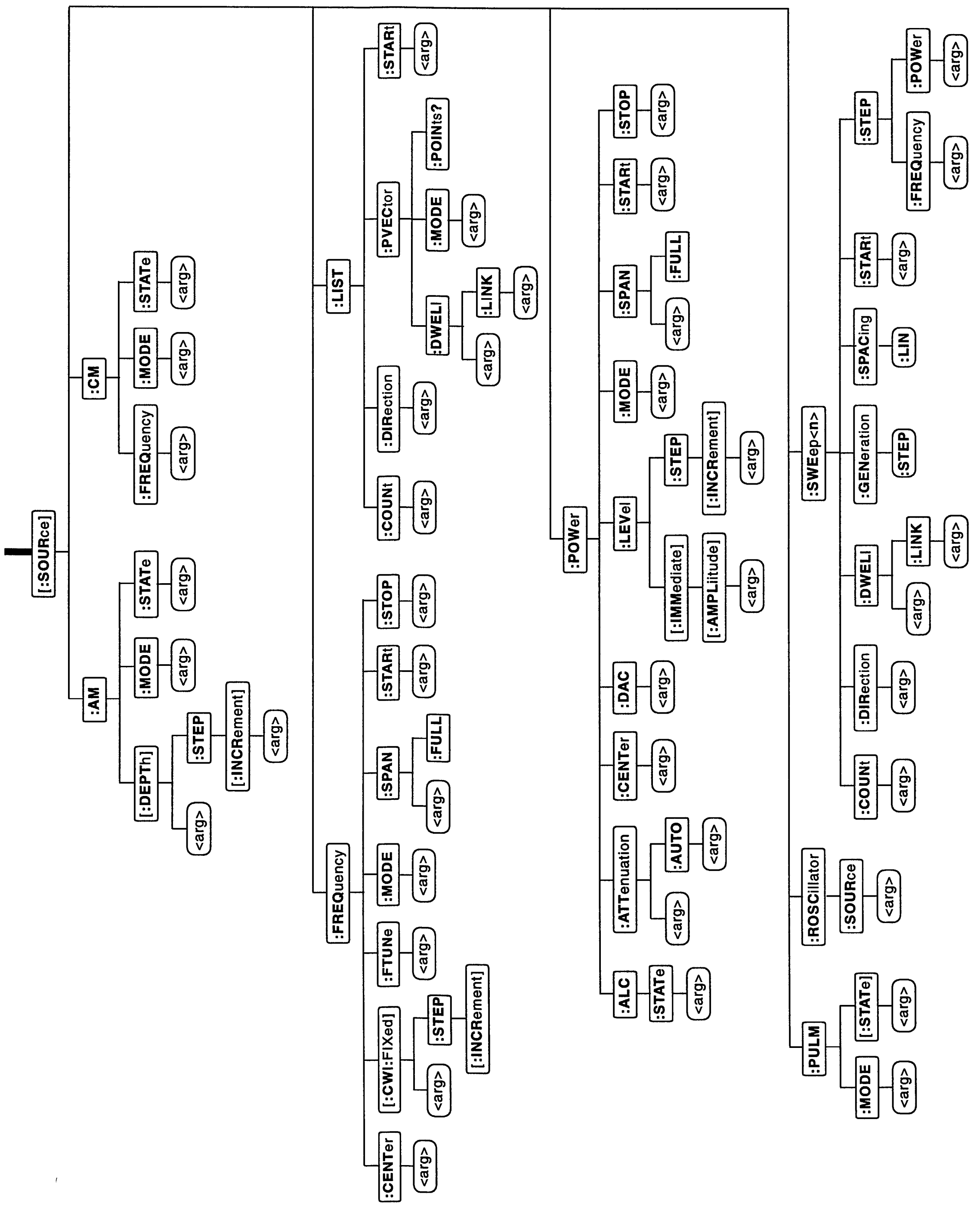
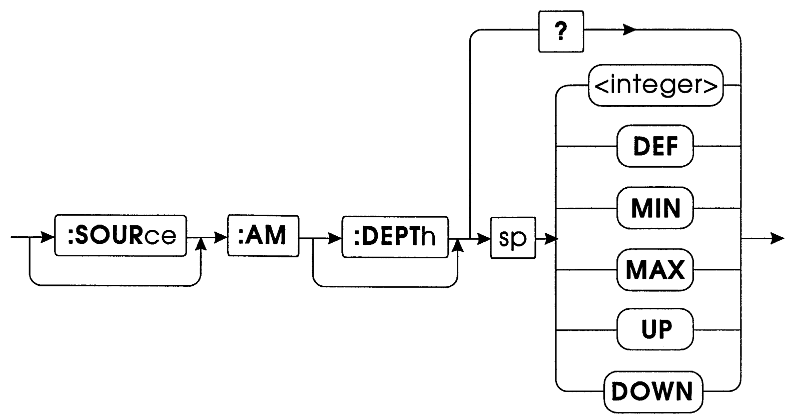


Figure 4-9. Source Subsystem Command Tree.

The Source Subsystem provides control of a variety of synthesizer source settings. The Source Subsystem is used to control output frequency, power, modulation, and to select the source of the reference oscillator used by the synthesizer.

SOURce:AM:DEPTh

Default: 0
Range: 0 to 100%



This command sets the depth of an amplitude modulation signal. The, optional, units for depth is percent (%).

Examples:

1. To set the amplitude modulation depth to 50, send the command

   :SOURce:AM:DEPTh 50

   or

   :AM 50%

2. To query the synthesizer for the current setting of the amplitude modulation depth, send the command

   :SOURce:AM:DEPTh?

   or

   :AM?

   After receiving the query command, a read will return the current amplitude modulation depth setting in the <NR1> format. For example, if the amplitude modulation depth was set to 40% and the synthesizer is queried for the current amplitude modulation depth, it should return

   40

   indicating the amplitude modulation depth is set to 40%.

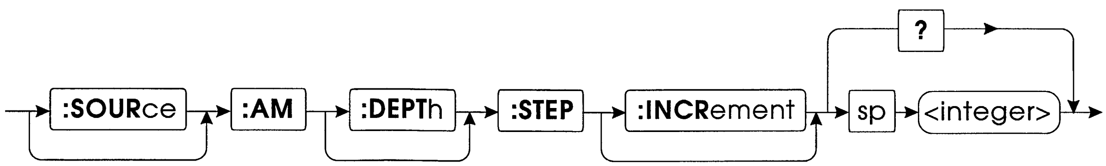
Related Commands:

:SOURce:AM:DEPTh:STEP:INCRement

:SOURce:AM:STATe

SOURce:AM:DEPTh:STEP:INCRement

Default: 0
Range: 0 to 100%
Resolution: 0.1 dB



This command sets the increment step size for stepping the amplitude modulation. The, optional, units associated with this command is percent (%). The increment value can be either positive or negative and no range checking is performed.

Examples:

1. To set the amplitude modulation step size to 10 percent, send the command

   :SOURce**:AM:DEPT**h**:STEP:INCR**ement **10**

   or

   :AM:STEP 10%

2. To query the synthesizer for the current amplitude modulation step size, send the command

   :SOURce**:AM:DEPT**h**:STEP:INCR**ement**?**

   or

   :AM:STEP?

   After receiving the query command, a read will return the current amplitude modulation step setting in the <NR1> format. For example, if the amplitude modulation step was set to 40% and the synthesizer is queried for the current amplitude modulation step size, it should return

   40

   indicating the amplitude modulation step is set to 40%.

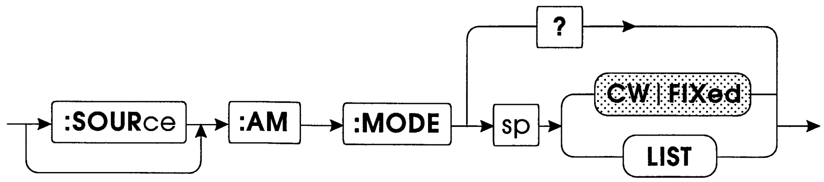
Related Commands:

:SOURce**:AM:DEPT**h

:SOURce**:AM:STAT**e

SOURce:AM:MODE

Default: CW|FIXed



This command couples control of the state of amplitude modulation to either the FIXed or LIST mode. It is normally in the FIXed mode which means that amplitude modulation can be turned on and off using the **:SOUR**ce**:AM:STAT**e command. In the LIST mode, the state of amplitude modulation is controlled by parameters set using the **:SOUR**ce**:LIST** commands.

The command is included to provide compatibility with SCPI defined syntax and is fully functional, but is not typically used. The command **:SOUR**ce**:LIST:PVEC**tor**:MODE LIST** is normally used for coupling to the LIST mode because it couples AM as well as: CM, FREQ, POWer, and PULM.

Examples:

1. To couple amplitude modulation to the LIST mode, send the command

   :SOURce**:AM:MODE LIST**

2. To query the synthesizer for the current mode setting, send the command

   :SOURce**:AM:MODE?**

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if the AM mode was previously set to the LIST mode and the synthesizer is queried for the current mode, it should return

   LIST

   indicating the state of amplitude modulation is currently being controlled by the triggered list function.

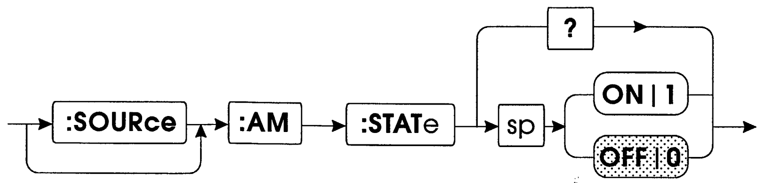
Related Commands:

:SOURce**:AM:STAT**e

:SOURce**:LIST:PVEC**tor**:MODE LIST**

SOURce:AM:STATe

Default: OFF



This command turns amplitude modulation on or off.

Examples:

1. To enable the amplitude modulation input, send the command

   :SOURce:AM:STATe ON

   or

   :SOURce:AM:STATe 1

2. To disable the amplitude modulation input, send the command

   :AM:STATe OFF

   or

   :AM:STATe 0

3. To query the synthesizer for the current amplitude modulation state, send the command

   :AM:STATe?

   After receiving the query command, a read will return the state of amplitude modulation in the <bNR1> format. For example, if the amplitude modulation was previously enabled and the synthesizer is queried for the current state, it should return

   1

   indicating that amplitude modulation is enabled (ON).

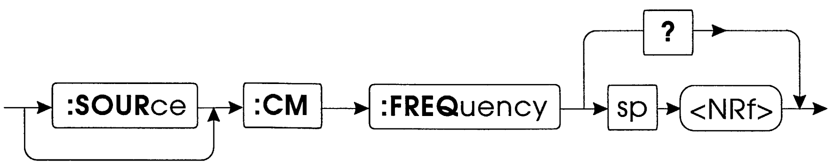
Related Commands:

:SOURce:AM:DEPTh

:SOURce:AM:DEPTh:STEP:INCRement

SOURce:CM:FREQuency

Default: 300 MHz
Range: 300 MHz to 1 GHz
Resolution: 1 Hz



The synthesizer provides an IF input used for superimposing modulation on the synthesized signal. The IF input signal can vary in both frequency and amplitude combining various types of modulation. This signal is applied to one input on a mixer. The other input to the mixer is the synthesized signal. The combined signals at the output from the mixer are applied to an electronically tunable microwave filter, labeled YTF/AMP in the block diagrams of the synthesizers shown in Appendix C. The microwave filter is a bandpass YIG-tuned filter with a nominal 3 dB bandwidth of 50 MHz.

The center frequency of the signal applied to the bandpass filter (YTF) is a function of the signal applied at the IF input and the frequency setting of the synthesizer.

To properly set the center frequency of the bandpass filter, the synthesizer must know the center frequency of the signal to be applied to the IF input. The **:SOURce:CM:FREQ**uency command is used for this purpose. The synthesizer adds the supplied IF frequency to the current frequency setting and adjusts the bandpass filter to that frequency. For example, if the IF input frequency were set to 700 MHz and the synthesizer frequency was 6 GHz then the synthesizer would adjust the YIG-tuned bandpass filter to 6.7 GHz.

The expected IF frequency must be set prior to enabling complex modulation, using the **:SOURce:CM:STAT**e command. If it is not, then the IF frequency setting at the time when complex modulation is enabled will be used.

Examples:

1. To set the frequency to be applied at the IF input to 500 MHz, send the command

   :SOURce:CM:FREQuency **500 MHZ**

2. To query the synthesizer for the current frequency expected at the IF input, send the command

   :SOURce:CM:FREQuency**?**

   After receiving the query command, a read will return the expected center frequency of the signal applied to the IF input in the <NR1> format. For example, if the complex modulation frequency was set to 500 MHz and the synthesizer is queried for the current expected center frequency, it should return

   500000000

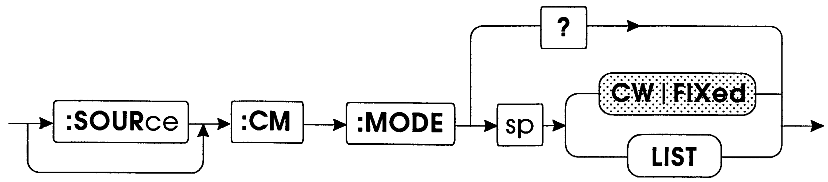
   indicating the complex modulation frequency is set to 500 MHz.

Related Command:

:SOURce:CM:STATe

SOURce:CM:MODE

Default: CW|FIXed



This command couples control of the state of complex modulation to either the FIXed or LIST mode. It is normally in the FIXed mode which means that complex modulation can be turned on and off using the **:SOURce:CM:STATe** command. In the LIST mode, the state of complex modulation is controlled by parameters set using the **:SOURce:LIST** commands.

The command is included to provide compatibility with SCPI defined syntax and is fully functional, but is not typically used. The command **:SOURce:LIST:PVECtor:MODE LIST** is normally used for coupling to the LIST mode because it couples CM as well as AM, FREQ, POWER, and PULM.

Examples:

1. To couple complex modulation to the LIST mode, send the command

   :SOURce:CM:MODE LIST

2. To query the synthesizer for the current mode setting, send the command

   :SOURce:CM:MODE?

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if the CM mode was previously set to the LIST mode and the synthesizer is queried for the current mode, it should return

   LIST

   indicating the state of complex modulation is being controlled by the trigged list function.

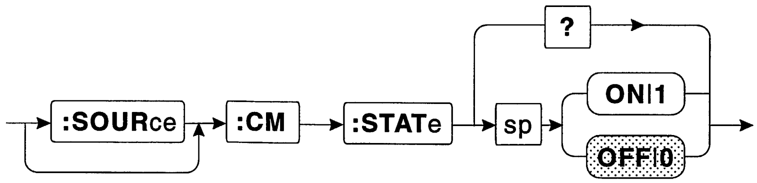
Related Commands:

:SOURce:CM:STATe

:SOURce:LIST:PVECtor:MODE LIST

SOURce:CM:STATe

Default: OFF



This command is used to turn complex modulation on and off. Prior to enabling complex modulation, the center frequency of the signal to be applied at the IF input connector must be set using the **:SOUR**ce**:CM:FREQ**uency command. If it is not, then the synthesizer will use the current setting of the complex modulation frequency for adjusting the center frequency of the bandpass filter.

Examples:

1. To enable complex modulation, send the command

   :SOURce**:CM:STAT**e **ON**

   or

   :SOURce**:CM:STAT**e **1**

2. To disable complex modulation, send the command

   :CM:STATe **OFF**

   or

   :CM:STATe **0**

3. To query the synthesizer for the current state of complex modulation, send the command

   :SOURce**:CM:STAT**e**?**

   After receiving the query command, a read will return the current state of complex modulation in the <bNR1> format. For example, if complex modulation is enabled and the synthesizer were queried for the current state, the read should return

   1

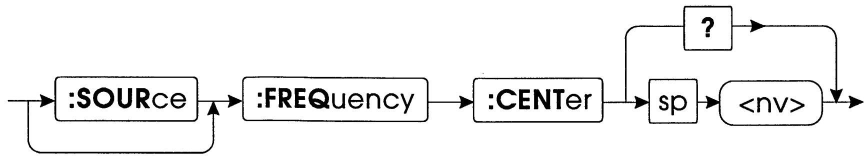
   indicating complex modulation is enabled.

Related Command:

:SOURce**:CM:FREQ**uency

SOURce:FREQuency:CENTer

Default: (MIN+MAX)/2
1140A: 10.005 GHz
1141A: 11 GHz
1142A: 8.2 GHz



This command sets the center frequency for the synthesizer. It defaults to one-half the value of the minimum frequency plus the maximum frequency. Whenever a new center frequency is received, the synthesizer recomputes the start and stop frequencies based on the current frequency span and the new center frequency. If the computed start or stop frequency is out of range, an error message will be generated.

A new center frequency is normally sent along with a new frequency span to reduce the chance of generating an error. For example, to set a center frequency of 7 GHz on an 1142A, send the center frequency along with the new frequency span of 6 GHz. The new frequency span was calculated as follows:

(Center Frequency - Minimum Frequency) x 2 = Frequency Span

(7 GHz - 4 GHz) x 2 = 6 GHz

Examples:

1. To set the center frequency, send both the new center frequency and frequency span that will prevent either the start or stop frequency from going out of band. For example, to set center frequency of 7 GHz and a frequency span of 6 GHz, send the command

   :SOURce:FREQ:CENTer 7 GHz;:SOURce:FREQ:SPAN 6 GHz

   Upon receiving the new center frequency and span, the synthesizer will recompute the corresponding start and stop frequency. For this example, the new start frequency would be 4 GHz and the new stop frequency would be 10 GHz.

2. To query the synthesizer for the current center frequency, send the command

   :SOURce:FREQ:CENT?

   After receiving the query command, a read will return the current mode setting in the <NR1> format. For example, if the 1142A is at the default center frequency and the synthesizer is queried for the current value, it should return

   8200000000

   indicating the current center frequency is 8.2 GHz.

Related Commands:

:SOURce:FREQ:SPAN

:SOURce:FREQ:STARt

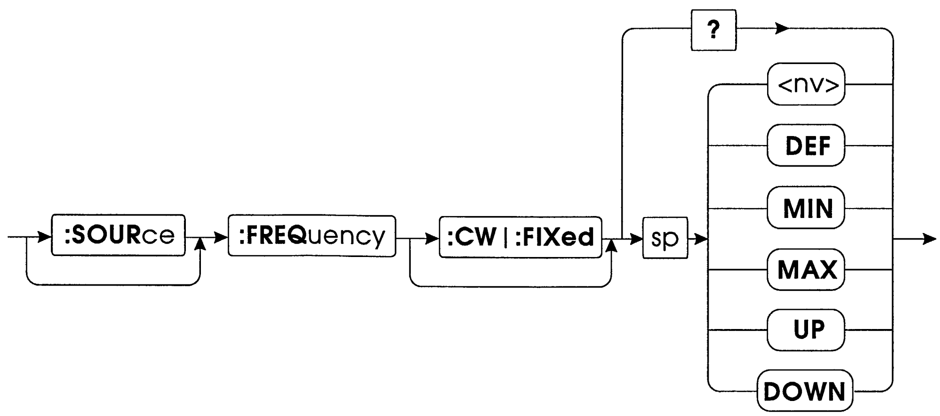
:SOURce:FREQ:STOP

SOURce:FREQuency:CW|FIXed

Default: (MIN + MAX)/2
1140A: 10.005 GHz
1141A: 11 GHz
1142A: 8.2 GHz

Resolution: 1 Hz

Range: 1140A: 0.01 GHz to 20 GHz
1141A: 2 GHz to 20 GHz
1142A: 4 GHz to 12.4 GHz



This command sets the output frequency from the synthesizer. The frequency can be sent to a 1 Hz resolution. The allowable range depends on the synthesizer.

Examples:

1. To select a frequency of 7 GHz, send the command

   :SOURce:FREQuency:CW 7E9

   or

   :FREQuency:FIXed 7 GHZ

2. To query the synthesizer for the current frequency setting, send the command

   :SOURce:FREQuency:CW?

   After receiving the query command, a read will return the current frequency setting in the <NR1> format. For example, if the frequency was set to 11 GHz and the synthesizer is queried for the current frequency, it should return

   11000000000

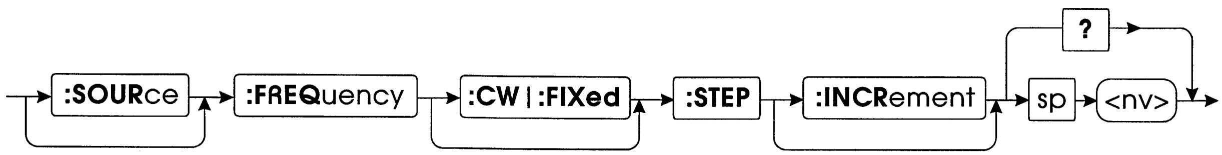
   indicating the frequency is set to 11 GHz.

Related Command:

:SOURce:FREQuency:CW:STEP:INCRement

SOURce:FREQuency:CW|FIXed:STEP:INCRement

Default: 0
Resolution: 1 Hz
Range: 1 Hz to 10 GHz



This command sets the incremental step size for stepping the output frequency from the synthesizer. The increment step size can be either positive or negative.

Examples:

1. To set the frequency step size to 10 kHz, send the command

   :SOURce**:FREQ**uency**:CW:STEP:INCR**ement **10 KHZ**

   or

   :FREQuency**:FIX**ed**:STEP 10 KHZ**

2. To query the synthesizer for the current frequency step size, send the command

   :SOURce**:FREQ**uency**:CW:STEP:INCR**ement**?**

   After receiving the query command, a read will return the current frequency step setting in the <NR1> format. For example, if the frequency step was set to 1 kHz and the synthesizer is queried for the current frequency step, it should return

   1000

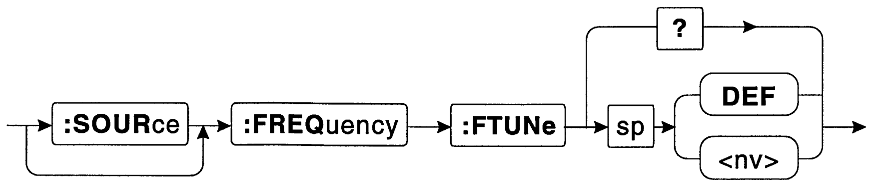
   indicating the frequency step is set to 1000 Hz (1 kHz).

Related Command:

:SOURce**:FREQ**uency**:CW|FIX**ed

SOURce:FREQuency:FTUNe

Default: 0 Hz
Resolution: 1 Hz
Range: ±1 MHz



This command allows the output frequency to be changed over a ±1 MHz range without unlocking the phase-lock-loop in the synthesizer. When output blanking is off, this command can be used to change the frequency over a limited bandwidth without the large frequency transient and power loss which normally occurs during changes in output frequency.

Commanding the synthesizer to a new frequency either directly or via triggered list will reset the fine tune frequency to zero.

Examples:

1. To fine tune the synthesizer 500 kHz above the current CW frequency, send the command

   :SOURce:FREQ:FTUNe 500 KHZ

   Upon receiving the fine tune command, the synthesizer will command the DDS to a new frequency causing the output frequency from the synthesizer to increase by 500 kHz.

2. To query the synthesizer for the current center frequency, send the command

   :SOURce:FREQ:FTUNEe?

   After receiving the query command, a read will return the current mode setting in the <NR1> format. For example, if the fine tune command was used to increase the output frequency from the synthesizer by 500 kHz and the synthesizer is queried for the current value, it should return

   500000

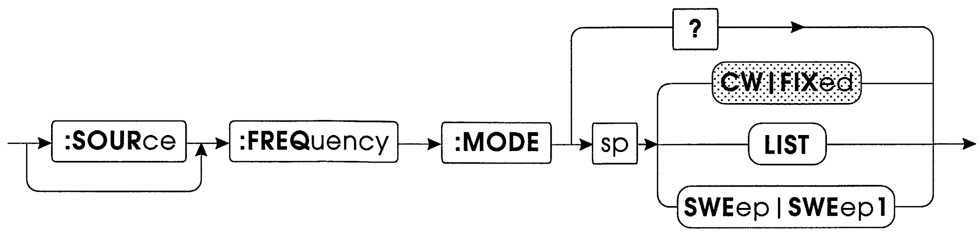
   indicating that current fine tune frequency is 500 kHz.

Related Command:

:SOURce:FREQ:CW

SOURce:FREQuency:MODE

Default: CW|FIXed



This command couples frequency control of the synthesizer to the FIXed, LIST, or SWEep1 mode. It is normally used to couple control of the synthesizer to the SWEep1 mode. It can also be used to couple control of the source frequency to the LIST mode, but is not typically used for this function. The command **:SOURce:LIST:PVECtor:MODE LIST** is a better choice for coupling frequency control to the LIST mode because it couples frequency as well as: AM, CM, POWER, and PULM.

Examples:

1. To couple frequency control to the SWEep1 mode, send the command

   :SOURce:FREQuency:MODE SWEep1

2. To query the synthesizer for the current mode setting, send the command

   :SOURce:FREQuency:MODE?

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if the frequency mode was previously set to SWEep1 and the synthesizer is queried for the current mode, it should return

   SWE

   indicating the output frequency from the synthesizer is currently be controlled by SWEep1.

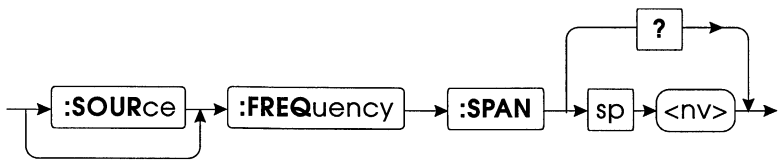
Related Commands:

:SOURce:SWEep1

:SOURce:LIST:PVECtor:MODE LIST

SOURce:FREQuency:SPAN

Default: 1140A: 19.99 GHz
1141A: 18 GHz
1142A: 8.4 GHz



This command sets the frequency span for the synthesizer. The frequency span is centered around the current center frequency. It defaults to the maximum range of the synthesizer. Whenever a new frequency span is received, the synthesizer recomputes the start and stop frequencies based on the current center frequency and the new frequency span. If the computed start or stop frequency falls outside the allowable range, an error message will be generated.

A new frequency span is normally sent along with a center frequency to reduce the chance of generating error messages. For example, to set a frequency span of 6 GHz, send the frequency span along with the desired center frequency. Reducing the frequency span increases the range of allowable center frequencies. At maximum span, only one center frequency is allowable. For example, if the frequency span on the Model 1142A is set to 6 GHz then the center frequency can be any value between 7 GHz and 9.4 GHz. In mathematical terms, the allowable range of the center frequency can be calculated using the following formula:

$$\text{Fcenter} = \frac{\text{Fmax} + \text{Fmin}}{2} \pm \frac{\text{Max Span} - \text{Span}}{2}$$

For the 1142A, this formula can be rewritten as:

$$\text{Fcenter} = 8.2\ \text{GHz} \pm \frac{8.4\ \text{GHz} - \text{Span}}{2}$$

for the 1141A, the formula can be rewritten as:

$$\text{Fcenter} = 11\ \text{GHz} \pm \frac{18\ \text{GHz} - \text{Span}}{2}$$

and for the 1140A, the formula can be rewritten as:

$$\text{Fcenter} = 10.005\ \text{GHz} \pm \frac{19.99\ \text{GHz} - \text{Span}}{2}$$

Examples:

1. To set the frequency span, send center frequency along with the desired span. For example, to set a frequency span of 6 GHz and a center frequency of 7 GHz, send the command

   :SOURce:FREQ:SPAN 6 GHz;:SOURce:FREQ:CENTer 7 GHz

Upon receiving the new center frequency and span, the synthesizer will recompute the corresponding start and stop frequency. For this example, the new start frequency would be 4 GHz and the new stop frequency would be 10 GHz.

2. To query the synthesizer for the current span, send the command

   :SOURce**:FREQ:SPAN?**

   After receiving the query command, a read will return the current mode setting in the <NR1> format. For example, if the 1142A is at the default frequency span and the synthesizer is queried for the current value, it should return

   8400000000

   indicating the current frequency span is 8.4 GHz.

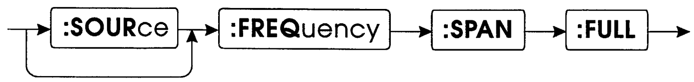
Related Commands:

:SOURce**:FREQ:CENT**er

:SOURce**:FREQ:STAR**t

:SOURce**:FREQ:STOP**

SOURce:FREQuency:SPAN:FULL



This command is used to set the start frequency, stop frequency, center frequency, and frequency span back to the default values shown below. Although this command is used to change synthesizer parameters, it is an event rather than a state, therefore it has no associated query.

For 1140A:	Start: 0.01 GHz	For 1141A:	Start: 2 GHz	For 1142A:	Start: 4 GHz
	Stop: 20 GHz		Stop: 20 GHz		Stop: 12.4 GHz
	Center: 10.005		Center: 11 GHz		Center: 8.2 GHz
	Span: 19.99		Span: 18 GHz		Span: 8.4 GHz

Example:

To set the start frequency, stop frequency, center frequency, and frequency span to the default values, send the command

:SOUR:FREQuency**:SPAN:FULL**

Related Commands:

:SOURce**:FREQ:STAR**t

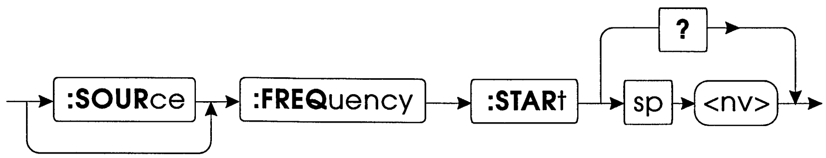
:SOURce**:FREQ:STOP**

:SOURce**:FREQ:CENT**er

:SOURce**:FREQ:SPAN**

SOURce:FREQuency:STARt

Default: 1140A: 0.01 GHz
1141A: 2 GHz
1142A: 4 GHz



This command sets the start frequency for the synthesizer. It defaults to minimum allowable frequency. Whenever a new start frequency is received, the synthesizer recomputes a new center frequency and frequency span based on the current stop frequency.

Normally a start frequency is less than a stop frequency, but it is valid to set a start frequency greater than the stop frequency. If the start frequency is greater than the stop frequency, the computed frequency span will be negative.

Examples:

1. To set the start frequency to 6 GHz, send the command

   :SOURce:FREQ:STARt 6 GHz

   Upon receiving the new start frequency, the synthesizer will change both the center frequency and frequency span based on the new start frequency and current stop frequency.

2. If it is necessary to send a new start frequency that is greater than the current stop frequency, send both the new start and stop frequencies on the same line as shown below.

   :SOURce:FREQ:STARt 6 GHz;:SOURce:FREQ:STOP 8 GHz

3. To query the synthesizer for the current start frequency, send the command

   :SOURce:FREQuency:STARt?

   After receiving the query command, a read will return the current start frequency in the <NR1> format. For example, if the 1142A is at the default start frequency and the synthesizer is queried for the current value, it should return

   4000000000

   indicating current start frequency is 4 GHz.

Related Commands:

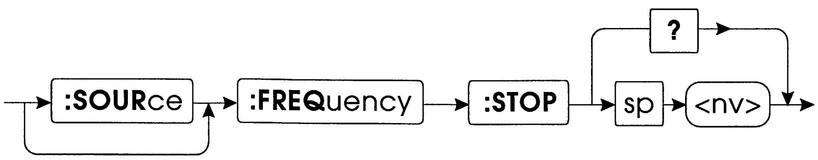
:SOURce:FREQ:STOP

:SOURce:FREQ:SPAN

:SOURce:FREQ:CENTer

SOURce:FREQuency:STOP

Default: 1140A: 20 GHz
1141A: 20 GHz
1142A: 12.4 GHz



This command sets the stop frequency for the synthesizer. It defaults to maximum allowable frequency. Whenever a new stop frequency is received, the synthesizer recomputes a new center frequency and frequency span based on the current start frequency.

Normally a stop frequency is greater than a start frequency, but it is valid to set a stop frequency less than the start frequency. If the stop frequency is less than the start frequency, the computed frequency span will be negative.

Examples:

1. To set the stop frequency to 10 GHz, send the command

   :SOURce:FREQ:STOP 10 GHz

   Upon receiving the new stop frequency, the synthesizer will change both the center frequency and frequency span based on the new stop frequency and current start frequency.

2. If it is necessary to send a new stop frequency that is less than the current start frequency, send both the new start and stop frequencies on the same line as shown below.

   :SOURce:FREQ:STARt 6 GHz;:SOURce:FREQ:STOP 8 GHz

3. To query the synthesizer for the current stop frequency, send the command

   :SOURce:FREQuency:STOP?

   After receiving the query command, a read will return the current stop frequency in the <NR1> format. For example, if the 1142A is at the default stop frequency and the synthesizer is queried for the current value, it should return

   12400000000

   indicating current stop frequency is 12.4 GHz.

Related Commands:

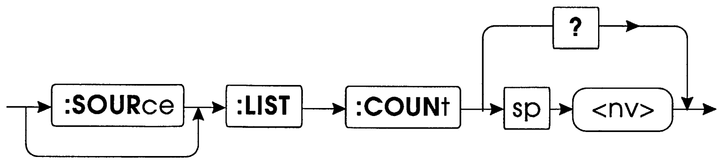
:SOURce:FREQ:STARt

:SOURce:FREQ:SPAN

:SOURce:FREQ:CENTer

SOURce:LIST:COUNt

Default: 1
Range: 1 to 32000



This command sets the number of times the synthesizer will cycle through a list for each start event.

Examples:

1. To command the synthesizer to cycle twice through the triggered list after detecting the trigger, send the command

   :SOURce:LIST:COUNt 2

2. To query the synthesizer for the current list count, send the command

   :SOURce:LIST:COUNt?

   After receiving the query command, a read will return the current list count in the <NR1> format. For example, if the list count was previously set to 10 and the synthesizer is queried for the current list count value, it should return

   10

   indicating the synthesizer is set up to cycle ten times through the triggered list for each trigger.

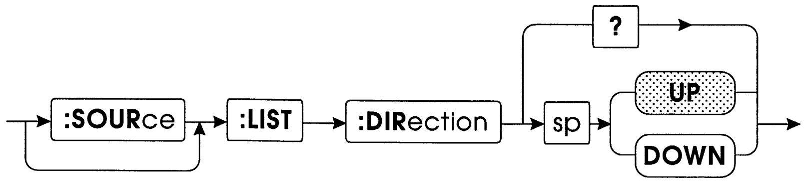
Related Commands:

:SOURce:LIST:DIRection

:SOURce:LIST:STARt

SOURce:LIST:DIRection

Default: UP



This command controls the direction the synthesizer will sequence through the list. Normally, the synthesizer will step from the lowest point vector (PVEC1) toward the highest point vector.

Examples:

1. To command the synthesizer to sequence through the list beginning at the highest point vector, send the command

   :SOURce**:LIST:DIR**ection **DOWN**

2. To query the synthesizer for the current list direction, send the command

   :SOURce**:LIST:DIR**ection**?**

   After receiving the query command, a read will return the current list direction. For example, if the list direction is set to the default direction and the synthesizer is queried for the current list direction, it should return

   UP

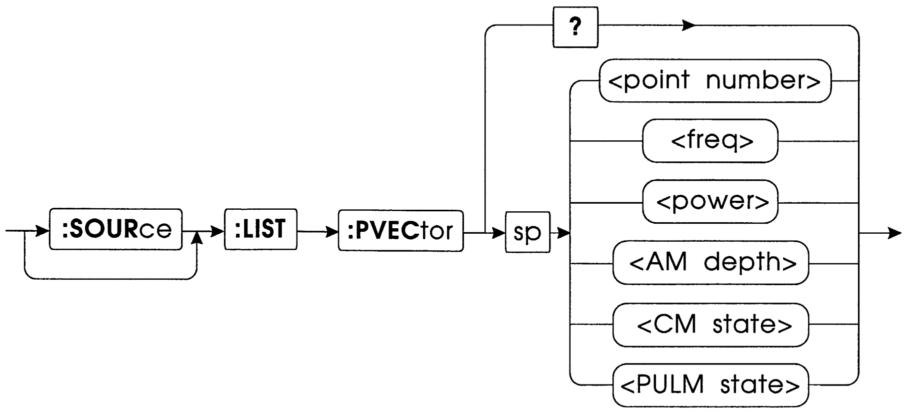
   indicating the synthesizer will sequence through the triggered list starting at the lowest point vector and step toward the highest point vector.

Related Commands:

:SOURce**:LIST:COUN**t

:SOURce**:LIST:STAR**t

SOURce:LIST:PVECtor



This command is used to build a list of instrument setups. Each element, or point, in the list contains up to five instrument states, including frequency, power, AM depth, CM state, and PULM (pulse modulation) state. Once the list is constructed, the synthesizer can be triggered to sequence through the points of the list reconfiguring the instrument at each step.

The programming syntax which identifies a single element in the list is called a "PVECtor", which stands for point vector. The first point vector in a triggered list is called "PVECtor1". The last possible point vector is called "PVECtor 201".

The setup data is programmed into the synthesizer one row at a time. In general, the syntax for sending PVECtor setup data to the synthesizer is:

:LIST:PVECtor <point #>, <frequency>, <power>, <AM depth>, <CM state>, <PULM State>

For example, to program point 1 of the list to 5 GHz at 0 dBm with AM depth at 0, and both complex modulation and pulse modulation turned off, the following command would be sent:

:LIST:PVECtor **1, 5 GHz, 0 dBm, 0, OFF, OFF**

A missing data item in an individual row will be replaced with the current instrument setting. The previous command set the first point vector, PVEC 1. The list could continue with the following elements:

:LIST:PVEC 2
:LIST:PVEC 3, 6 GHz
:LIST:PVEC 4, 5 GHz, 0 dBm, 0, OFF, OFF

Some of the data items for both point vector 2 and 3, shown above, are missing. The missing data would be replaced with the current instrument setting. That is, point vector 2 would be filled in as shown below.

:LIST:PVEC 2, 5 GHz, 0 dBm, 0, OFF, OFF

The same is true for point vector 3, with the exception of the frequency which is called out. That is, point vector 3 would be filled in as shown below.

:LIST:PVEC 3, 6 GHz, 0 dBm, 0, OFF, OFF

When specifying a new list table, identical LIST:PVECtors can be created without specifying all of them. Using a point vector number higher than any previously received causes the synthesizer to fill all points vectors up to the highest number with data contained in the next highest point vector. For example, if the following set of commands are sent to the synthesizer

:MEMory**:DEL**ete **LIST**
:LIST:PVECtor **1, 5 GHz, 0 dBm, 0, OFF, OFF**
:LIST:PVECtor **2, 6 GHz, 0 dBm, 0, OFF, OFF**
:LIST:PVECtor **20, 7 GHz**

Point vectors 3 through 19 will be filled with the data from PVEC 2 and the missing data elements in point vector 20 will be filled in with the corresponding data elements from PVEC 2.

The data fields: frequency, power, etc. can contain any value or keyword that would be valid for the normal command syntax. For example, 8 GHz, UP, DOWN, MIN, MAX are all acceptable frequency data fields.

Controlling List Sequencing

In the triggered list mode, there are a variety of commands to control sequencing through the list. These commands determine what starts the sequencing, what stops the sequencing, the sequencing direction, and how long the synthesizer will remain at each point vector in the list.

:LIST:STARt

When the event specified by the above command occurs, the synthesizer begins sequencing through the list in the direction specified by the command:

:LIST:DIRection

In the trigger list mode, a trigger is required to cause the synthesizer to move to the next point. This trigger can be based on time by setting a dwell time and linking dwell to dwell time using the commands

:LIST:PVECtor**:DWEL**l
:LIST:PVECtor**:DWEL**l**:LINK DWEL**l

To step the synthesizer between list points based on some real event, the dwell can be linked to one of the four trigger sequences.

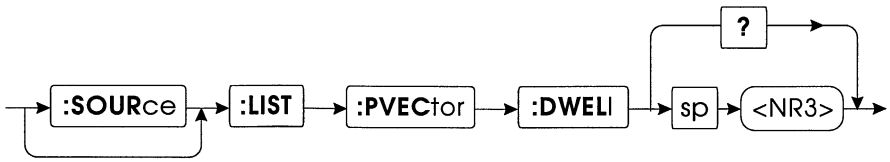
:LIST:COUNt

After a hard reset or MEMory:DELete LIST command, the first LIST:PVECtor command will allocate internal RAM memory for the largest allowable data table. Subsequent LIST:PVEC commands do not allocate any more or less memory. Each LIST:PVEC command causes the instrument to compute and save all the internal parameters that define the state of the synthesizer hardware. The associated point vector commands are not saved, therefore point vectors cannot be read back from the synthesizer. However, a list that is already stored can be modified. Previously specified points can be changed by sending that point vector again with the new data. No point can be deleted once it is successfully sent to the synthesizer.

For complete programming examples of the triggered list feature refer to Section 5 of this manual.

SOURce:LIST:PVECtor:DWELl

Default: 1 second
Range: 5 ms to 32,000



This command controls the amount of time the synthesizer will remain at each point in a trigger list. Using this command, the dwell time can be set. The synthesizer can also be programmed to dwell at each point until an external event is detected using the **:SOURce:LIST:PVECtor:DWELl:LINK** command.

Examples:

1. To set the dwell time to 5 seconds, send the command

   :SOURce:LIST:PVECtor:DWELl 5

2. To query the synthesizer for the dwell time, send the command

   :SOURce:LIST:PVECtor:DWELl?

   After receiving the query command, a read will return the current dwell time in seconds. For example, if the synthesizer was just turned on and it is queried for the current (default) dwell time, it should return

   1.000

   indicating a dwell time of 1 second.

NOTE

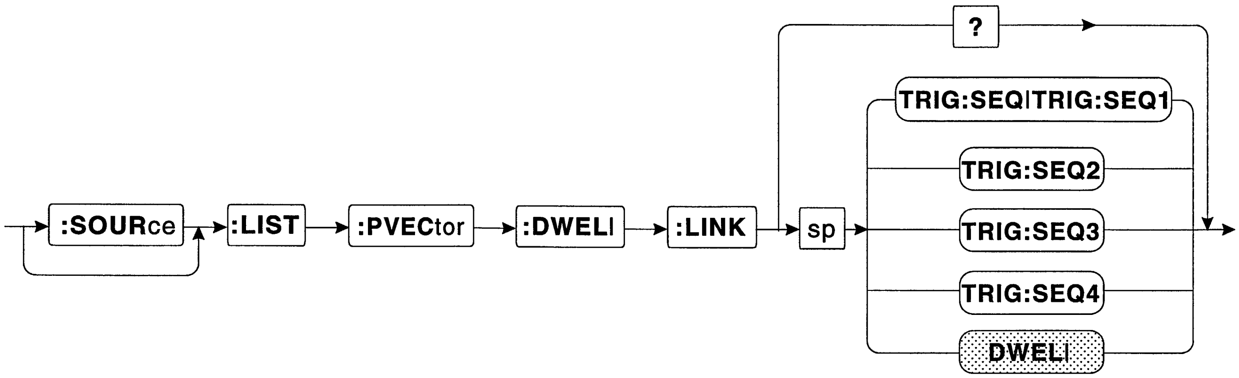
The switching overhead is 35 ms, so the minimum repetition rate is 40 ms (<dwell time> + 35ms).

Related Command:

:SOURce:LIST:PVECtor:DWELl:LINK

SOURce:LIST:PVECtor:DWELl:LINK

Default: DWELl



This command links dwell to the event which will trigger the synthesizer to move to the next point in a triggered list. It can be a specific time by linking dwell to dwell or some real signal or bus command by linking dwell to one of the four trigger sequences. The trigger sequence, in turn, must be linked to the event source using the **:TRIG:SEQ<n>:SOURce** command. See Section 5 for programming examples.

Examples:

1. To command the synthesizer to step between points of a triggered list based on trigger sequence 2 (TRG:SEQ2), send the command

   :SOURce:LIST:PVECtor:DWELl:LINK TRIG:SEQ2

2. To command the synthesizer to step between points of a triggered list every 5 seconds, send the commands

   :SOURce:LIST:PVECtor:DWELl 5
   :SOURce:LIST:PVECtor:DWELl:LINK DWELl

3. To query the synthesizer for the current link to dwell, send the command

   :SOURce:LIST:PVECtor:DWELl:LINK?

   After receiving the query command, a read will return the current link to dwell. For example, if the synthesizer was just turned on and it is queried for the current (default) dwell link, it should return

   DWEL

   indicating dwell is linked to dwell timer.

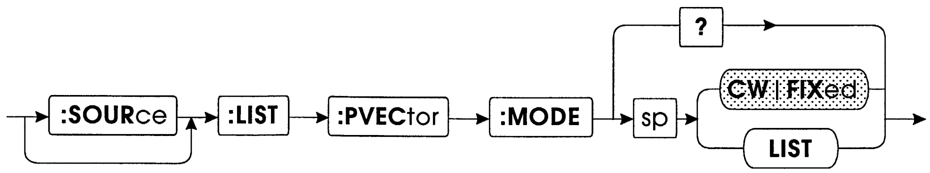
Related Commands:

:SOURce:LIST:PVECtor:DWELl

:SOURce:TRIG:SEQ<n>:SOURce

SOURce:LIST:PVECtor:MODE

Default: CW|FIXed



This command couples control of output power, frequency, amplitude modulation, pulse modulation, and complex modulation to either the FIXed or LIST mode. It is normally in the FIXed mode, but when using the triggered list feature control of each parameter must be coupled to the LIST mode.

Examples:

1. To couple control of output frequency, and power along with the state of amplitude modulation (AM), pulse modulation (PULM), and complex modulation (CM), send the command

   :SOURce:LIST:PVECtor:MODE LIST

2. To query the synthesizer for the current mode, send the command:

   :SOURce:LIST:PVECtor:MODE?

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if the mode was previously set to LIST and the synthesizer is queried for the current mode, it should return

   LIST

   indicating that control of output frequency and power, along with the state of amplitude modulation (AM), pulse modulation (PULM), and complex modulation (CM) is coupled to triggered list.

Related Commands:

:SOURce:FREQ:MODE

:SOURce:POWer:MODE

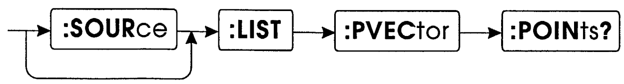
:SOURce:AM:MODE

:SOURce:CM:MODE

:SOURce:PULM:MODE

SOURce:LIST:PVECtor:POINts?

Default: 0
Range: 0 to 201



This is a query only command which returns the current number of trigger list point vectors.

For example, to query the synthesizer for the current number of point vectors, send the command

:SOURce:LIST:PVECtor:POINts?

or the shortened version

:LIST:PVEC:POINts?

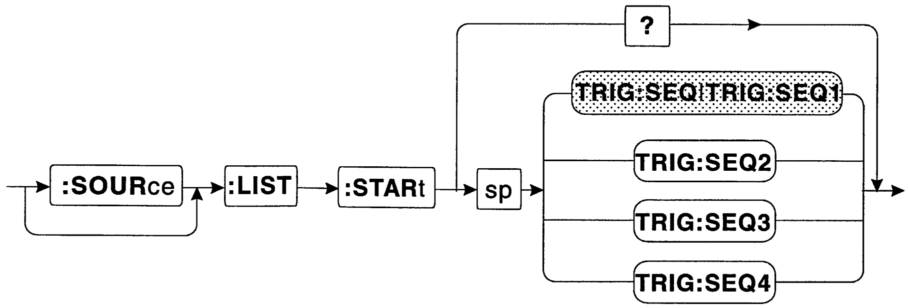
After receiving the query command, a read will return the current number of trigger list point vectors in the <NR1> format. For example, if the synthesizer was just turned on and it was queried for the number of point vectors, it should return

0

indicating that no point vectors have been created.

SOURce:LIST:STARt

Default: TRIG:SEQ1



This command selects the event that will start the synthesizer sequencing through a triggered list. Any one of the four trigger sequences can be programmed to start the synthesizer sequencing through the list. The selected sequence must in turn be programmed to provide the trigger based on some event. See section on triggering for more information on trigger sequences.

Examples:

1. To command the synthesizer to start sequencing through the trigger list upon receiving a trigger from trigger sequence 1, send the command

   :SOURce:LIST:STARt TRIG:SEQ1

   or the shortened version

   :LIST:STAR TRIG:SEQ

   In the shortened version, the number one was not included. If the sequence number is not specified it defaults to 1. That is TRIG:SEQ is an alias for TRIG:SEQ1.

2. To query the synthesizer for the current trigger sequence, send the command

   :SOURce:LIST:STARt?

   After receiving the query command, a read will return the current trigger sequence in the <char> format. For example, if triggered list is using the default trigger sequence, a query should return

   TRIG:SEQ

   indicating that trigger sequence 1 will start the synthesizer sequencing through the triggered list.

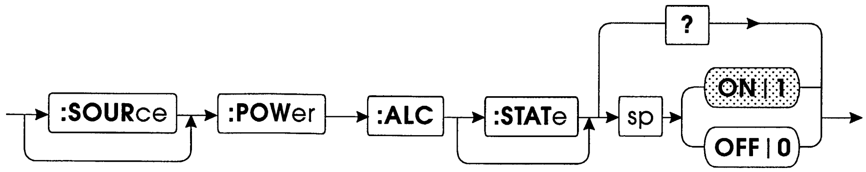
Related Commands:

:SOURce:LIST:COUNt

:SOURce:LIST:DIRection

SOURce:POWer:ALC:STATe

Default: ON



This command turns on and off the automatic leveling control (ALC) on the synthesizer. It should normally be left on to maintain leveled output power. Typically, turning the ALC off reduces the output power, but under certain conditions it can result in output levels of up to +20 dBm at the front panel connector (see Caution).

Since the ALC controls the output power level, turning it off disables normal level control and will cause bit 0 in the Questionable Power event register to be set to a "1".

Examples:

1. To turn off the ALC, send the command

   :SOURce**:POW**er**:ALC:STAT**e **OFF**

   or the shortened version

   :POWer**:ALC 0**

2. To query the synthesizer for the state of the ALC, send the command

   :SOURce**:POW**er**:ALC:STAT**e**?**

   After receiving the query command, a read will return the ALC state in the <bNR1> format. For example, if the ALC was previously turned off and the synthesizer is queried for the current state, it should return

   0

   indicating the ALC has been turned off.

CAUTION

When the ALC in the synthesizer is turned off, the output power at the front panel connector can exceed the maximum specified output. Generally turning off the ALC will cause the output power to drop, but if this command is used in conjunction with the DAC command, the power at the front panel could increase up to +20 dBm.

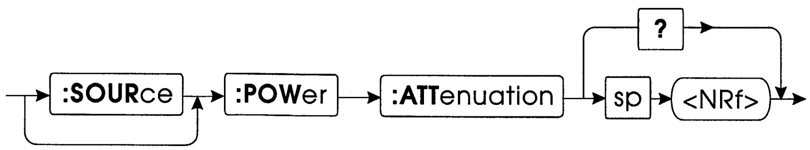
Related Commands:

:SOURce**:POW**er**:ATT**enuation**:AUTO**

:SOURce**:POW**er**:ATT**enuation

SOURce:POWer:ATTenuation

Default: 0 dB
Resolution: 10 dB
Range: 1140A: 0 to 90 dB
1141A: 0 to 90 dB
1142A: 0 to 100 dB



This command sets the attenuation of the RF output step attenuator to a resolution of 10 dB.

Examples:

1. To set RF output to 10 dB, send the command

   :SOURce**:POW**er**:ATT**enuation **10**

2. To set RF output to 20 dB, send the command

   :POWer**:ATT**enuation **20**

3. To query the synthesizer for the status of RF output, send the command

   :POWer**:ATT**enuation**?**

   After receiving the query command, a read will return the current attenuation setting in the <NR2> format. For example, if the attenuation was previously set to 10 dB and the synthesizer is queried for the current attenuation setting, it should return

   10.0

   indicating the attenuation is set to 10 dB.

NOTE

Use of this command automatically decouples the attenuator from the output power level.

Related Commands:

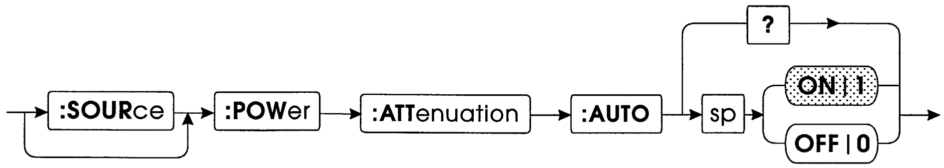
:SOURce**:POW**er**:ALC:STAT**e

:SOURce**:POW**er**:ATT**enuation**:AUTO**

:SOURce**:Pow**er**:LEV**el

SOURce:POWer:ATTenuation:AUTO

Default: ON



This command couples the attenuator to the output power level command. Decoupling the attenuator does not change the current output power.

Examples:

1. To couple attenuator setting to the power setting, send the command

   :SOURce**:POW**er**:ATT**enuation**:AUTO ON**

   or

   :SOURce**:POW**er**:ATT**enuation**:AUTO 1**

2. To leave attenuator to last coupled setting, send the command

   :POWer**:ATT**enuation**:AUTO OFF**

   or

   :POWer**:ATT**enuation**:AUTO 2**

3. To query the synthesizer for the current state of the attenuator, send the command

   :POWer**:ATT**enuation**:AUTO?**

   After receiving the query command, a read will return the current attenuation setting in the <NR2> format. For example, if attenuation coupling was previously set to default and the synthesizer is queried for the current attenuation coupling, it should return

   1

   indicating the ALC and step attenuator are coupled.

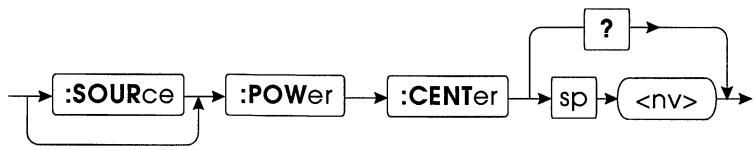
Related Commands:

:SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMPL**itude

:OUTPut**:STAT**e

SOURce:POWer:CENTer

Default: 5 dBm
Range: Minimum output level plus 1/2 power span to maximum output level minus 1/2 power span



This command sets a reference center power for the synthesizer which is used to establish a midpoint for power sweeps. It defaults to 5 dBm, which is the midpoint between the default start and stop power for the synthesizer. Whenever a new center power is received, the synthesizer recomputes the start and stop powers to correspond to the current power span setting. If the computed start or stop power levels are out or range, an error message will be generated.

Examples:

1. To set a center power of -10 dBm, send the command

   :SOURce:POWer:CENTer -10

2. If the power limits are less than 1/2 of the power span from the entered center power, an error message will be generated. To prevent this from occurring, when sending a new center power also send a power span which will prevent the calculated start and stop powers from being out of range.

   For example, to set a center power of 11 dBm with a span of 2 dB, send the center power command combined with the power span command to prevent error messages from being generated.

   :SOURce:POWer:CENTer 11;:SOURce:POWer:SPAN 2

   Upon receiving the new center power and span, the synthesizer will recompute the corresponding start and stop powers. For this example, the new start power would be 10 dBm and the new stop power would 12 dBm.

3. To query the synthesizer for the current center power, send the command

   :SOURce:POWer:CENT?

   After receiving the query command, a read will return the current center power in the <NR2> format. For example, if the synthesizer is at the default center power and it is queried for the current value, it should return

   5.0

   indicating the current center power is 5 dBm.

Related Commands:

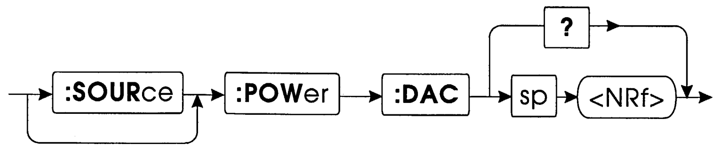
:SOURce:POWer:SPAN

:SOURce:POWer:STARt

:SOURce:POWer:STOP

SOURce:POWer:DAC

Range: 0 to 100%



This command sets the RF power control DAC directly in percent full scale. The command is primarily used for troubleshooting the synthesizer and in some special applications. It does provide control of the output power, but normally the **:SOUR:POW:LEV:IMM:AMP** command is used to control the output power level.

Using this command in combination with the ALC turned off can result in signal levels of up to +20 dBm at the front panel output connector (see Caution below).

Examples:

1. To set the IF power control DAC to 60%, send the command

   :SOURce:POWer:DAC 60%

2. To query the synthesizer for the current setting of the RF power control DAC, send the command

   :POWer:DAC?

   After receiving the query command, a read will return the current setting of the RF power control DAC in the <NR2> format. For example, if the RF power control DAC was previously set to 60% and the synthesizer is queried for the current setting, it should return

   60

   indicating the RF power control DAC is at 60 percent of full scale.

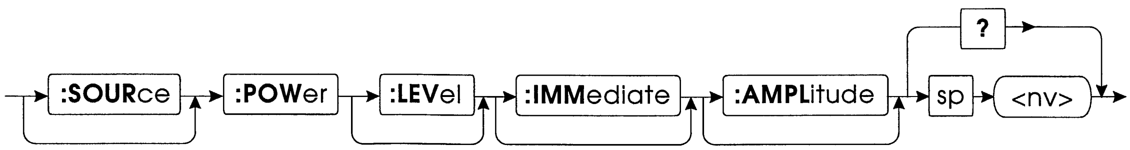
CAUTION

When the ALC in the synthesizer is turned off, the output power at the front panel connector can exceed the maximum specified output. Generally turning off the ALC will cause the output power to drop, but if this command is used in conjunction with the DAC command, the power at the front panel could increase up to +20 dBm.

SOURce:POWer:LEVel:IMMediate:AMPlitude

Default: 0 dBm
Resolution: 0.1 dB

Range:	**1140A: Coupled**		**see 1140A power range specification**
		Uncoupled	**≤18 GHz: +10 to -3 dBm**
			>18 GHz: +5 to -5 dBm
	1141A: Coupled		**see 1141A power range specification**
		Uncoupled	**+10 to -3 dBm**
	1142A: Coupled		**see 1142A power range specification**
		Uncoupled	**+10 to -3 dBm**



This is the main command used to control the output signal level when the synthesizer is operating in the CW mode. If AUTO is on (coupled), it adjusts both the ALC and step attenuator providing level control over the full dynamic range of the synthesizer. If AUTO is off (uncoupled), it only adjusts the ALC and has no control of the step attenuator.

Examples:

1. To select an amplitude of 5 dBm, send the command

   :SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMP**Litude **5**

2. To select an amplitude of 10 dBm, send the command

   :POW 10 DBM

3. To query the synthesizer for the current setting of the output power, send the command

   :SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMPL**itude**?**

   After receiving the query command, a read will return the current setting of the output power in the <NR2> format. For example, if the output power was previously set to the default value and the synthesizer is queried for the current setting, it should return

   0.0

   indicating the output signal amplitude is set to 0 dBm.

NOTE

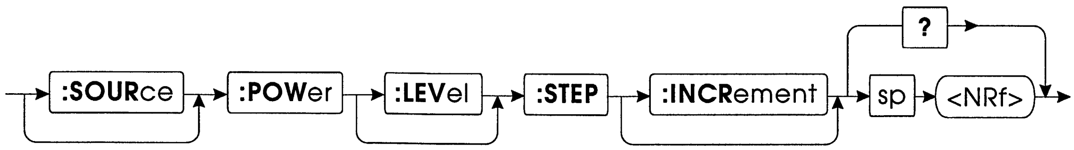
In the default or (***RST**) state, this command is coupled to the RF output step attenuator (i.e., **:POW:ATT:AUTO** defaults to ON). This allows the power to be set over the full range of the synthesizer. It automatically selects the proper output attenuation and internal power drive to achieve the requested output power. Explicitly commanding an attenuation with **:POW:ATT** or **:POW:ATT:AUTO OFF** decouples the attenuator from the :**POW:LEVEL** command. The legal range is then limited to -3 to 10 dBm.

Related Command:

:OUTPut**:STAT**e

SOURce:POWer:LEVel:STEP:INCRement

Default: 0
Resolution: 0.1 dB
Range: 0 to 50 dB



This command sets the incremental step size for stepping the output power from the synthesizer.

Examples:

1. To set the amplitude step size to 10 dB, send the command

   :SOURce**:POW**er**:LEV**el**:STEP:INCR**ement **10**

   or

   :POWer**:STEP 10 DB**

2. To step the output power of the synthesizer by the current step size, send the command

   :SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMPL**itude **UP**

   or

   :POWer **UP**

3. To query the synthesizer for the current power level step size, send the command

   :POWer**:STEP?**

   After receiving the query command, a read will return the current power level step size in the <NR2> format. For example, if the power level step size was previously set to 2 dB and the synthesizer is queried for the current setting, it should return

   2.0

   indicating the step size for power sweep is 2 dB.

Related Commands:

:SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMPL**itude

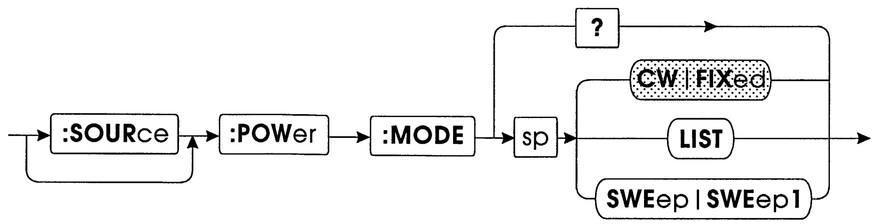
:OUTPut**:STAT**e

NOTE

When attenuator is coupled to output level, setting output power to minimum will generate an out of range error.

SOURce:POWer:MODE

Default: CW|FIXed



This command couples control of the output signal power to either the FIXed or LIST mode. It is normally used to select either the FIXed or SWEep2 modes which couples output power to control the family of source power commands.

The command can be used to couple power control to Triggered List, but is not typically used. The command **:SOURce:LIST:PVECtor:MODE LIST** is normally used for coupling to the LIST mode because it couples POWer as well as AM, CM, FREQ, and PULM.

Examples:

1. To couple control of output power to the LIST mode, send the command

   :SOURce:POWer:MODE LIST

2. To query the synthesizer for the current mode setting, send the command

   :SOURce:POWer:MODE?

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if control of output power is in the default mode and the synthesizer is queried for the current mode, it should return

   FIX

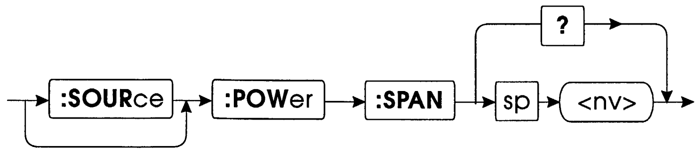
   indicating power level control is coupled to the FIXed mode.

Related Commands:

:SOURce:POWer:LEVel:IMMediate:AMPlitude

:SOURce:LIST:PVECtor:MODE LIST

SOURce:POWer:SPAN
Default: 0 dB
Resolution: 0.1
Range: 1140A: 0.1 dB to 103 dB
1141A: 0.1 dB to 103 dB
1142A: 0.1 dB to 110 dB



This command sets a power span for the synthesizer which is used to establish a range for power sweeps. It defaults to 10 dB, which is the difference between the default start and stop power levels for the synthesizer.

Whenever a power span is received, the synthesizer recomputes the start and stop powers levels based on the center power and new power span. If the computed start or stop power levels are out of range, an error message will be generated.

When performing a power sweep, both the ALC and the internal 10 dB step attenuator work together to provide the proper output level. The step attenuator switches every 10 dB from -2 dBm to the minimum output level (i.e., -2 dBm, -12 dBm, -22 dBm, etc.). For maximum power sweep rates, the center power and sweep span should be selected to avoid step attenuator switching points.

Examples:

1. To set the power power span to 2 dB, send the command

   :SOURce:POWer:SPAN 2

2. When a new power span is received by the synthesizer, new start and stop power points are computed. If these computed power levels are out of range, an error message will be generated. To prevent this from occurring, the a power span can be set along with the require center power. For example, to set a center power of 11 dBm combined with a power span of 2 dB, send the command

   :SOURce:POWer:CENTer 11;:SOURce:POWer:SPAN 2

   Upon receiving the new center power and span, the synthesizer will recompute the corresponding start and stop powers. For this example, the new start power would be 10 dBm and the new stop power would 12 dBm.

3. To query the synthesizer for the current power span, send the command

   :SOURce:POWer:SPAN?

   After receiving the query command, a read will return the current power span in the <NR2> format. For example, if the synthesizer is at the default power span and it is queried for the current value, it should return

   10.0

   indicating the current power span is 10 dB.

Related Commands:

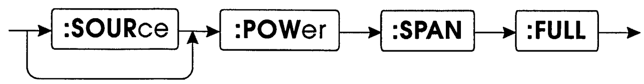
:SOURce:POWer:CENTer

:SOURce:POWer:STAR

:SOURce:POWer:STOP

SOURce:POWer:SPAN:FULL

Full Range: 1140A: 103 dB
1141A: 103 dB
1142A: 110 dB



This command sets a power span to cover the full dynamic range of the synthesizer. This command causes the synthesizer to set the start power to minimum and the stop power to maximum. This type of command is an event rather than a setting and therefore cannot be queried directly, but a power span query could be used.

Examples:

1. To set the power power span to full, send the command

   :SOURce:POWer:SPAN:FULL

2. To query the synthesizer for the current power span, send the command

   :SOURce:POWer:SPAN?

   After receiving the query command, a read will return the current power span in the <NR2> format. For example, if the 1142A synthesizer was commanded to full power span and then queried for the current power span, it should return

   110.0

   indicating a full power span of 113 dB.

Related Commands:

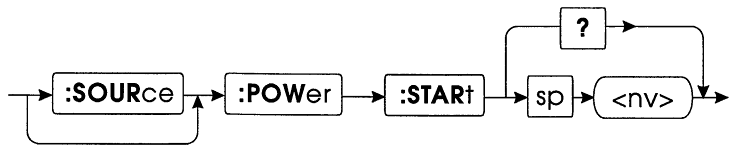
:SOURce:POWer:SPAN

:SOURce:POWer:STARt

:SOURce:POWer:STOP

SOURce:POWer:STARt

Default: 0 dBm
Resolution: 0.1 dB
Range: 1140A: -90 dBm
1141A: -90 dBm
1142A: -100 dBm



This command sets the start power used by the synthesizer when performing power sweeps. It defaults to 0 dBm, but can be set anywhere within the dynamic range of the synthesizer. Whenever a new start power is received, the synthesizer recomputes a new power span and center power based on the current stop power.

Normally a start power level is less than a stop power level, but it is valid to set the start power to a level greater than the stop power level. If the start power is greater than the stop power, the computed power span will be negative.

Examples:

1. To set the start power to -20 dBm, send the command

   :SOURce:POWer:STARt -20 dBm

   or the shortened version

   :POW:STAR -20

   Upon receiving the new center frequency and span, the synthesizer will change both the center frequency and frequency span based on the new start frequency and current stop frequency.

2. To query the synthesizer for the current start power, send the command

   :SOURce:POWer:STARt?

   After receiving the query command, a read will return the current start power in the <NR2> format. For example, if the synthesizer is at the default start power and it is queried for the current value, it should return

   0.0

   indicating the current start power is 0 dBm.

Related Commands:

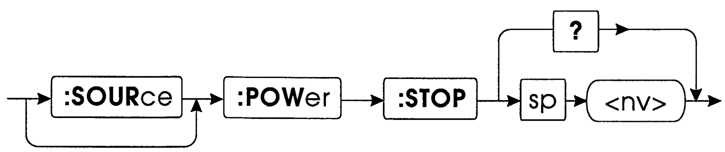
:SOURce:POWer:CENTer

:SOURce:POWer:STOP

:SOURce:POWer:SPAN

SOURce:POWer:STOP

Default: 10 dBm
Resolution: 0.1 dB
Range: 1140A: +10 dBm (10 MHz to <2 GHz)
+13 dBm (2 GHz to <10 GHz)
+10 dBm (10 GHz to <18 GHz)
+5 dBm (18 GHz to 20 GHz)
1141A: +13 dBm (2 GHz to <10 GHz)
+10 dBm (10 GHz to 20 GHz)
1142A: +10 dBm (10 GHz to 20 GHz)



This command sets the stop power used by the synthesizer when performing power sweeps. It defaults to 10 dBm, but can be set anywhere within the dynamic range of the synthesizer. Whenever a new stop power is received, the synthesizer recomputes a new power span and center power based on the current start power.

Normally a stop power level is greater than a start power level, but it is valid to set the stop power to a level less than the start power level. If the stop power is less than the start power, the computed power span will be negative.

Examples:

1. To set the stop power to +5 dBm, send the command

   :SOURce:POWer:STOP 5 dBm

   or the shortened version

   :POW:STOP 5

2. To query the synthesizer for the current stop power, send the command

   :SOURce:POWer:STOP?

   After receiving the query command, a read will return the current stop power in the <NR2> format. For example, if the synthesizer is at the default stop power and it is queried for the current value, it should return

   10.0

   indicating the current stop power is 10 dBm.

Related Commands:

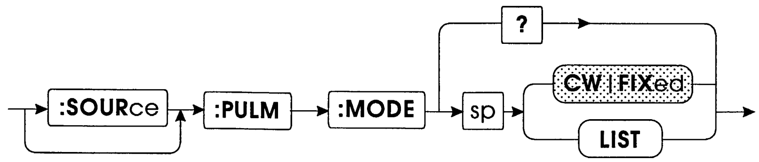
:SOURce:POWer:STARt

:SOURce:POWer:SPAN

:SOURce:POWer:CENTer

SOURce:PULM:MODE

Default: CW|FIXed



This command couples control of the state of pulse modulation to either the FIXed or LIST mode. It is normally in the FIXed mode which means that pulse modulation can be turned on and off using the **:SOUR**ce**:PULM:STAT**e command. In the LIST mode, the state of pulse modulation is controlled by parameters set using the **:SOUR**ce**:LIST** commands.

The command is included to provide compatibility with SCPI defined syntax and is fully functional, but is not typically used. The command :**SOUR**ce**:LIST:PVEC**tor**:MODE LIST** is normally used for coupling to the LIST mode because it couples PULM as well as AM, CM, FREQ, and POWer.

Examples:

1. To couple pulse modulation to the LIST mode, send the command

   :SOURce**:PULM:MODE LIST**

2. To query the synthesizer for the current mode setting, send the command

   :SOURce**:PULM:MODE?**

   After receiving the query command, a read will return the current mode setting in the <char> format. For example, if pulse modulation was previously set to the list mode and the synthesizer is queried for the current mode, it should return

   LIST

   indicating the state of pulse modulation is currently being controlled by the triggered list function.

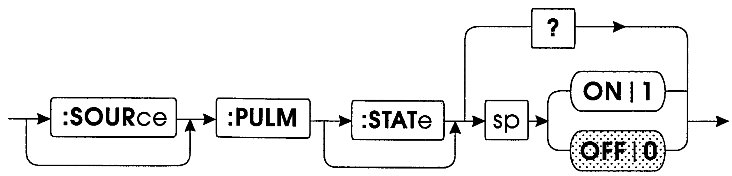
Related Commands:

:SOURce**:PULM:STAT**e

:SOURce**:LIST:PVEC**tor**:MODE LIST**

SOURce:PULM:STATe

Default: OFF



This command turns the internal pulse modulator on or off.

Examples:

1. To turn on the pulse modulation input, send the command

   :SOURce:PULM:STATe ON

   or

   :SOURce:PULM:STATe 1

2. To turn off the pulse modulation input, send the command

   :PULM:STATe OFF

   or

   :PULM:STATe 0

3. To query the synthesizer for the current state of the pulse modulation input, send the command

   :PULM:STATe?

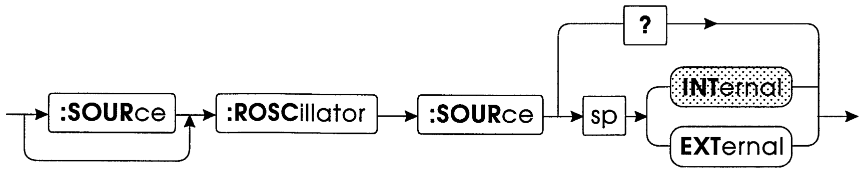
   After receiving the query command, a read will return the state of amplitude modulation in the <bNR1> format. For example, if the pulse modulation was previously enabled and the synthesizer is queried for the current state, it should return

   1

   indicating that pulse modulation is enabled (ON).

SOURce:ROSCillator:SOURce

Default: INTernal



This command selects the source of the reference oscillator for the synthesizer. EXTernal causes the synthesizer to use an external 10 MHz from the front panel connector.

Examples:

1. To select an external reference, send the command

   :OUTPut**:ROSC**illator**:STAT**e **OFF;:SOUR**ce**:ROSC**illator**:SOUR**ce **EXT**

   In this example, two commands were actually sent, separated by a semicolon. The first command turns off the 10 MHz output connector on the front panel which configures it as an input. The second command switches the source of the 10 MHz reference from the internal timebase to the front panel connector. Since the front panel 10 MHz connector defaults to off, only the second command would be normally be necessary.

2. To select the internal 10 MHz reference oscillator, send the command

   :SOURce**:ROSC**illator**:SOUR**ce **INT**

3. To query the synthesizer for the reference oscillator source, send the command

   :ROSCillator**:SOUR**ce**?**

   After receiving the query command, a read will return the source of the reference oscillator in the <char> format. For example, if the reference oscillator was previously set to internal and the synthesizer is queried for the current state, it should return

   INT

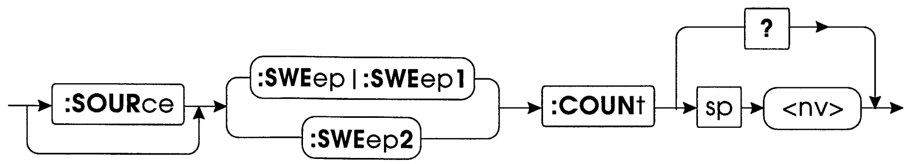
   indicating that source of the reference oscillator is set to internal.

Related Command:

:OUTPut:ROSCillator

SOURce:SWEep<n>:COUNt

Default: 1
Range: 1 to 32000



The sweep count determines the how many times the synthesizer will repeat a sweep function enabled by a single trigger event. The SWEep variable “n” selects between frequency and power sweeps. Using either a “1” or leaving it blank denotes a frequency sweep and using a “2” denotes a power sweep.

Examples:

1. To set sweep count for a frequency sweep to 4, send the command

   :SOURce:SWEep1:COUNt 4

   or

   :SOURce:SWEep:COUNt 4

   Note: As indicated in the example, SWEep and SWEep 1 are equivalent.

2. To set sweep count for a power sweep to 3, send the command

   :SOURce:SWEep2:COUNt 3

3. To query the synthesizer for the current value of SWEep2 count, send the command

   :SOURce:SWEep2:COUNt?

   After receiving the query command, a read will return the current SWEep2 count in the <NR1> format. For example, if the SWEep2 count was previously set to the default value and the synthesizer is queried for the current setting, it should return

   1

   indicating that for power sweeps, the synthesizer will perform a single sweep in response to a trigger.

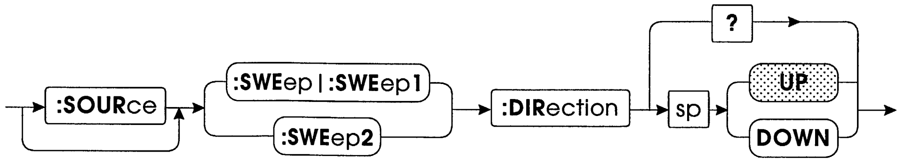
Related Commands:

:SOURce:SWEep<n>:DIRection

:SOURce:SWEep<n>:DWELl

SOURce:SWEep<n>:DIRection

Default: UP



The sweep direction command determines which direction the synthesizer will step for either frequency of power sweeps. The SWEep variable "n" selects between frequency and power sweeps. Using either a "1" or leaving it blank denotes a frequency sweep and using a "2" denotes a power sweep.

Examples:

1. To command the synthesizer to step down when performing a power sweep, send the command

   :SOURce:SWEep2:DIRection DOWN

   or

   :SWEep2:DIR DOWN

2. To set up the synthesizer to step down when performing a frequency sweep, send the command

   :SOURce:SWEep1:DIRection DOWN

   or

   :SWEep:DIR DOWN

   Note: As demonstrated in this example, SWEep and SWEep1 are equivalent.

3. To query the synthesizer for the current direction of SWEep2, send the command

   :SOURce:SWEep2:DIR?

   After receiving the query command, a read will return the current sweep direction in the <char> format. For example, if the sweep direction was previously set to down and the synthesizer is queried for the current setting, it should return

   DOWN

   indicating the sweep direction is set to down.

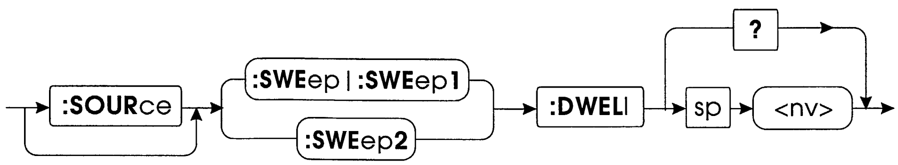
Related Commands:

:SOURce:SWEep<n>:COUNt

:SOURce:SWEep<n>:DWELl

SOURce:SWEep<n>:DWELl

Default: 0.250 seconds
Resolution: 0.1 milliseconds
Accuracy: ±5%
Range: 0.1 to 32000 seconds for :SWEep1
0.05 to 32000 seconds for :SWEep2



In the sweep mode the synthesizer moves between adjacent points in steps. The length of time the synthesizer remains at each point is controlled by setting dwell time variable. Dwell time does not include switching time which is the time it takes for the synthesizer hardware to change power or to settle to the new frequency. Typically, the switching time is 35 ms. Dwell time only controls how long the synthesizer will remain at each point once the signal is settled. The total time required to complete a sweep is determined by the formula

Total Time = (Dwell Time + Switching Time) x Number of Points

The SWEep variable "n" selects between frequency and power sweeps. Using either a "1" or leaving it blank denotes a frequency sweep and using a "2" denotes a power sweep.

Examples:

1. To set the amount of time the synthesizer waits at each frequency point when performing a frequency sweep to 0.3 seconds, send the command

   :SOURce:SWEep1:DWELl 0.3

   or

   :SOURce:SWEep:DWELl 0.3

   Note: As demonstrated in this example, SWEep and SWEep1 are equivalent.

2. To set the amount of time the synthesizer waits at each power point when performing a power sweep to 1 second, send the command

   :SOURce:SWEep2:DWELl 1

3. To query the synthesizer for the current dwell time of SWEep1, send the command

   :SOURce:SWEep1:DWELl?

   After receiving the query command, a read will return the current dwell time of SWEep1 in the <NR3> format. For example, if the SWEep1 dwell time was previously set to 2 seconds and the synthesizer is queried for the current setting, it should return

   2.000

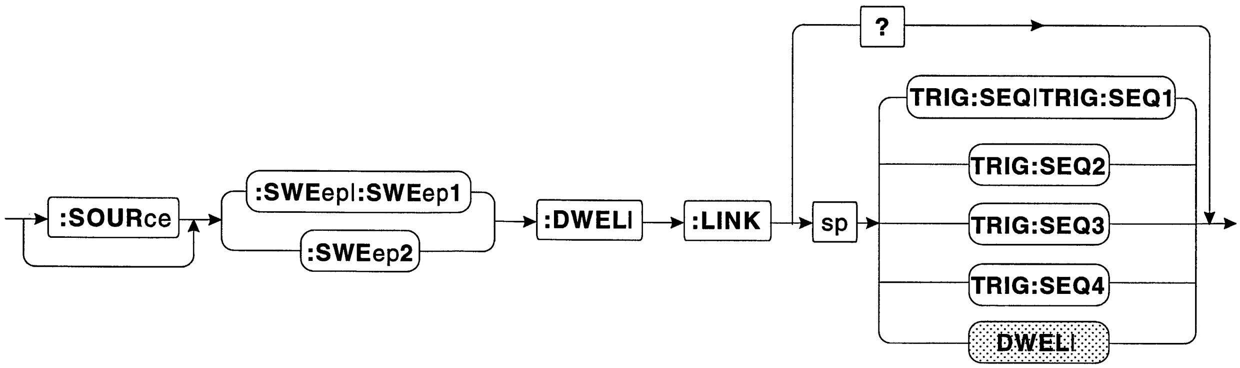
   indicating the synthesizer will dwell (wait) at each step in a frequency sweep for 2 seconds.

Related Commands:

:SOURce:SWEep<n>:COUNt
:SOURce:SWEep<n>:DIRection
:SOURce:SWEep<n>:DWELl:LINK

SOURce:SWEep<n>:DWELl:LINK

Default: DWELl



In the sweep mode, the synthesizer moves between adjacent points in steps. The length of time the synthesizer remains at each point is controlled by dwell. Dwell can be a fixed time, controlled by an internal timer, or dwell can be linked to one of the four trigger sequences. This command links control of stepping to the internal dwell timer or one of the four trigger sequences.

Examples:

1. To set a fixed dwell time based on the internal timer, send the command

   :SOURce:SWEep1:DWELl:LINK DWELl

2. To link dwell to a trigger from trigger sequence 2, send the command

   :SOURce:SWEep1:DWELl:LINK TRIG:SEQ2

3. To query the synthesizer for the current dwell linkage for SWEep1, send the command

   :SOURce:SWEep1:DWELl:LINK?

   After receiving the query command, a read will return the current SWEep1 dwell linkage in the <NR3> format. For example, if the synthesizer is using the default linkage for SWEep1 and is queried for the current linkage, it should return

   DWEL

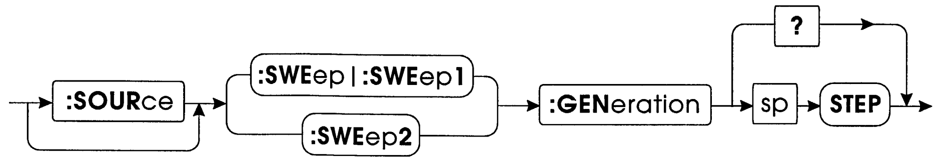
   indicating the internal dwell timer is providing the trigger to move between steps in a frequency sweep.

Related Command:

:SOURce:SWEep<n>:DWELl

SOURce:SWEep<n>:GENeration

Default: STEP



This command is defined in SCPI to be used for selecting between analog and stepped sweep modes. The command has no real purpose for the synthesizer because the synthesizer only supports the step mode, however it was included to maintain compatibility with the SCPI language.

The SWEep variable "n" selects between frequency and power sweeps. Using either a "1" or leaving it blank denotes a frequency sweep and using a "2" denotes a power sweep.

Examples:

1. To set up the synthesizer to use the step mode for frequency sweeps, send the command

   :SOURce:SWEep1:GEN STEP

   or

   :SOURce:SWEep:GEN STEP

   Note: As demonstrated in this example, SWEep and SWEep1 are equivalent.

2. To query the synthesizer for the current sweep mode of SWEep2, send the command

   :SOURce:SWEep2:GEN?

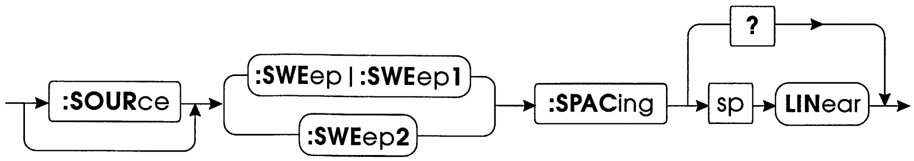
   After receiving the query command, a read will return the current sweep mode in the <char> format. The synthesizer only supports the step mode so if the synthesizer is queried for the current sweep mode, it should always return

   STEP

   indicating the step mode.

SOURce:SWEep<n>:SPACing

Default: LINear



This command is defined in SCPI to be used for selecting between linear and logarithmic modes for sweep functions. The command has no real purpose for the synthesizer because the synthesizer only supports the linear mode for both frequency and power sweeps, however it is included to maintain compatibility with the SCPI language.

The SWEep variable "n" selects between frequency and power sweeps. Using either a "1" or leaving it blank denotes a frequency sweep and using a "2" denotes a power sweep.

Examples:

1. To command the synthesizer to use the linear mode for frequency sweeps, send the command

   :SOURce**:SWE**ep1**:SPAC**ing **LIN**ear

   or

   :SOURce**:SWE**ep**:SPAC**ing **LIN**ear

   Note: As demonstrated in this example, SWEep and SWEep1 are equivalent.

2. To query the synthesizer for the current sweep spacing mode of SWEep2, send the command

   :SOURce**:SWE**ep**2:SPAC?**

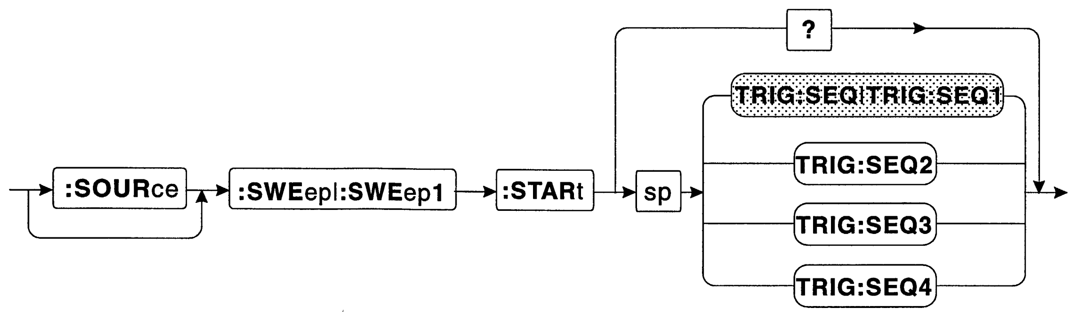
   After receiving the query command, a read will return the current sweep spacing mode in the <char> format. The synthesizer only supports the linear mode so if the synthesizer is queried for the current sweep spacing mode, it should always return

   LIN

   indicating the linear mode.

SOURce:SWEep1:STARt

Default: TRIG:SEQ1



The synthesizer provides four trigger sequences. This command is used to select which trigger sequence will initiate a frequency sweep.

Examples:

1. To command the synthesizer to initiate a frequency sweep based on trigger sequence 2, send the command

   :SOURce:SWEep1:STARt TRIG:SEQ2

   or

   :SOURce:SWEep:STARt TRIG:SEQ2

2. To query the synthesizer for the trigger sequence being used for SWEep1, send the command

   :SOURce:SWEep1:STARt?

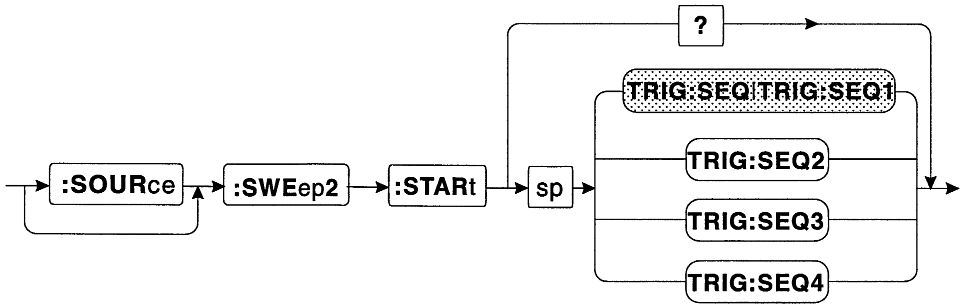
   After receiving the query command, a read will return the current trigger sequence in the <char> format. For example, if the SWEep1 trigger sequence was previously set to the default and the synthesizer is queried for the current setting, it should return

   TRIG:SEQ1

   indicating the SWEep1 trigger sequence is set to trigger from sequence 1.

SOURce:SWEep2:STARt

Default: TRIG:SEQ1



The synthesizer provides four trigger sequences. This command is used to select which trigger sequence will initiate a power sweep.

Examples:

1. To command the synthesizer to initiate a power sweep based on trigger sequence 3, send the command

   :SOURce:SWEep2:STARt TRIG:SEQ3

2. To query the synthesizer for the trigger sequence being used for SWEep2, send the command

   :SOURce:SWEep2:STARt?

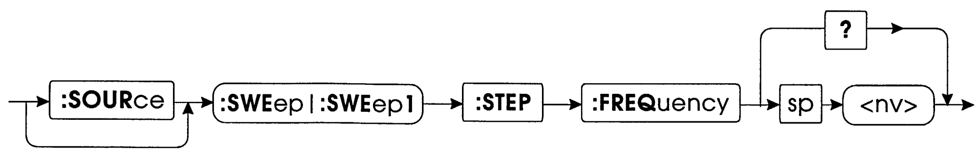
   After receiving the query command, a read will return the current trigger sequence in the <char> format. For example, if the SWEep1 trigger sequence was previously set to the default and the synthesizer is queried for the current setting, it should return

   TRIG:SEQ1

   indicating the SWEep2 trigger sequence is set to trigger from sequence 1.

SOURce:SWEep1:STEP:FREQuency

Default: 1 GHZ
Resolution: 1 Hz



In the frequency sweep mode, the synthesizer moves between adjacent points in steps. This command is used to control the step size of the steps. The step size can be as small as 1 Hz, or as large as the full frequency range. If the step size is too large, an error message will be generated.

Examples:

1. To set the frequency step size to 1 MHz, send the command

   :SOURce:SWEep1:STEP:FREQuency 1 MHz

   If the frequency units (MHz) were omitted, then the units selected by the units subsystem would be used. By default, the units associated with frequency are Hertz, so assuming default units for frequency the following command could also be used to set the step size to 1 MHz

   :SOURce:SWEep1:STEP:FREQuency 1000000

   Since SWEep and SWEep1 are equivalent, "SOURce" is optional, and the lower case letters are not require, the following shortened version of the command is also valid.

   :SWE:STEP:FREQ 1000000

2. To query the synthesizer for the current frequency step size, send the command

   :SOURce:SWEep1:STEP:FREQuency?

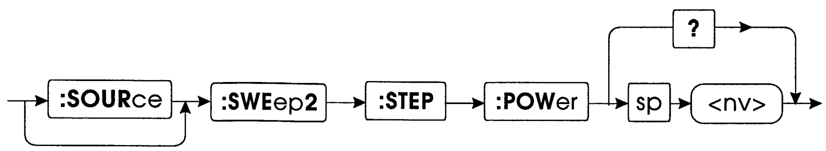
   After receiving the query command, a read will return the current frequency step size in the <NR2> format. For example, if the SWEep1 step size was previously set to the default value and the synthesizer is queried for the current setting, it should return

   1000000000

   indicating the SWEep1 frequency step size is set to 1 GHz.

SOURce:SWEep2:STEP:POWer

Default: 1 dB
Resolution: 0.1 dB



In the power sweep mode, the synthesizer moves between adjacent points in steps. This command is used to control the size of a power step. The step size can be as small as 0.1 dB or as large as the full dynamic range. If the step size is too large, an error message will be generated.

Examples:

1. To set the power step size to 0.5 dB, send the command

   :SOURce:SWEep2:STEP:POWer 0.5 dB

   The power terminator (dB) is not really necessary, since dB and dBm are the only units that can be associated with power.

2. To query the synthesizer for the current power step size, send the command

   :SOURce:SWEep2:STEP:POWer?

   After receiving the query command, a read will return the current power step size in the <NR2> format. For example, if the SWEep2 step size was previously set to the default value and the synthesizer is queried for the current setting, it should return

   1.0

   indicating the SWEep2 power step size is set to 1 dB.

STATUS SUBSYSTEM

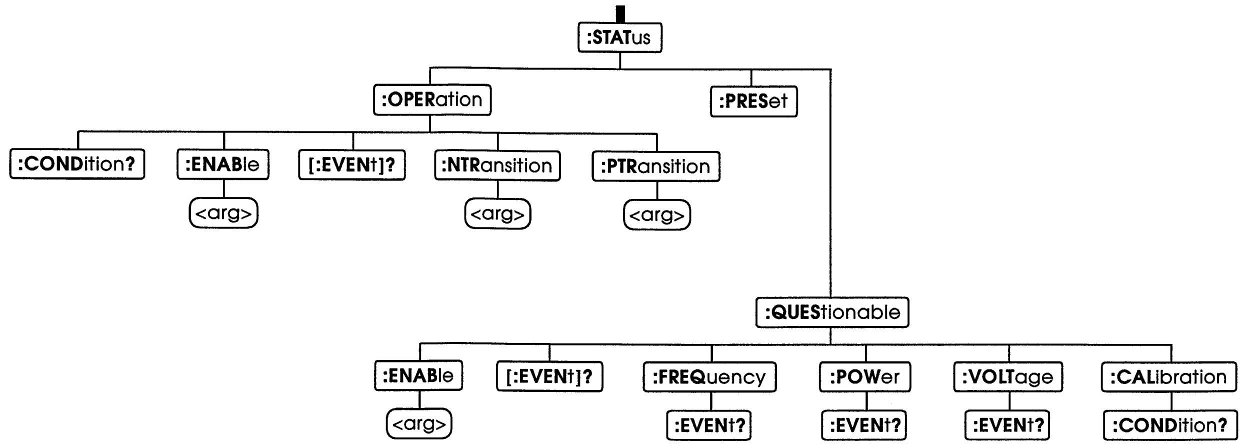


Figure 4-10. Status Subsystem Command Tree.

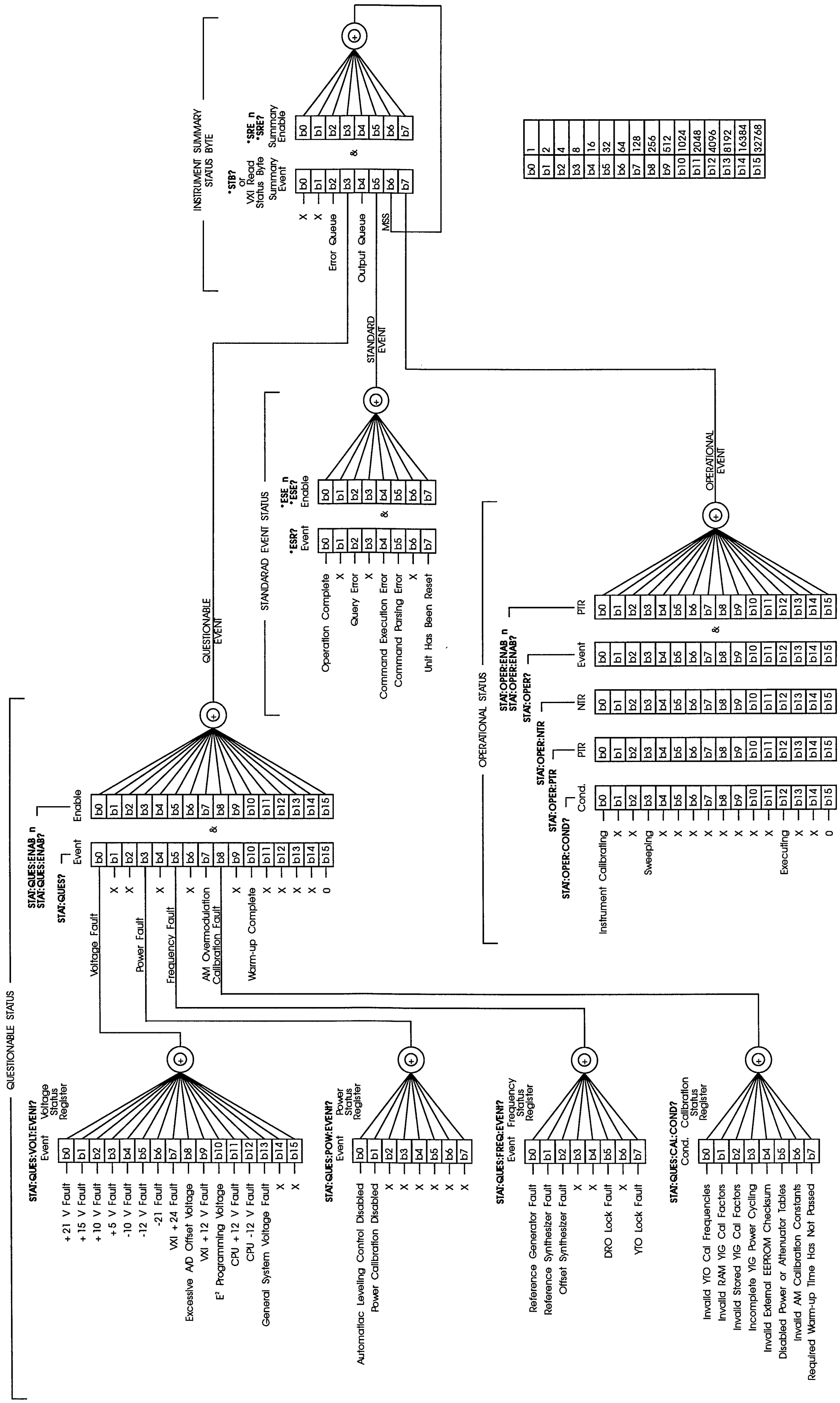
The Status Subsystem controls the SCPI-defined status-reporting structures. These structures are functionally divided into two main groups: the Operational Status registers and the Questionable Status registers. The synthesizer also contains two more groups of status registers: the Standard Events Status registers and the Instrument Status Byte registers. These last two groups are not covered in this section. They are defined by IEEE 488.2 and are listed under Common Commands section of the manual.

There are three types of registers contained in the status reporting structures: condition, event, and enable. The condition registers are continuously updated to reflect the current status of the synthesizer. Reading the contents of a condition register does not change its contents. The event registers are initially all cleared to zero. Whenever an event occurs corresponding to a particular bit, that bit is set to a "1". Reading the contents of an event register causes it to be cleared to zero. The enable registers are used to generate interrupts based on a particular event(s) occurring in the synthesizer. Reading the contents of an enable register does not change its contents.

Transition Filters

Transition filters are special registers associated with the Operational Status registers. Transition filters allow events to be reported based on a condition change within the synthesizer. Negative transition filters (NTR) are used to detect condition changes from True (1) to False (0). Positive transitions filters (PTR) are used to detect condition changes from False to True. Setting both positive and negative filters True allows an event to be reported anytime the condition changes. Reading the contents of a transition filter does not change its contents. The contents of the transition filters are also unchanged by ***CLS** and ***RST** commands.

The detailed structure of the status registers, including those defined by IEEE 488.2, is shown in Figure 4-11.



NOTE: X denotes unused bits which are always cleared to 0.

b0	1
b1	2
b2	4
b3	8
b4	16
b5	32
b6	64
b7	128
b8	256
b9	512
b10	1024
b11	2048
b12	4096
b13	8192
b14	16384
b15	32768

Figure 4-11. Status Structure.

STATus:OPERation:CONDition?

→ :STATus → :OPERation → :CONDition → ?

This query returns the contents of the Operational condition register. Reading a condition register does not change its contents. Bit 15 of this register is always zero. All undefined bits are always cleared to zero.

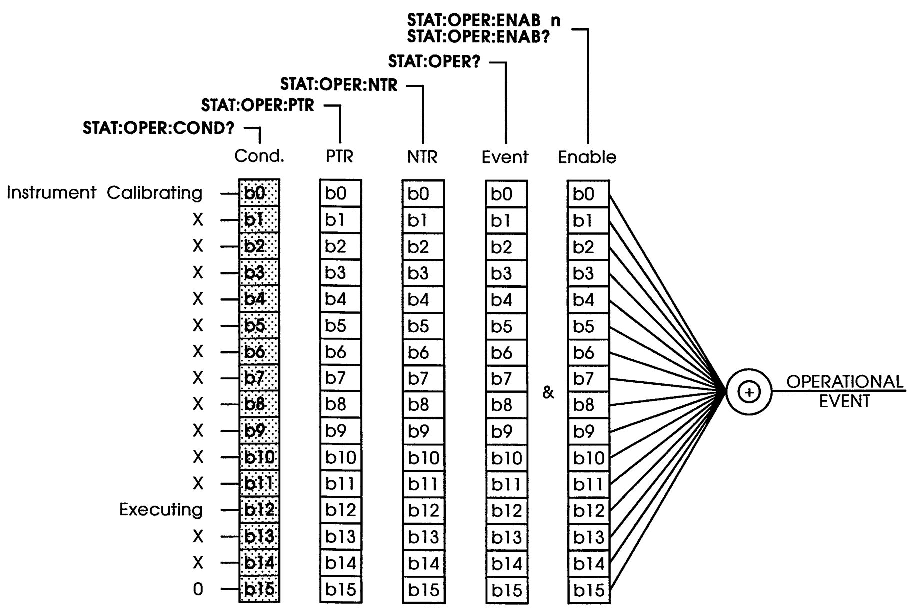
Example:

To query the Operational condition register, send the command

:STATus:OPERation:CONDition?

This query will return a numeric string in the <NR1> format. To determine which bits are set in the register, convert the returned string to its numeric value and then to the corresponding binary bit pattern.

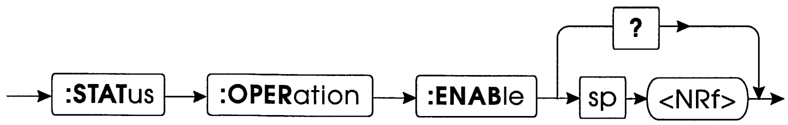
At this time, the only bit that can be set in this register is bit 0, so a query can only return a "0" or a "1". If a "1" is returned, it indicates the synthesizer is performing a calibration.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:OPERation:ENABle
Default: 0



This command sets the enable mask which allows True conditions in the event register to be reported in the Summary event register. The Operational enable register monitors the Operational event register. If a particular bit in the Operational event register becomes True and the corresponding bit in the Operational enable register is True, a "1" will be written in bit 7 of the Summary event register. Querying an enable register will not change its contents.

Examples:

1. To cause bit 7 of the Summary event register to be set when bit 12 of the Operational event status register is set, send the command

   :STATus:OPERation:ENABle 4096

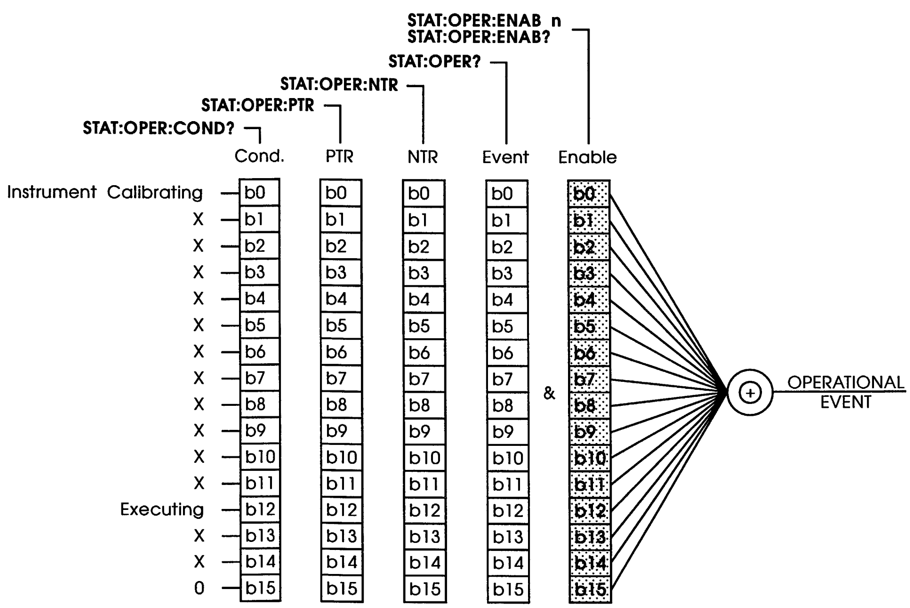
   This sets bit 12 in the Operational enable register which will cause a "1" to be written in bit 7 of the Summary event register if bit 12 of the Operational event register becomes True.

2. To query the synthesizer for the current contents of the Operational enable register, send the command

   :STATus:OPERation:ENABle?

   The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

   For example, if the query returns a "1", it indicates that if bit 0 of the Operational event register becomes True, a "1" will be written to bit 7 of the Summary event register.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:OPERation:EVENt?

This query returns the contents of the Operational event register in the <NR1> format. Its contents are dependent on the Operational condition register and both the positive and negative transition filters. Performing a query on any event register will clear the register.

For example, set all the bits in the positive transition filter to True and all the bits in the negative transition filter to False. At this point, if a query is performed and it returns a "4096", it would indicate a False to True transition of bit 12 in the Operational condition register had occurred since the last query of the Operational event register.

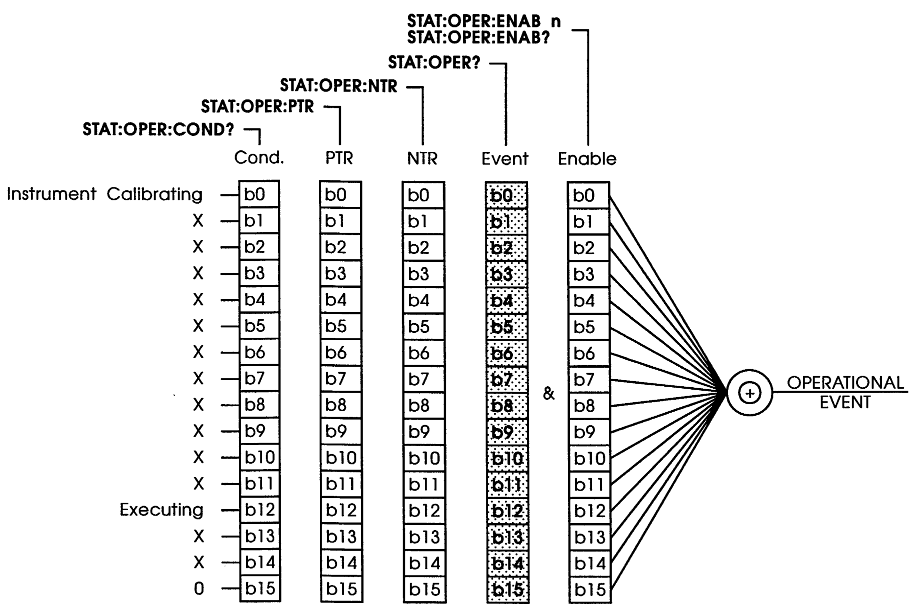
To query the synthesizer for the contents of the Operational event register, send the command

:STATus:OPERation:EVENt?

or

:STATus:OPERation?

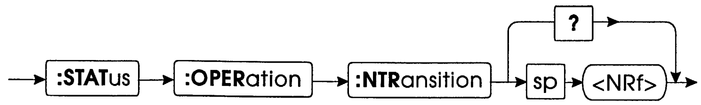
The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:OPERation:NTRansition



This command sets up the contents of the negative transition filter for the Operational status condition register. This negative transition filter monitors changes in the Operational condition register. If a True to False transition occurs in the Operational Condition register and the corresponding bit is set in the negative transition filter, the corresponding bit in the Operational event status register will be set.

Examples:

1. To cause bit 12 of the Operational event status register to be set whenever the synthesizer is finished executing a command, send the command

   :STATus**:OPER**ation**:NTR**ansition **4096**

   Whenever the negative transition filter detects a True to False transition in bit 12 of the Operational Condition register, indicating the synthesizer is finished executing a command, a "1" will be written to bit 12 of the Operational event status register.

2. To query the synthesizer for the current contents of the negative transition filter, send the command

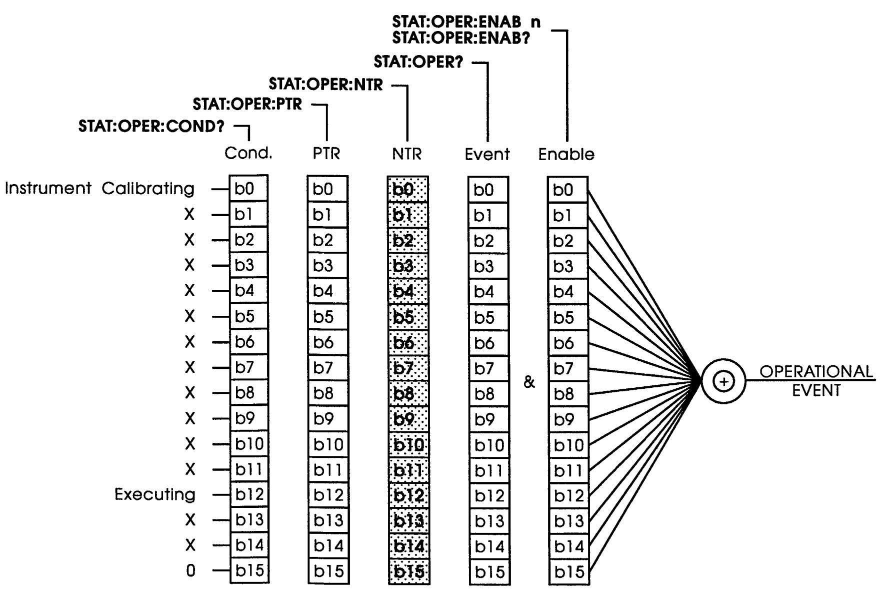
   :STATus**:OPER**ation**:NTR**ansition**?**

   The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

   For example, if the query returns a "1", it indicates the negative transition filter will detect the True to False transition of bit 0 in the Operational condition register and will set bit 0 of the Operational event status register if detected.

NOTE

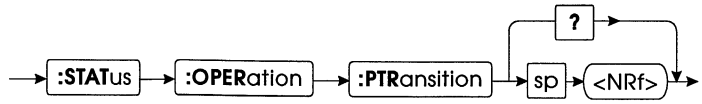
Sending the common commands ***CLS** and ***RST** will not change the contents of a transition filter.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:OPERation:PTRansition



This command sets up the contents of the positive transition filter for the Operational status condition register. This positive transition filter monitors changes in the Operational condition register. If a False to True transition occurs in the Operational Condition register and the corresponding bit is set in the positive transition filter, the corresponding bit in the Operational event status register will be set.

Examples:

1. To cause bit 0 of the Operational event status register to be set whenever the synthesizer is calibrating, sent the command

   :STATus**:OPER**ation**:PTR**ansition **1**

   When the positive transition filter detects a False to True transition in bit 0 of the Operational condition register, indicating the synthesizer is calibrating, a "1" will be written to bit 0 of the Operational event status register.

2. To query the synthesizer for the current contents of the positive transition filter, send the command

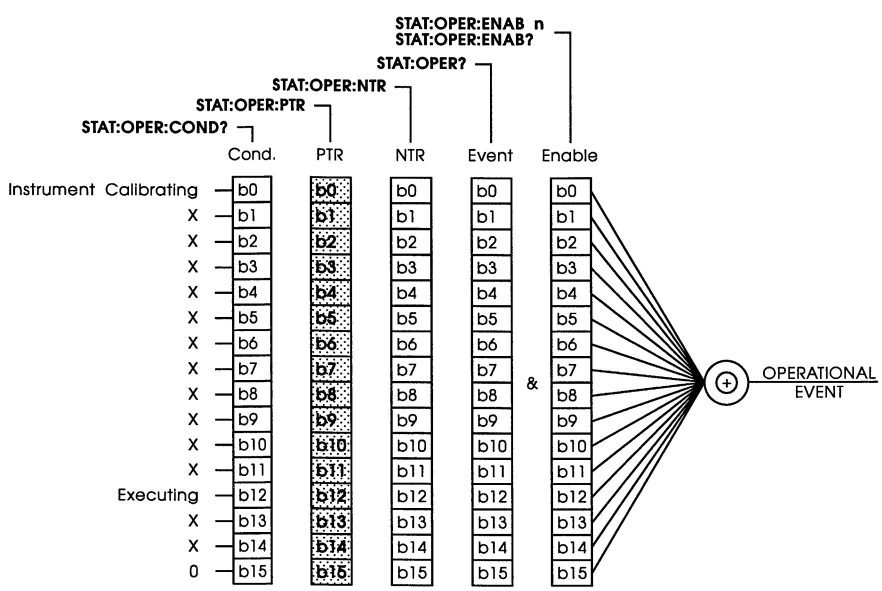
   :STATus**:OPER**ation**:PTR**ansition**?**

   The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

   For example, if the query returns a "4096", it indicates the positive transition filter is set to detect the False to True transition of bit 12 in the Operational condition register. The number 4096 corresponds to a binary bit pattern of 0001000000000000, indicating that bit 12 of the positive filter is set.

NOTE

Sending the common commands ***CLS** and ***RST** will not change the contents of a transition filter.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:PRESet

→ **:STAT**us → **:PRES**et

This command performs four functions: 1) it clears the Questionable enable register to all zero's, 2) it clears the Operational enable register to all zero's, 3) it clears the Operational Negative transition filter to all zero's, and 4) it sets the Operational Positive transition filter to all 1's.

The **:STAT**us**:PRES**et command is equivalent to sending:

:STATus**:QUES**tionable**:ENAB**le **0**
:STATus**:OPER**ation**:ENAB**le **0**
:STATus**:OPER**ation**:NTR**ansition **0**, or
:STATus**:OPER**ation**:PTR**ansition **65535**

This command does not affect either the status byte or the standard event status and does not clear any of the event registers.

Example:

To set up the registers as defined above, send the command

:STATus**:PRES**et

STATus:QUEStionable:CALibration:CONDition?

This query returns the contents of the Questionable Calibration condition register. This condition register indicates a problem with synthesizer calibration.

If a fault is detected, performing a query will not clear the fault since it is a condition register. Cycling synthesizer power may clear the fault. If it does not, the unit needs to be serviced.

Example:

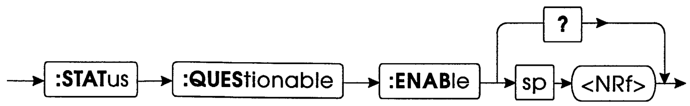
To query the synthesizer for the current contents of the Questionable Calibration condition register, send the command

:STATus:QUEStionable:CALibration:CONDition?

The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

Invalid YTO Cal Frequencies	b0
Invalid RAM YIG Cal Factors	b1
Invalid Stored YIG Cal Factors	b2
Incomplete YIG Power Cycling	b3
Invalid External EEPROM Checksum	b4
Disabled Power or Attenuator Tables	b5
Invalid AM Calibration Constants	b6
Required Warm-up Time Has Not Passed	b7

STATus:QUEStionable:ENABle
Default: 1



This command sets the enable mask which allows True conditions in the Questionable event register to be reported to the Summary event register. The Questionable enable register monitors the Questionable event register. If a particular bit in the Questionable event register becomes True and the corresponding bit in the Questionable enable register is True, a 1" will be written in bit 3 of the Summary event register. Querying an enable register will not change its contents.

Examples:

1. To cause bit 3 of the Summary event register to be set whenever a fault condition is indicated by the Questionable event status register, send the command

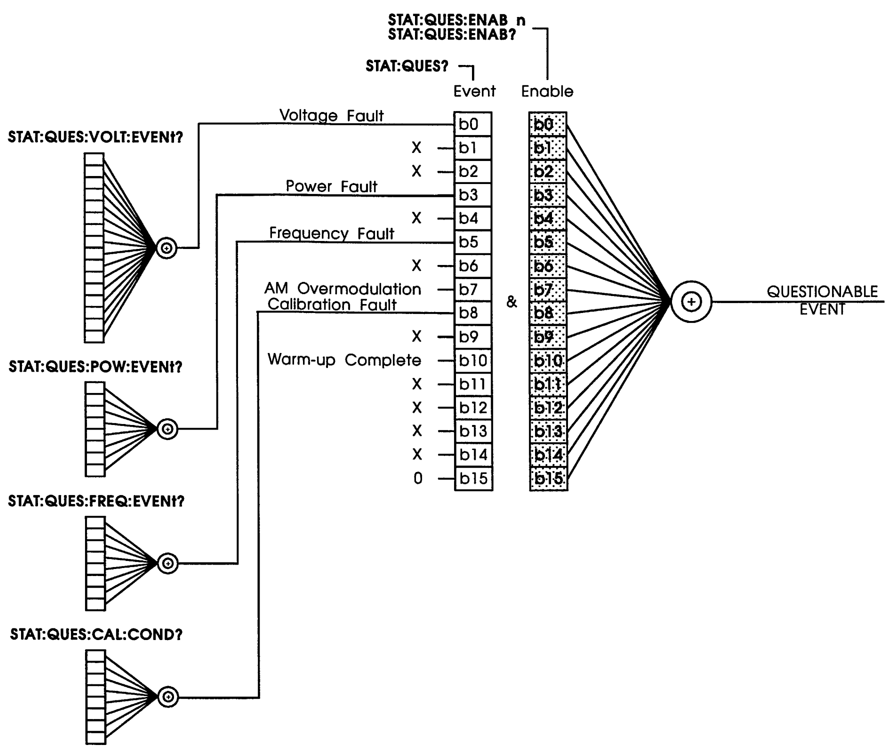
   :STATus**:QUES**tionable**:ENAB**le **425**

   This sets bit 0, 3, 5, 7, and 8 in the Questionable enable register which will cause a "1" to be written in bit 3 of the Summary event register if a fault is detected.

2. To query the synthesizer for the current contents of the Questionable enable register, send the command

   :STATus**:QUES**tionable**:ENAB**le**?**

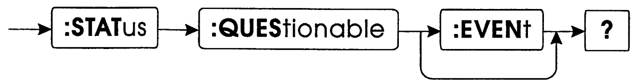
   The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.



NOTE

X denotes unused bits which are always cleared to 0.

STATus:QUEStionable:EVENt?



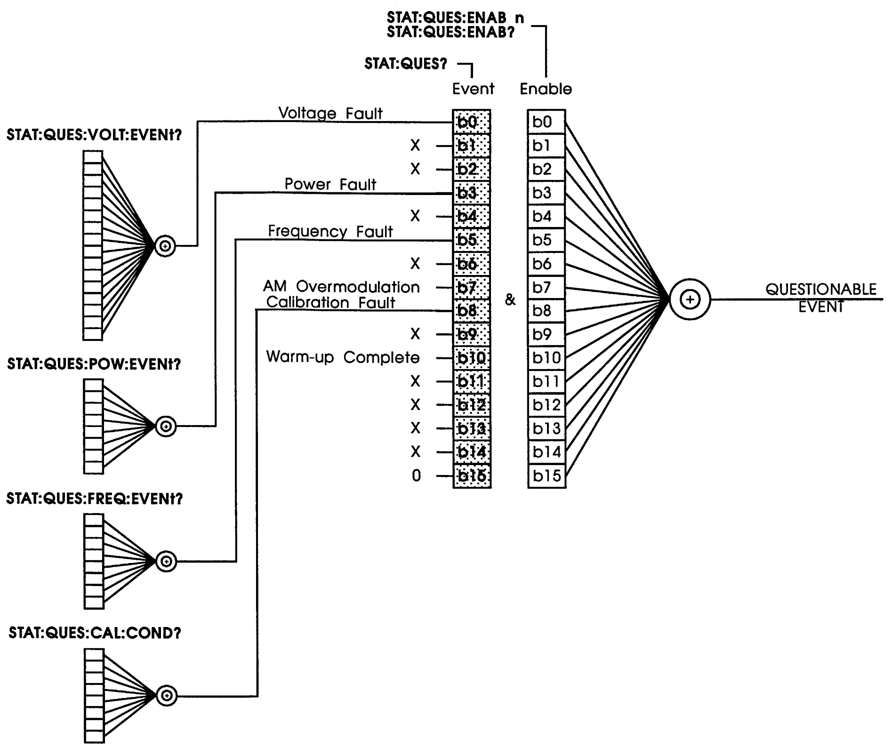
This query returns the contents of the Questionable event register in the <NR1> format. Its contents are determined by the four registers reporting to it and by the two synthesizer conditions of AM Overmodulation and Warm-up Complete as shown below. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

All of the bits used in the Questionable event register indicate a fault when True except bit 10. This bit is proceeded by an implied negative transition filter. It will be True only once each time synthesizer power is cycled after the required warm-up time (15 minutes) has passed. Reading the Questionable event register will clear bit 10 and it will not become True unless the synthesizer power is cycled.

Example:

To query the synthesizer for the current contents of the Questionable event register, send the command

:STATus:QUEStionable:EVENt?



NOTE

X denotes unused bits which are always cleared to 0.

STATus:QUEStionable:FREQuency:EVENt?

→ :STATus → :QUEStionable → :FREQuency → :EVENt → ? →

This query returns the contents of the Questionable Frequency event register. This event register indicates an output frequency fault.

If a fault is detected, perform a query to clear the register. If the fault reappears, it indicates the unit needs to be serviced.

Example:

To query the synthesizer for the current contents of the Questionable Frequency event register, send the command

:STATus:QUEStionable:FREQuency:EVENt?

The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

Reference Generator Fault	b0
Reference Synthesizer Fault	b1
Offset Synthesizer Fault	b2
X	b3
X	b4
X	b5
X	b6
YTO Lock Fault	b7

NOTE

X denotes unused bits which are always cleared to 0.

STATus:QUEStionable:POWer:EVENt?

→ [:STATus] → [:QUEStionable] → [:POWer] → [:EVENt] → [?] →

This query returns the contents of the Questionable Power event register. This event register indicates an output power fault.

Bit 0 of this register will be set when automatic leveling for the synthesizer is disabled. Sending the command **:SOURce:POWer:ALC OFF** will disable automatic power leveling for the synthesizer and will cause bit 0 to be set. Once set, it will remain set until the register is queried.

Bit 1 of this register will be set when power calibration is disabled. Sending the command **:CALibration:POWer:STATe OFF** will disable power calibration for the synthesizer and will cause bit 0 to be set. Once set, it will remain set until the register is queried.

Example:

To query the synthesizer for the current contents of the Questionable Power event register, send the command

:STATus:QUEStionable:POWer:EVENt?

The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

Automatic Leveling Control Disabled	b0
Power Calibration Disabled	b1
X	b2
X	b3
X	b4
X	b5
X	b6
X	b7

NOTE

X denotes unused bits which are always cleared to 0.

STATus:QUEStionable:VOLTage:EVENt?

→ :STATus → :QUEStionable → :VOLTage → :EVENt → ? →

This query returns the contents of the Questionable Voltage event register. This event register indicates a voltage is out of tolerance.

If a fault is detected, perform a query to clear the register. If the fault reappears, it indicates the unit needs to be serviced.

Example:

To query the synthesizer for the current contents of the Questionable Voltage event register, send the command

:STATus:QUEStionable:VOLTage:EVENt?

The synthesizer will return a numeric string, in the <NR1> format, corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

+21 V Fault	b0
+15 V Fault	b1
+10 V Fault	b2
+5 V Fault	b3
-10 V Fault	b4
-12 V Fault	b5
-21 V Fault	b6
VXI +24 V Fault	b7
Excessive A/D Offset Voltage	b8
VXI +12 V Fault	b9
E^2 Programming Voltage	b10
CPU +12 V Fault	b11
CPU -12 V Fault	b12
General System Voltage Fault	b13
X	b14
X	b15

NOTE

X denotes unused bits which are always cleared to 0.

SYSTEM SUBSYSTEM

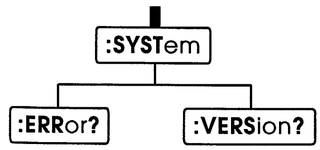


Figure 4-12. System Subsystem Command Tree.

The System Subsystem contains the functions that are not related to synthesizer performance. These are error query and version query.

SYSTem:ERRor?

→ :SYSTem → :ERRor → ? →

This command queries the synthesizer for the next entry from the synthesizer's error queue.

Example:

To query the system error, send the command

:SYSTem**:ERR**or**?**

The synthesizer will respond to this query with an error number, error description, and, optionally, device-dependent information about the error event in the following format:

<Error number>,"<Error description>;<Device-dependent info>"

For example, if a parameter suffix was incorrectly entered, the error message might read

-131,"Invalid suffix"

and sending **:SYST:ERR**or without the ? would result in the error message

-113,"Undefined Header;Query mismatch"

As errors are detected, they are placed in a queue. This queue is first in, first out. If the queue overflows, the last error in the queue is replaced with the error

-350,Queue overflow"

If there has been more than one error, the synthesizer will respond with the first one in its queue. If the queue is empty because the unit was cleared by power on, there has been no error, all errors have been read, or the ***CLS** command has been issued, the synthesizer will respond with

0,"No error"

The following is a list of possible error messages.

-100 "Command error"	-213 "Initiate ignored	-313 "Calibration memory lost"
-104 "Data type error"	-220 "Parameter error"	-314 "Save/recall memory lost"
-109 "Missing parameter"	-221 "Settings conflict"	-330 "Self-test failed"
-110 "Command header error"	-222 "Data out of range"	-350 "Queue overflow"
-113 "Undefined header"	-225 "Out of memory"	-360 "Sweep1 internal error"
-120 "Numeric data error"	-240 "Hardware error"	-400 "Query error"
-131 "Invalid suffix"	-300 "Device Specific Error"	-410 "Query interrupted
-141 "Invalid character data"	-310 "System error"	-420 "Query unterminated"
-200 "Execution error"	-311 "Memory error"	-430 "Query deadlocked"
-211 "Trigger ignored"		

SYSTem:VERSion?

→ [:SYSTem] → [:VERSion] → [?] →

This query is used to determine the SCPI version number for which the synthesizer complies. The response is in the form YYYY.V where the Y's indicate the year-version and the V indicates the approved revision number for that year.

Example:

To determine the system version for the synthesizer, send the command

:SYSTem**:VERS**ion**?**

After receiving the query command, a read will return the current system version in the <char> format. Specifically, the string returned should be

1993.0

which indicates the synthesizer complies to the 1993 version of SCPI.

TRIGGER SUBSYSTEM

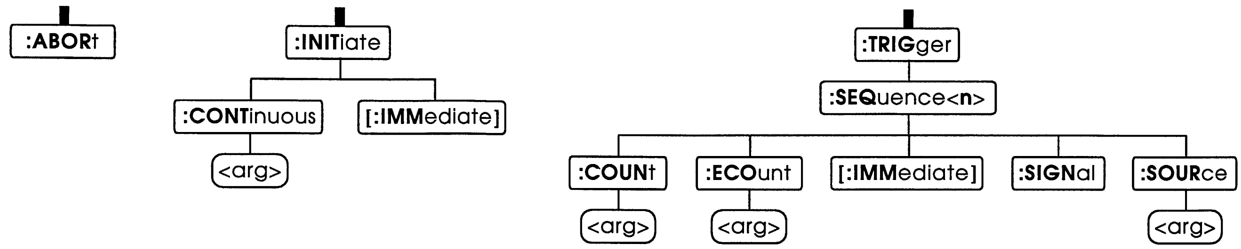


Figure 4-13. Trigger Subsystem Command Tree.

The EIP 114XA implements the Expanded Capability Trigger Model as defined in SCPI. In the standard model, only one triggering sequence is available. The Expanded Capability Trigger Model allows multiple triggering sequences increasing the versatility of the synthesizer. The EIP 114XA synthesizer not only responds to triggers from the VXI backplane, but it can also source these triggers to provide synchronization between a variety of instrument modules.

In general, the trigger subsystem includes commands used to synchronize an action in the synthesizer with an event. Changing the output frequency or power is an example of a synthesizer action. An event is a change such as a trigger from the VXI backplane. The triggering model used in the synthesizer consists of four identical sequences, denoted SEQ1 through SEQ4, with three states per sequence. The trigger model of these sequences is shown below.

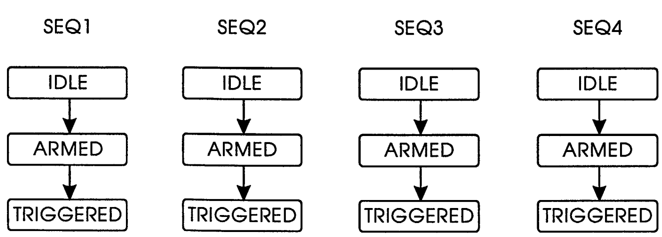


Figure 4-14. Trigger Model.

Each trigger sequence consists of three states, or layers. The idle state is the least active of the three. In the idle state, the synthesizer is simply waiting to be commanded to the armed state. Upon command, the synthesizer moves from the idle state to the armed state where it remains while waiting for the trigger. After receiving the trigger, the synthesizer moves from the armed state to the triggered state and initiates the programmed action, such as changing frequency. After completing the action, the synthesizer normally returns to the idle state, but can be programmed to bypass the idle state. The following paragraphs provide more detail on the commands used to control the movement between layers and to select trigger sources.

At power on, after receiving a ***RST** command, or after receiving an **:ABOR**t command, all four triggering sequences are placed in the idle state. The commands **:INIT**iate[**:IMMediate**] or **:INIT**iate**:CONT**inuous **ON** move the synthesizer out of the idle state and into the armed state. The **:INIT**iate[**:IMMediate**] command simultaneously places all four sequences into the armed state. The four trigger sequences are armed as a group; they cannot be armed individually. The **:INIT**iate**:CONT**inuous **ON** command effectively eliminates the idle state from all four trigger sequins. After receiving this command, the synthesizer returns to the armed state instead of the idle state after triggering an action.

The synthesizer remains in the armed state until the trigger is detected. The **:TRIG:SEQ<n>:SOUR** command is used to select one of the twelve trigger sources listed below to serve as the trigger for a particular sequence (SEQ1...SEQ4). By default, a single transition from the selected trigger source will provide the trigger, but using the **:TRIG:SEQ<n>:ECO** command the synthesizer can be programmed to wait for a specific number of transitions (ECOunt) before proceeding to the triggered state.

BUS
ECLTrg0,ECLTrg1,
TTLTrg0,TTLTrg1,TTLTrg2,TTLTrg3,TTLTrg4,TTLTrg5,TTLTrg6,TTLTrg7

Table 4-15. Trigger Sources.

The BUS source actually represents two sources, either the SCPI command ***TRG** or the VXI word serial trigger command. Each trigger sequence (SEQ1...SEQ4) may have a different trigger source or a single source may be linked to all four trigger sequences. By default, all four trigger sequences are linked to BUS.

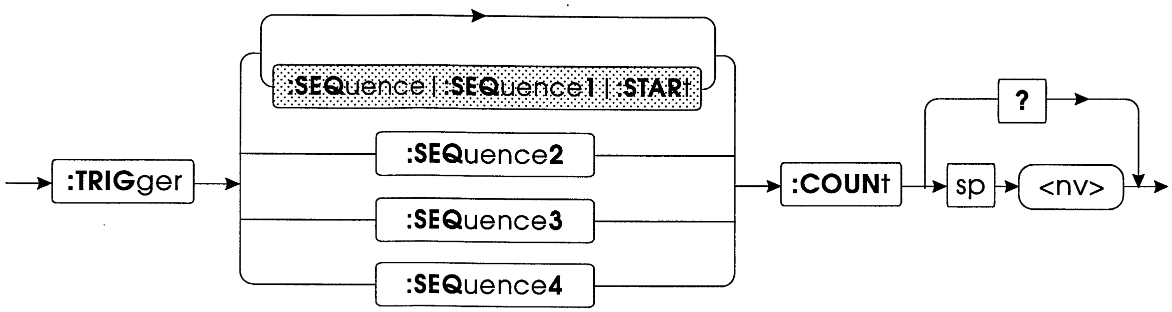
Related Commands:

See Abort Subsystem

See Initiate Subsystem

TRIGger:SEQuence<n>:COUNt

Default: SEQuence1
Range: 1 to 32767



The trigger count variable controls the return to the idle state in the trigger model. In the trigger model, the synthesizer begins in the idle state. Upon receiving a command such as **:INIT**, the synthesizer moves from the idle state to the armed state. It remains in the armed state until the selected event occurs at which time it moves into the triggered state and initiates some action in the synthesizer. After initiating the action, the synthesizer will either move back to the idle state or back to the armed state depending on the trigger sequence count variable. This variable controls how many times to loop between the armed state and the triggered state before returning to the idle state.

Examples:

1. To set up the synthesizer to loop 6 times between the armed state and the triggered state in trigger sequence 1 before returning to the idle state, send the command

   :TRIGger**:SEQ**uence**1:COUN**t **6**

   or

   :TRIGger**:SEQ**uence**:COUN**t **6**

2. To set up the synthesizer to loop 34 times between the armed state and the triggered state in trigger sequence 3 before returning to the idle state, send the command

   :TRIGger**:SEQ**uence**3:COUN**t **34**

3. To query the synthesizer for the current trigger sequence 3 count value, send the command

   :TRIGger**:SEQ**uence**3:COUN**t**?**

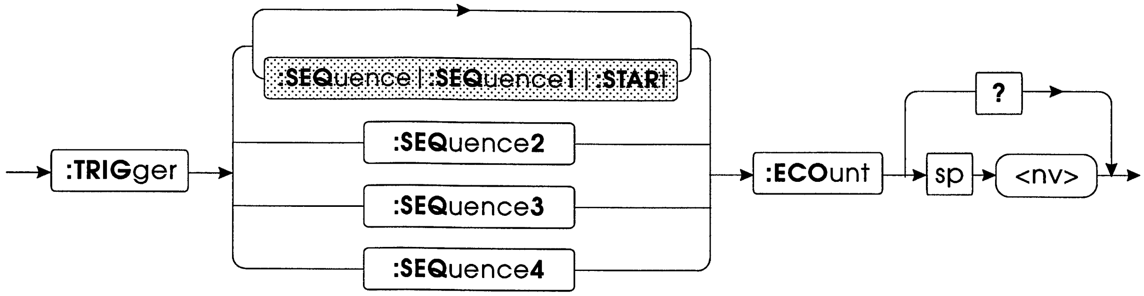
   After receiving the query command, a read will return the current sequence count value in the <NR1> format. For example, if the count value was previously set to the default value and the synthesizer is queried for the current count value, it should return

   1

   indicating that trigger sequence 3 will go from the armed state to the triggered state only once and, after being triggered, it will return to the idle state.

TRIGger:SEQuence<n>:ECOunt

Default: SEQuence1
Range: 1 to 32767



The event count variable controls the number of occurrences of a particular event required to move from the synthesizer from the armed state to the triggered state. When the synthesizer is in the armed state it is waiting for an event to occur before moving to the triggered state and triggering the action. By default, the occurrence of a single event will cause the synthesizer to move from the armed state to the triggered state. However, the synthesizer can be programmed to wait for a particular number of occurrences of the same event before moving from the armed state to the triggered state by programming the ECOunt variable.

Examples:

1. To program the synthesizer to wait in the armed state of trigger sequence 2 for 10 events prior to moving to the triggered state, send the command

   :TRIGger:SEQuence2:ECOunt 10

2. To program the synthesizer to wait in the armed state of trigger sequence 4 for 12 events prior to moving to the triggered state and initiating the desired action, send the command

   :TRIGger:SEQuence4:ECOunt 12

3. To query the synthesizer for the current event count for trigger sequence 4, send the command

   :TRIGger:SEQuence4?

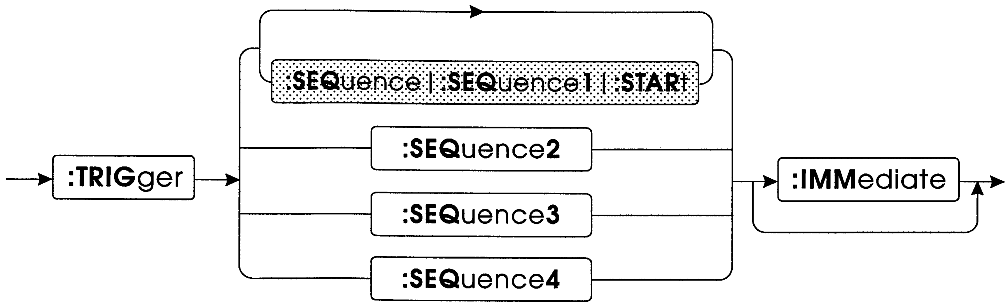
   After receiving the query command, a read will return the current event count value in the <NR1> format. For example, if the count was previously set to ten and the synthesizer is queried for the current count, it should return

   10

   indicating that trigger sequence 4 will wait for ten events before sourcing a trigger.

TRIGger:SEQuence<n>:IMMediate

Default: SEQuence1



This is the master trigger control command. It can be used to immediately trigger a particular sequence (SEQ1...SEQ4). When the synthesizer receives this command, it ignores the event count variable (ECOunt) and immediately triggers the selected sequence providing the sequence is already armed. The variable "n" is used to select which sequence to trigger. If the variable is omitted, it defaults to 1.

Examples:

1. To trigger sequence 1, send the command

   :TRIGger**:STAR**t**:IMM**ediate

   or

   :TRIGger**:SEQ**uence**:IMM**ediate

   or just

   :TRIG

   Note: all variations are equivalent.

2. To selectively trigger sequence 2, send the command

   :TRIGger**:SEQ**uence**2:IMM**ediate

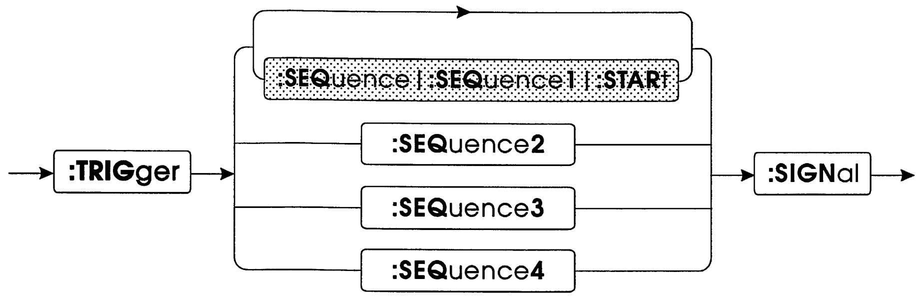
   or

   :TRIG:SEQ2

   Any of the other three sequences can be selectively triggered by replacing the 2 in the preceding command with the particular sequence number.

TRIGger:SEQuence<n>:SIGNal

Default: SEQuence1



This command initiates an immediate trigger acting as if the selected TRIG:SOURce signal/event occurred. The **:TRIG**ger**:SEQ**uence<**n**>**:SOUR**ce command selects between a BUS trigger, one of the two ECLTrg lines, or one of the eight TTLTrg lines to provide the trigger for a particular trigger sequence. This command generates an immediate trigger to one of the four trigger sequences, acting as if the trigger source selected for the particular sequence had occurred.

This command is an event rather than a condition and, as such, it does not support a query.

Examples:

1. If the BUS trigger was previously selected to serve as the trigger source for trigger sequence 2 using the command **:TRIG:SEQ2:SOUR BUS**, then a simulated BUS trigger would be issued by sending the command

   :TRIG:SEQ2:SIGNal

2. If the ECLTrg1 was previously selected to serve as the trigger source for trigger sequence 3 using the command **:TRIG:SEQ3:SOUR ECLT**rg**1**, then a simulated ECLTrg1 trigger would be issued by sending the command

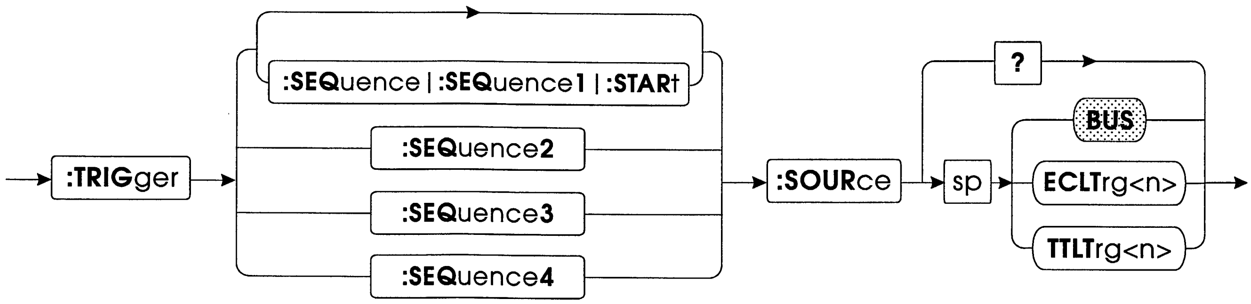
   :TRIG:SEQ3:SIGNal

Related Command:

:TRIGger**:SEQ**uence<**n**>**:SOUR**ce

TRIGger:SEQuence<n>:SOURce

Default: BUS



This command selects the source of a trigger for a particular trigger sequence. By default, the trigger source for all four trigger sequences (SEQ1...SEQ4) is bus. The bus trigger is the trigger generated by either the IEEE 488.2 common command ***TRG** or the VXI word serial trigger command. Besides the bus triggers, there are 10 other triggers available including TTLTrg0 through TTLTrg7 and ECLTrg0 and ECLTrg1.

Examples:

1. To set up the synthesizer to trigger sequence one off the VXIbus ECLTrg1 trigger source, send the command

   :TRIGger**:SEQ**uence**1:SOUR**ce **ECLT**rg**1**

2. To set up the synthesizer to trigger sequence one off the VXIbus TTLTrg4 trigger source, send the command

   :TRIGger**:SEQ**uence**1:SOUR**ce **TTLT**rg**4**

3. To query the synthesizer for the trigger source of sequence 3, send the command

   :TRIGger**:SEQ**uence**3:SOUR**ce**?**

   After receiving the query command, a read will return the sequence 3 trigger source in the <char> format. For example, if the trigger source was previously set to ECLTrg0 and the synthesizer is queried for the current trigger source, it should return

   ECLT0

   indicating the source of triggers for trigger sequence 3 (TRIG:SEQ3) is ECLTrg0 from the VXI backplane.

UNIT SUBSYSTEM

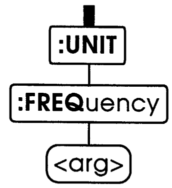
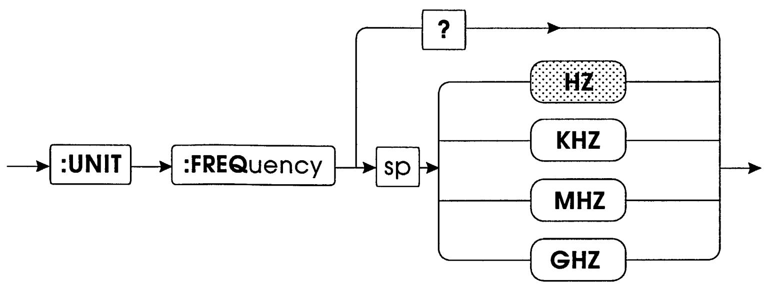


Figure 4-16. Unit Subsystem Command Tree.

The Unit subsystem controls default units associated with data entries.

UNIT:FREQuency

Default: Hz



This command specifies a default metric suffix associated with frequency units. The unit selected will be used with all frequency responses from the synthesizer and will apply to all frequency entries without an explicit suffix.

Examples:

1. To change the default frequency to GHz, send the command

   :UNIT:FREQuency **GHZ**

2. To query the synthesizer for the current frequency unit, send the command

   :UNIT:FREQuency**?**

   After receiving the query command, a read will return the current frequency unit in the <char> format. For example, if the frequency unit was previously set to MHz and the synthesizer is queried for the current frequency unit, it should return

   MHZ

   indicating the current frequency unit is set to MHz.

This Page Intentionally Left Blank

SECTION 5
OPERATION AND PROGRAMMING EXAMPLES

INTRODUCTION

This section provides general information on using the EIP 1140A, 1141A, and 1142A VXIbus Microwave Synthesized Signal Generators in a VXIbus system. Specifically, information is provided on the VXIbus support hardware required to operate the synthesizer in a system along with a variety of programming examples.

VXIbus SYSTEM CONFIGURATIONS

Any VXIbus system consists of four main parts: the VXIbus chassis, the Slot 0 device, the system controller, and the instrumentation modules. The main functions of the VXIbus chassis are to provide the common backplane along with dc power and cooling. The main functions of the Slot 0 device are to provide backplane clocks, configuration signals, and synchronization signals for the system. The system controller is the computer for the system; it provides the user interface to the system along with a platform for developing and running the system software. A typical system providing control of VXIbus system through a Slot 0 Resource Manager is shown below:

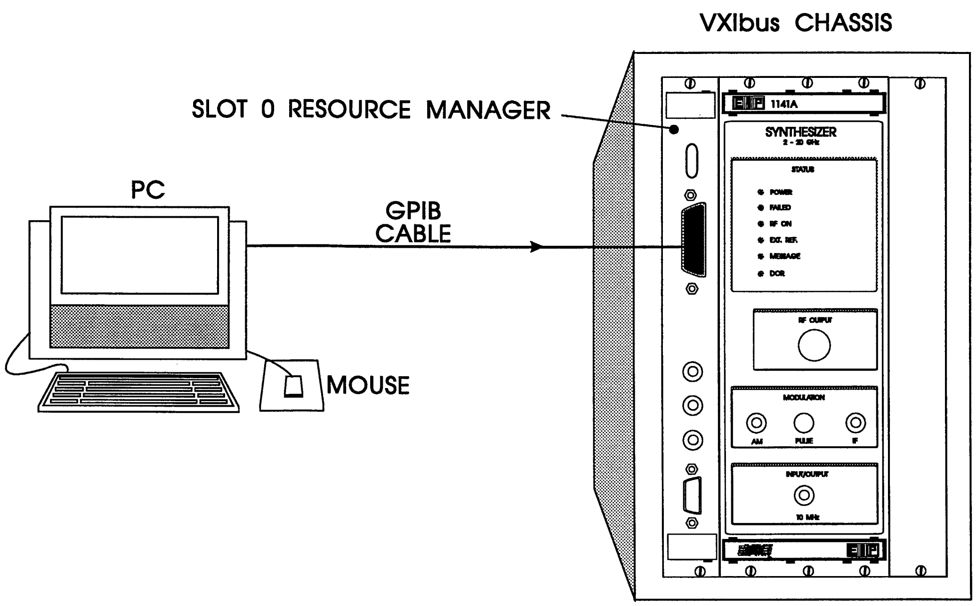


Figure 5-1. VXIbus System Using External Computer.

In this system, a standard DOS based computer (PC) is used to control a VXIbus system. The PC must include a GPIB interface card, such as the National Instruments PCIIA, to provide the communications link between the computer and the VXIbus chassis. The Slot 0 Resource Manager routes the commands from the GPIB interface over the VXI backplane to the addressed instrument, and sends data received off the VXI backplane, from the individual modules, back to the computer via the GPIB interface. One of the advantages of this configuration is that almost any computer with a GPIB interface can be used. Although a VXI system can be controlled using GPIB, this approach precludes many of the advantages offered by VXI by limiting the potential data rates. To realize the full potential of the VXIbus an embedded PC should be used.

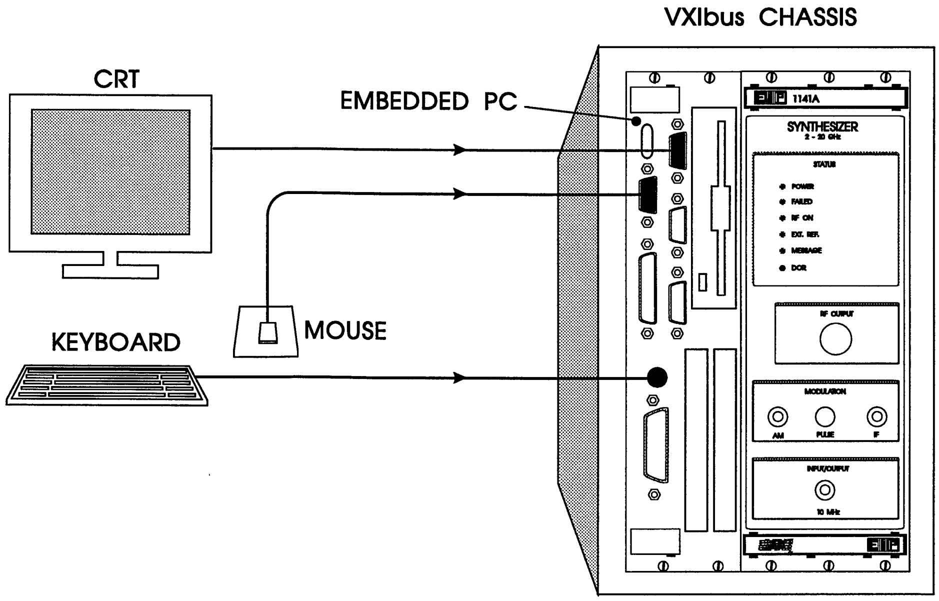


Figure 5-2. VXIbus System using Embedded Computer.

For this configuration, an embedded PC is used to control the system. The embedded PC is plugged into Slot 0 of the VXIbus chassis and serves as both the system computer and resource manager. This configuration allows maximum system throughput by providing direct access to the shared memory space, for high speed data transfers, and direct control of the VXIbus trigger lines. Typically, the embedded PC includes a GPIB port to control standard GPIB instruments. For larger systems, multiple VXI chassis can be combined by configuring each additional chassis with a Slot 0 Resource Manager containing a GPIB port. The main computer could control instrument modules in the additional VXI chassis by communicating through the GPIB interface.

SYSTEM CONFIGURATION

System configuration includes verifying the VXIbus chassis has the power and cooling capacity needed to run the system modules, setting the hardware switches on the modules (such as logical addresses) and configuring the system firmware.

There are a variety of VXIbus chassis available in varying physical sizes, cooling capacity, and dc output power. It is easy to determine if the chassis is physically large enough to house the modules. To determine if the chassis can provide sufficient cooling and dc power to meet the requirements of the various modules, refer to the peak module current specifications for each module and calculate the load on each of the dc supply lines required for the system. If the chassis can provide the dc power, then it will normally provide sufficient cooling because cooling is directly related to the power requirements.

In older chassis, modules may not get proper air flow it the chassis is not full. The open space provides a low resistance path for the cooling air lowering the air flow for the modules. To reduce this problem baffles should be use to limit airflow through unused slots in the chassis. On newer chassis, this is not a problem because a plenum system is used to ensure ample air flow to each slot regardless of chassis loading.

One other consideration when configuring a VXIbus system for RF and microwave applications is the RF characteristics of the chassis. Manufactures have designed chassis specifically for RF and Microwave applications. These chassis offer increased intermodule shielding, low EMI, and low noise (linear) power supplies. These precautions help ensure that the sensitive equipment can operate without interference from the chassis or external environment.

These special RF chassis are not required for all microwave systems. In fact the 1140A, 1141A, and 1142A Microwave Signal Generators will fully meet all performance specifications in any of the commercially available VXIbus chassis. However, the switching power supplies typically used in standard VXIbus chassis can cause switching noise on the dc supply lines which can degrade the spectral purity of the synthesized output signal. These special RF chassis should be considered to obtain optimum system performance.

SAMPLE PROGRAMS

TIMED FREQUENCY SWEEP

This example programs the synthesizer to perform a single timed frequency stepped sweep from 4 to 5 GHz in 10 MHz steps. The synthesizer will remain (dwell) at each step, in the stepped sweep, for one second.

The program listing for the Timed Frequency Sweep is shown below. It is written in lower case to illustrate that commands are not case sensitive; any combination of upper and lower case is acceptable.

```
:init:cont off; :abort; *rst
:unit:freq ghz; *cls; :status:preset
:sweep1:count 1
:sweep:dwell 1 s; dwell:link dwell
:sweep1:start trig:seq2
:freq:start 4; stop 5; :sweep1:step:freq 10 mhz
:freq:mode sweep1
:output on
:init:cont on
:trig:seq2:immediate
```

To provide more insight into programming the synthesizer, each of the steps in the preceding program will be discussed separately.

The program begins with the command:

```
:init:cont off; :abort; *rst
```

This line consists of three commands separated by semicolons. This group of commands initializes the synthesizer. The first command ":init:cont off" causes the synthesizer to remain in the idle state until commanded to the armed state. The second command ":abort" returns all four trigger sequences (TRIG:SEQ1 to TRIG:SEQ4) to the idle state. The third command "*rst" is a IEEE 488.2 Common Command which sets all user programmable variables to default values. Taken together, these three commands set the synthesizer to a known state.

Line two of the program is:

```
:unit:freq ghz; *cls; :status:preset
```

This line also consists of three commands separated by a semicolons. The first command "unit:freq ghz" selects the units to be associated with all frequency entries. By default, frequency entries are expected to be in units of Hertz, but can be set to kHz, MHz, or GHz using the units subsystem. The second command "*cls" is a IEEE 488.2 Common Command which clears all event registers in the status structure and empties the output and error queues. The last command ":status:preset" performs four functions: 1) it clears the Questionable enable register to all zero's, 2) it clears the Operational enable register to all zero's, 3) it clears the Operational Negative transition filter to all zero's, and 4) it sets the Operational Positive transition filter to all 1's.

Line three of the program is:

```
:sweep1:count 1
```

This command sets the frequency sweep count variable to one, which causes the synthesizer to sweep through one time for each trigger.

Line four of the program is:

```
:sweep:dwell 1 s; dwell:link dwell
```

The first command ":sweep:dwell 1 s" sets an internal dwell timer to generate a trigger every second. The second command "dwell:link dwell" links a single frequency or power step to the trigger from the dwell timer. Taken together, the two commands will cause the synthesizer to wait (dwell) at each step in the frequency sweep for 1 second.

Line five of the program is:

```
:sweep1:start trig:seq2
```

This command selects the trigger sequence that will start the frequency sweep. For this example, a trigger from trigger sequence 2 will start the frequency sweep. A command is still needed to cause trigger sequence 2 to generate a trigger.

Line six of the program is:

```
:freq:start 4; stop 5; :sweep1:step:freq 10 mhz
```

This line consists of three commands separated by semicolons. The first two commands ":freq:start 4; stop 5" set the frequency sweep range. The frequency units were set in the second line, to GHz, so the start frequency will be 4 GHz and the stop frequency will be 5 GHz. The third command ":sweep1:step:freq 10 mhz" sets the frequency step size to 10 MHz.

Line seven of the program is

```
:freq:mode sweep1
```

The frequency mode command couples frequency control to one of three modes in the synthesizer. The first mode is CW (or FIXed); it is the mode to use when setting a fixed frequency. The second mode is LIST. The list mode is used to couple frequency control to a triggered list. The third mode is SWEep1; it is the mode required for frequency sweeps.

Line eight of the program is:

```
:output on
```

This line simply turns on the RF output power at the front panel connector.

Line nine of the program is:

:init:cont on

This command causes the synthesizer to move from the idle state to the armed state in the trigger model and wait for a trigger. Once a trigger is received, the synthesizer would normally take a single frequency step and then return to the idle state. With initiate continuously on, the synthesizer will bypass the idle state, fall through to the armed state, and wait for the trigger from the dwell timer. When the trigger occurs, the synthesizer will step to the next frequency. This process will continue until the synthesizer has stepped through the entire range. If the sweep count is one when the synthesizer reaches the stop frequency, it will stop.

Line ten of the program is:

:trig:seq2:immediate

This command immediately triggers sequence 2 causing the synthesizer to start the frequency sweep.

TIMED POWER SWEEP

This example program commands the synthesizer to perform a single timed stepped power sweep from 0 to 8 dBm in 1 dB steps at a frequency of 5 GHz. The synthesizer will remain (dwell) at each power step in the stepped sweep for one second.

The program listing for the Timed Power Sweep is shown below. It is very similar to the previous program with only minor changes required for a power sweep instead of a frequency sweep.

```
:init:cont off; :abort; *rst
:unit:freq ghz; *cls; :status:preset
:sweep2:count 1
:sweep2:dwell 1 s; dwell:link dwell
:sweep2:start trig:seq2
:freq 5; :pow:start 0; stop 8; :sweep2:step:pow 1 db
:pow:mode sweep2
:output on
:init:cont on
:trig:seq2:immediate
```

The first two lines of the program initialize the synthesizer to a known state. Frequency sweeps are identified by "sweep1"; power sweeps are identified by "sweep2". Line three of the program sets a sweep count variable of 1 for power sweeps. This means the power sweep will only be performed one time. The fourth line sets up the dwell for the power sweep. The fifth line links the start of a power sweep to a trigger from trigger sequence 2. The sixth line sets the output frequency to 5 GHz and sets the power sweep range along with power step size. The seventh line couples output power control to sweep2. The remaining three lines are identical to those in the first program and serve the same purpose.

FREQUENCY AND POWER SWEEP

This program combines elements from the first two programs to command the synthesizer to perform a combination frequency and power sweep. Specifically, the program commands the synthesizer to sweep from 5 to 10 GHz in 1 GHz and wait at each frequency step for 15 seconds. At each frequency step, the power is swept from -3 to +10 dBm in 1 dB steps per second. The coupling between sweeps in achieved via VXI TTLTrg1. Each change of the synthesizer's signal is set to cause a pulse on TTLTrg1 of the VXI backplane. The pulse that occurs when the frequency changes triggers the power sweep.

Initialize synthesizer

```
:init:cont off; :abort; *rst
:unit:freq ghz; *cls; :status:preset
```

Set up frequency sweep from 5 to 10 GHz

```
:sweep1:start trig:seq1
:trig:seq1:source bus
:sweep:dwell 15 s; dwell:link dwell
:freq:start 5; stop 10; :sweep1:step:freq 1
:freq:mode sweep1
```

Set up power sweep

```
:sweep2:start trig:seq3
:trig:seq3:source ttltrg1
:sweep2:dwell 1s; dwell:link dwell
:power:start -3; stop 10; :sweep2:step:power 1 db
:power:mode sweep2
```

Start the action

```
:outp on; :outp:ttltrg1:state on
:init:cont on
:trig:seq1:immediate
```

TRIGGERED FREQUENCY SWEEP

This program sets up the synthesizer to sweep from 4 GHz to 5 GHz in 0.1 GHz steps. Dwell time at each frequency step is controlled by a trigger from VXITrg1 off the VXI backplane. This triggered approach to frequency stepping enables another instrument in the system to signal the synthesizer to make a frequency step.

Initialize synthesizer

```
:init:cont off; :abort; *rst
:unit:freq ghz; *cls; :status:preset
```

Set up frequency sweep from 4 to 5 GHz in 0.1 GHz steps

```
:freq:start 4; stop 5; :sweep1:step:freq 0.1 :freq:mode sweep1
:freq:mode sweep1
```

Set up triggering

```
:sweep1:start trig:seq2      ! Start frequency sweep upon trigger from sequence 2
:dwell:link trig:seq2        ! Link dwell time to trigger from sequence 2
:trig:seq2:source ttltrg1    ! Set source of Sequence 2 triggers to TTLTrg1 from VXI backplane
```

Start the action

```
:output on                   ! Turn on RF output power
:initiate:immediate          ! Moves trigger sequence into armed state
:trig:seq2:immediate         ! Trigger sequence 2
```

TRIGGERED LIST

This program uses the the triggered list feature to step through a a quasi random sequence of frequency and power levels and remain at each step for 2 seconds.

When using Triggered List, it is important to set up the synthesizer in the desired configuration prior to sending the list data because the synthesizer stores setup information along with the other variables at each point vector in the list. For example, it the ":output on" command was changed to ":output off" each point vector would be stored with the output RF turned off.

Initialize Synthesizer

```
:init:cont off; :abort; *rst
:unit:freq ghz; *cls; :status:preset
:memory:delete list          ! Delete old list data
:output:rosc on              ! Connect 10 MHz reference to front panel
:output on                   ! Turn on RF output power
```

Set up Triggered List Point Vectors

```
LIST:PVEC <num>,<freq>,<power>,<AM depth>,<CM state>,<PULM state>
:List:pvec  1, 4.0,  1.0 dbm,0,off,off
:List:pvec  2,12.4,  2.0 dbm,0,off,off
:List:pvec  3,  5.5, 3.0 dbm,0,off,off
:List:pvec  4,11.3,  4.0 dbm,0,off,off
:List:pvec  5,  6.4, 5.0 dbm,0,off,off
:List:pvec  6,  8.3, 0.0 dbm,0,off,off
:List:pvec  7,  7.0, 1.0 dbm,0,off,off
:List:pvec  8,  8.0, 2.0 dbm,0,off,off
:List:pvec  9,  9.0, 3.0 dbm,0,off,off
:List:pvec 10,10.0, 4.0 dbm,0,off,off
:List:pvec 11,11.0, 5.0 dbm,0,off,off
:List:pvec 12,12.0, 6.0 dbm,0,off,off
```

Setup triggering

```
:list:pvec:mode list          ! Select list mode
:list:pvec:dwell 2            ! Setup dwell time at each point
:list:pvec:dwell:link dwell   ! Link dwell time to dwell timer
:list:start trig:seq2         ! Start list upon trigger from trigger sequence 2
```

Start the action

```
:initiate:cont on
:trig:seq2:immediate
```

This Page Intentionally Left Blank

APPENDIX A
VXIbus OVERVIEW

This VXIbus overview is from the VXIbus Specification, revision 1.3.

INTRODUCTION

The goal of the VXIbus is to define a technically sound modular instrument standard based on the VMEbus that is open to all manufacturers and is compatible with present industry standards.

VXIbus is an acronym for VMEbus extensions for Instrumentation. The VXIbus specification details the technical requirements of VXIbus compatible components, such as mainframes, backplanes, power supplies, and modules. Before studying the VXIbus architecture, one should become familiar with the VMEbus and its specifications.

VMEbus BACKGROUND

The VMEbus is an open system architecture primarily focused at computer systems, though there presently is a limited offering of instrumentation. VMEbus modules are approximately six inches deep and come in two heights, about four and nine inches. The VXIbus specification refers to these as the A and B sizes, respectively. The precise dimensions are specified by the Eurocard standard, which describes a family of printed circuit boards and their associated DIN connector locations. VMEbus modules are designed for 0.8 inch slot to slot spacing. The A size board has a single 96 pin connector known as P1, while the B size may include a P1 and P2 connector. Each of these DIN connectors consists of three rows of 32 pins apiece on 0.1 inch centers. Typically, these boards are positioned vertically in a frame with the P1 connector closest to the top. Neither the VMEbus nor the VXIbus mandates a physical orientation, since orientation is only an implementation issue not needed for compatibility. Many VMEbus systems are designed to accept boards horizontally.

The VMEbus specification allows a maximum of 21 modules. However, if installed vertically in a mainframe intended for mounting in a standard 19 inch rack, 20 is the practical maximum. VMEbus makes no particular provision for an extension chassis or frame to frame communication. Multiple frame systems can be created by electrically buffering the VMEbus (at the loss of some bandwidth between cages) or by using standard data communication links that disguise the underlying VMEbus architecture. There are no EMC (electromagnetic compatibility) requirements dictated by VMEbus, either conducted or radiated, nor are there power dissipation limits or chassis cooling requirements. VMEbus has left these issues to the system integrator, while VXIbus addresses these issues more rigorously.

Although electrically and logically similar to the 68000 microprocessor architecture, the VMEbus interface has been specified broadly enough that it is not dependent on any particular processor, and many processors are already supported on VMEbus, including the 80386. Many of the simpler VMEbus boards do not have processors at all.

A minimum VMEbus system requires only the P1 connector. All handshaking, arbitration, and interrupt support exists on P1, with P2 used to expand the system to 32 bits of address and data (A32 and D32). P1 will support 16 bit and 24 bit addressing (A16 and A24), as well as 8 and 16 bit data paths (D08 and D16). The extra lines needed for A32 and D32 are contained on the center row of P2, while the outer rows are user defined. These undefined pins are typically used for interface connections, such as allowing a module to drive a chassis mounted connector, access an internal disk drive, or provide for module to module communication. VSB (VMEbus Subsystem Bus) is a standard "subsystem bus" that has defined P2 as an additional communication path for up to six modules. Multiple VSBs may exist within any one VMEbus system. This is important to note, because VXIbus defines a subsystem of up to 13 modules and, like VSB, multiple VXIbus subsystems may exist within any one VXIbus system.

THE VXIbus EXTENSIONS

VXIbus retains P1 and the center row of P2 exactly as defined by VMEbus. This includes the 5 volt and ±12 volt power pins on P1, and the additional 5 volt pins on P2. VXIbus includes the A and B card sizes, and these modules remain totally VMEbus compatible. However, VXIbus has made substantial additions to the VMEbus specification oriented towards instrumentation that can best be described as an electromechanical superset and a logical subset.

VXIbus MODULES

VXIbus has added two Eurocard module sizes of about 13 inch depth referred to as the C and D sizes. These modules are 9 and 14 inches high respectively, and are placed on 1.2 inch centers. The C Eurocard is the same height as the VMEbus B size board, and may sport both the P1 and P2 connectors. The D size module is a triple high Eurocard that may include a P3 connector in addition to P1 and P2. The 1.2 inch module width allows feasible implementation of high density instrumentation modules while allowing enough space for shielding both sides of a module and inserting an optional chassis shield. It also has the added benefit of allowing a high degree of compatibility with the shorter and narrower A and B sizes by allowing them to be mounted on full length board carriers or adapters. These carriers/adapters may also shield the sides of standard VMEbus cards, giving them a high degree of electromagnetic compatibility with VXIbus systems.

VXIbus SUBSYSTEMS

A VXIbus system may have up to 256 devices, including one or more VXIbus subsystems. A VXIbus subsystem consists of a central timing module referred to as Slot 0 with up to 12 additional instrument modules. P2 and P3 are completely defined in a VXIbus subsystem. These 13 modules conveniently fill a standard 19 inch cabinet when mounted vertically on 1.2 inch centers. Many VXIbus systems will consist only of a single frame with these 13 modules. A common configuration will load the Slot 0 module with system resources such as the VXIbus mandated timing generation, the VMEbus required system controller functions, and a data communication port such as IEEE 488 or RS-232. Slot 0 may also include optional instrumentation. The other positions are general purpose slots for the user to mix and match modules. A single VXIbus subsystem may have less than 12 additional slots, but may not have more. Any combination of VXIbus subsystems may exist within a VXIbus system. For instance, one VXIbus system may consist of a frame with one Slot 0 and 12 VXIbus modules extended to another frame that has a Slot 0 adjacent to three instrument slots, another Slot 0 with five instrument slots, and four standard VMEbus slots of undefined P2.

P2 Connector Definition

As mentioned previously, a VXIbus subsystem defines all P2 and P3 pins. The VXIbus P2 adds a 10 MHz ECL clock, ECL and analog supply voltages, ECL and TTL trigger lines, an analog summing bus, a module identification line, and a daisy chain structure known as the local bus. The trigger lines serve primarily as resources for signaling between instruments in a VXIbus subsystem, while the local bus lines are preferred for use within a multiple module instrument set (adjacent slots). The daisy chain local bus use is left to the module manufacturer to define, and several classes of electrical signals are permitted. Allowed signals are TTL, ECL, low voltage analog, and analog up to 42 volts. A keying mechanism near the faceplate indicating that module's local bus class prevents incompatible classes from accidentally being placed adjacently and potentially causing a destructive condition. Typical uses of the local bus include creating an internal analog bus or a chain of serial digital signal processors. There are a total of 24 local bus pins on P2, 12 lines in and 12 lines out for each slot; thus creating a 12 line bus that may or may not be passed on to adjacent slots.

P3 Connector Definition

The VXIbus P3 connector adds many of the same resource types as described for P2, but is aimed at higher performance instrumentation. Included on P3 is a 100 MHz clock and sync signal, additional power pins of the same supply voltages, more ECL trigger lines, and 24 additional lines (48 pins) of daisy chain local bus. Also defined on P3 is a "star" trigger system where precision ECL trigger signals are routed through Slot 0 acting as a cross point switch. This allows very precisely matched trigger timing between modules regardless of module position.

VXIbus SYSTEM ARCHITECTURE

The VXIbus device protocols define how modules are granted non-conflicting portions of the VMEbus address space. A device is typically a single module, but this is not required. Several devices may exist on a single module, and a single device may consist of multiple modules. 256 devices may exist in any one VXIbus system, and are referred to by logical device address ranging from 0 to 255. A VXIbus system configuration space is defined in the upper 16K of the 64K A16 address space. Each device is granted a total of 64 bytes in this space, which is sufficient for many of the simpler devices. Devices requiring additional address space have their address requirements readable in a defined register in the A16 address space. A "resource manager" reads this value shortly after power-on, and then assigns the requested memory space by writing the module's new VMEbus address into the device's offset register. This method positions a device's additional memory space in the A24 (16 Mbyte) or A32 (4 Gbyte) address space. If present day VMEbus cards are used in a system, the resource manager must position the VXIbus devices around the space taken by the standard VMEbus cards.

Higher level communication protocols are defined to allow sharing of interface modules and other devices by multiple manufacturers.

This Page Intentionally Left Blank

APPENDIX B
COMMAND SUMMARY

COMMON COMMANDS, IEEE 488.2

		ARGUMENTS
*CLS	Clear Status Command	
*ESE	Standard Event Status Enable Command	<NRf>
*ESE?	Standard Event Status Enable Query	
*ESR?	Standard Event Status Register Query	
*IDN?	Identification Query	
*OPC	Operation Complete Command	
*OPC?	Operation Complete Query	
*RCL	Recall Command	<NR1>
*RST	Reset Command	
*SAV	Save Command	<NR1>
*SRE	Service Request Enable Command	<NRf>
*SRE?	Service Request Enable Query	
*STB?	Read Status Byte Query	
*TRG	Trigger Command	
*TST?	Self-test Query	
*WAI	Wait-to-Continue Command	

INSTRUMENT DEPENDENT COMMANDS

KEYWORDS	ARGUMENTS
ABORt	
CALibration	
[:ALL]?	
:ATTenuator[:STATe]	ON|OFF|1|0
:POWer[:STATe]	ON|OFF|1|0
INITiate	
:CONTinuous	ON|OFF|1|0
[:IMMediate]	
MEMory	
:DELete	LIST
:FREE[:ALL]?	
:NSTates?	
:QUEue:ERRor:MALLocate	<NR1>
OUTPut	
BLANking	ON|OFF|1|0
:ECLTrg<n>	
:IMMediate	
[:STATe]	ON|OFF|1|0
:FILTer:CENTer	<NR1>
:ROSCillator[:STATe]	ON|OFF|1|0
[:STATe]	ON|OFF|1|0
:TTLTrg<n> (n = 0|1|2|3|4|5|6|7)	
:IMMediate	
[:STATe]	ON|OFF|1|0

<table>
<tr><th colspan="4">KEYWORDS</th><th>ARGUMENTS</th></tr>
<tr><td>[SOURce]</td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td>:AM</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>[:DEPTh]</td><td></td><td><integer>|DEFault|MIN|MAX|UP|DOWN</td></tr>
<tr><td></td><td></td><td></td><td>:STEP[:INCRement]</td><td>< integer></td></tr>
<tr><td></td><td></td><td>:MODE</td><td></td><td>CW|FIXed|LIST</td></tr>
<tr><td></td><td></td><td>:STATe</td><td></td><td>ON|OFF|1|0</td></tr>
<tr><td></td><td>:CM</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:FREQuency</td><td></td><td><NRf></td></tr>
<tr><td></td><td></td><td>:MODE</td><td></td><td>CW|FIXed|LIST</td></tr>
<tr><td></td><td></td><td>:STATe</td><td></td><td>ON|OFF|1|0</td></tr>
<tr><td></td><td>:FREQuency</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:CENTer</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>[:CW|FIXed]</td><td></td><td><nv>|DEFault|MIN|MAX|UP|DOWN</td></tr>
<tr><td></td><td></td><td></td><td>:STEP[:INCRement]</td><td><nv></td></tr>
<tr><td></td><td></td><td>:FTUNe</td><td></td><td><nv>|DEFault</td></tr>
<tr><td></td><td></td><td>:MODE</td><td></td><td>CW|FIXed|LIST|SWEep|SWEep1</td></tr>
<tr><td></td><td></td><td>:SPAN</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td></td><td>:FULL</td><td></td></tr>
<tr><td></td><td></td><td>:STARt</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>:STOP</td><td></td><td><nv></td></tr>
<tr><td></td><td>:LIST</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:COUNt</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>:DIRection</td><td></td><td>UP|DOWN</td></tr>
<tr><td></td><td></td><td>:PVECtor</td><td></td><td><char></td></tr>
<tr><td></td><td></td><td></td><td>:DWELl</td><td><NR3></td></tr>
<tr><td></td><td></td><td></td><td>   :LINK</td><td><char></td></tr>
<tr><td></td><td></td><td></td><td>:MODE</td><td>CW|FIXed|LIST</td></tr>
<tr><td></td><td></td><td></td><td>:POINts?</td><td></td></tr>
<tr><td></td><td></td><td>:STARt</td><td></td><td><char></td></tr>
<tr><td></td><td>:POWer</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:ALC[:STATe]</td><td></td><td>ON|OFF|1|0</td></tr>
<tr><td></td><td></td><td>:ATTenuation</td><td></td><td><NRf></td></tr>
<tr><td></td><td></td><td></td><td>:AUTO</td><td>ON|OFF|1|0</td></tr>
<tr><td></td><td></td><td>:CENTer</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>:DAC</td><td></td><td><NRf></td></tr>
<tr><td></td><td></td><td>[:LEVel]</td><td></td><td></td></tr>
<tr><td></td><td></td><td></td><td>[:IMMediate][:AMPlitude]</td><td><nv></td></tr>
<tr><td></td><td></td><td></td><td>:STEP[:INCRement]</td><td><NRf></td></tr>
<tr><td></td><td></td><td>:MODE</td><td></td><td>CW|FIXed|LIST|SWEep|SWEep1</td></tr>
<tr><td></td><td></td><td>:SPAN</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td></td><td>:FULL</td><td></td></tr>
<tr><td></td><td></td><td>:STARt</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>:STOP</td><td></td><td><nv></td></tr>
<tr><td></td><td>:PULM</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:MODE</td><td></td><td>CW|FIXed|LIST</td></tr>
<tr><td></td><td></td><td>[:STATe]</td><td></td><td>ON|OFF|1|0</td></tr>
<tr><td></td><td>:ROSCillator</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:SOURce</td><td></td><td>INTernal|EXTernal</td></tr>
<tr><td></td><td>:SWEep<n> (n = 1|2)</td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td>:COUNt</td><td></td><td><nv></td></tr>
<tr><td></td><td></td><td>:DIRection</td><td></td><td>UP|DOWN</td></tr>
</table>

```
KEYWORDS                                              ARGUMENTS
[SOURce] (Continued)
          :DWELl                                      <nv>
               :LINK                                  <char>
          :GENeration                                 STEP
          :SPACing                                    LINear
          :STARt                                      SWE1 = <char>
                                                      SWE2 = TRIG:SEQ3
          :STEP
               :FREQuency (SWE1 Only)                 <nv>
               :POWer (SWE2 Only)                     <nv>
STATus
     :OPERation
          :CONDition?
          :ENABle                                     <NRf>
          [:EVENt]?
          :NTRansition<NRf>
          :PTRansition                                <NRf>
     :PRESet
     :QUEStionable
          :CALibration:CONDition?
          :ENABle                                     <NRf>
          [:EVENt]?
          :FREQuency:CONDition?
          :POWer:CONDition?
          :VOLTage:CONDition?
SYSTem
     :ERRor?
     :VERSion?
TRIGger
     [:SEQuence|SEQuence1|STARt]|:SEQuence<n> (n = 2|3|4)
          :COUNt                                      <nv>
          :ECOunt                                     <nv>
          [:IMMediate]
          :SIGNal
          :SLOPe                                      POSitive|NEGative|EITHer
          :SOURce                                     BUS|ECLTrg<n>|TTLTrg<n>
                                                      (For ECLTrg<n>, n = 0|1)
                                                      (For TTLTrg<n>, n = 0|1|2|3|4|5|6|7)
UNIT
     :FREQuency                                       HZ|KHZ|MHZ|GHZ
```

This Page Intentionally Left Blank

APPENDIX C
FUNCTIONAL BLOCK DIAGRAMS

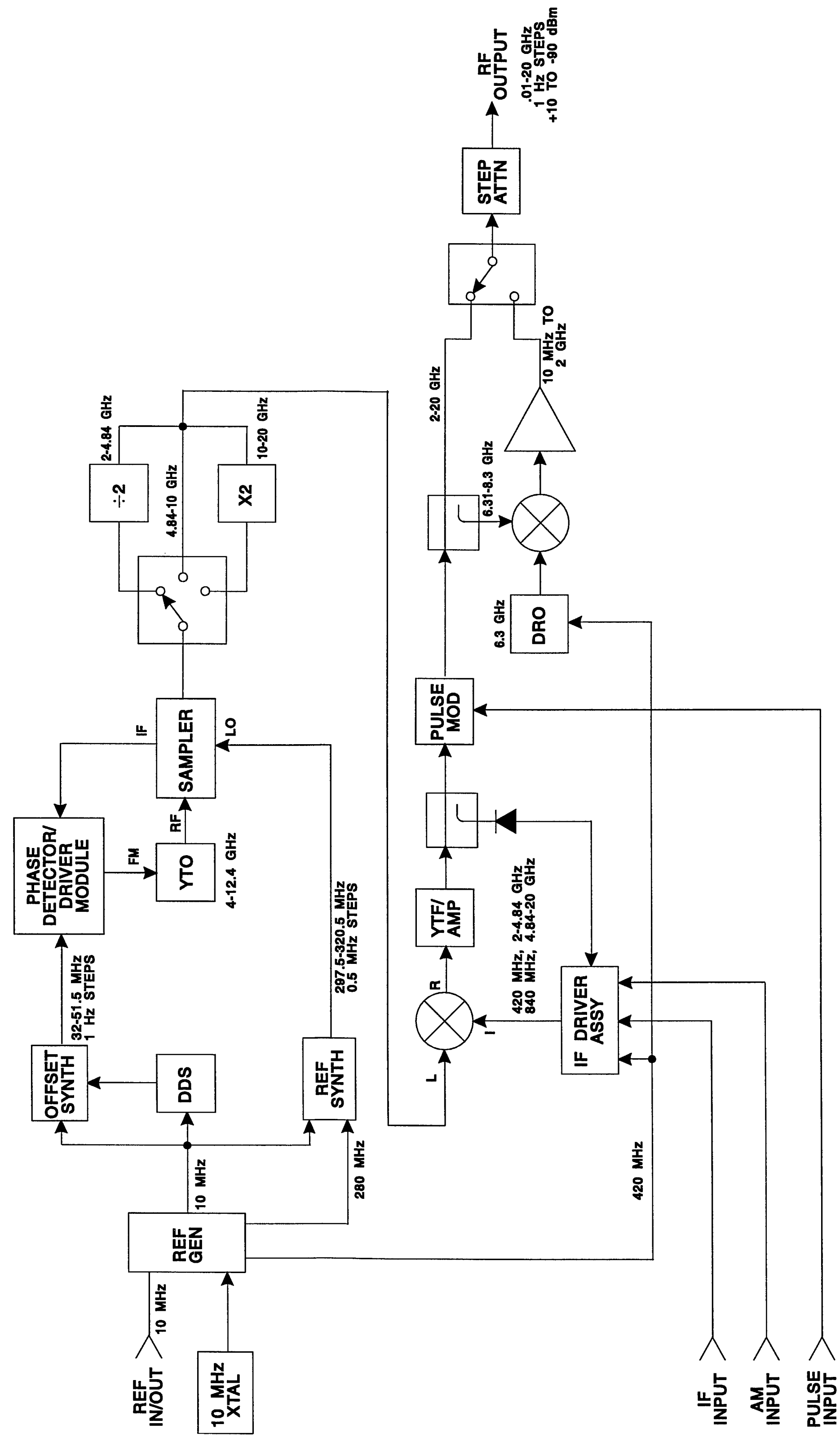


Figure C-1. Model 1140A Functional Block Diagram.

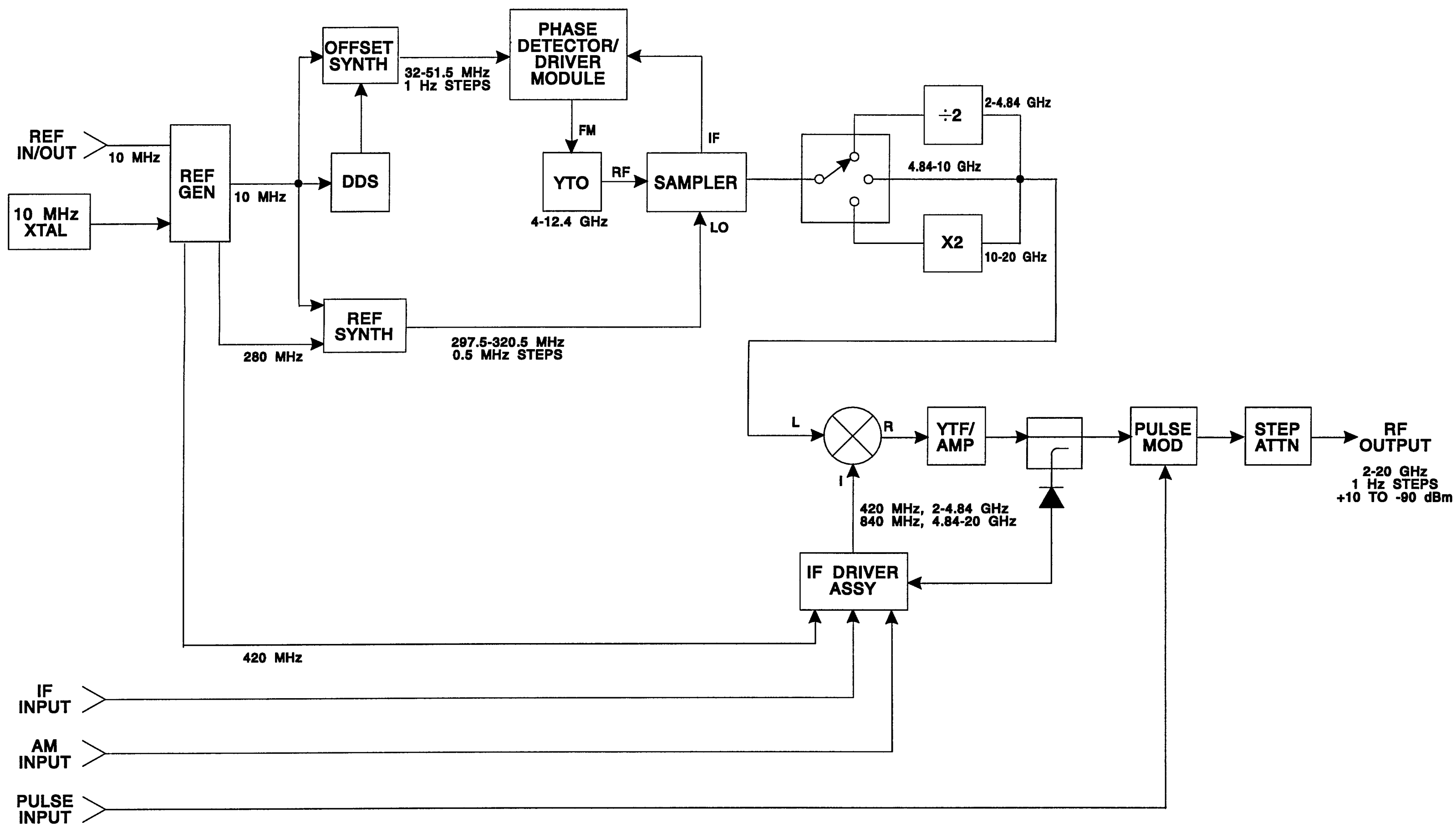


Figure C-2. Model 1141A Functional Block Diagram.

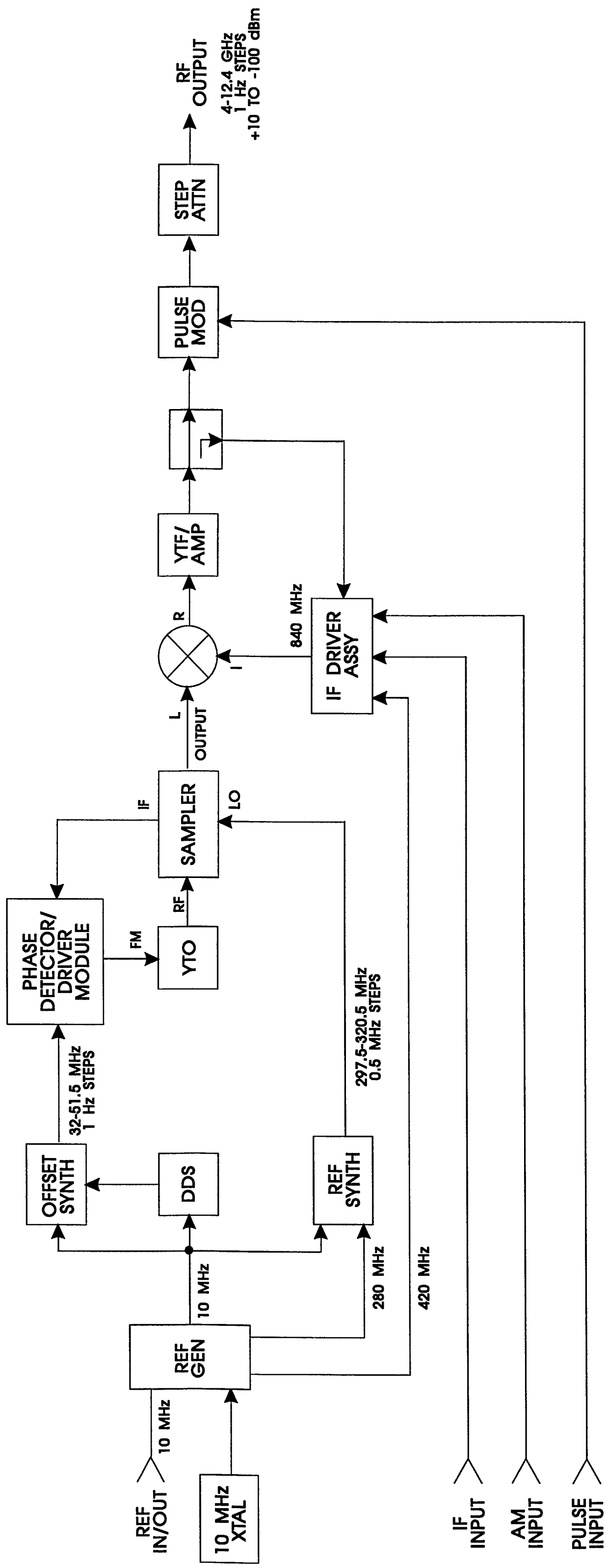


Figure C-3. Model 1142A Functional Block Diagram.

This Page Intentionally Left Blank

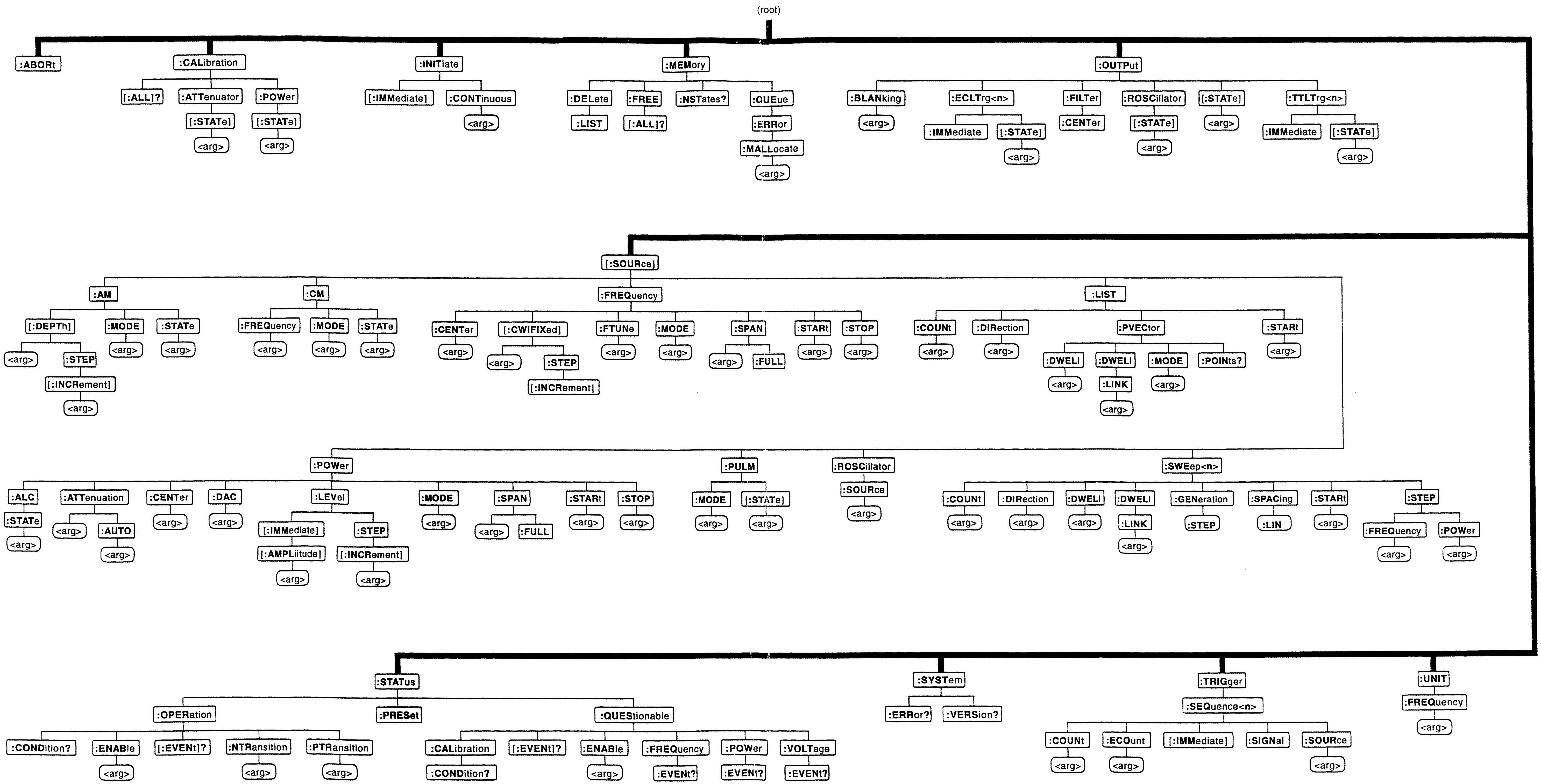


Figure D-1. Command Tree.

INDEX

INDEX (Continued)

I

M

O

P

S

INDEX (Continued)

INDEX (Continued)