ARTISAN®
TECHNOLOGY GROUP

Your **definitive** source for quality pre-owned equipment.

Artisan Technology Group
(217) 352-9330 | sales@artisantg.com | artisantg.com

Full-service, independent repair center
with experienced engineers and technicians on staff.

We buy your excess, underutilized, and idle equipment
along with credit for buybacks and trade-ins.

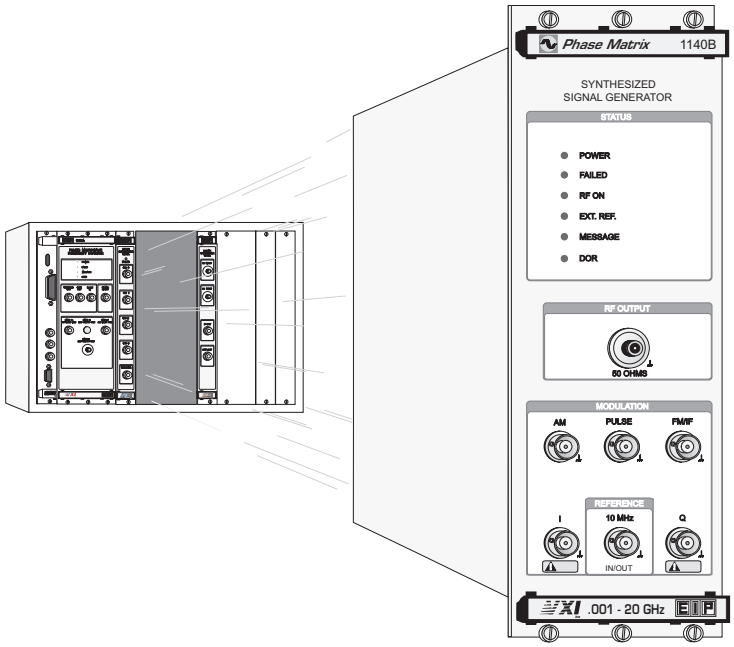
Custom engineering
so your equipment works exactly as you specify.

- Critical and expedited services
- In stock / Ready-to-ship
- Leasing / Rentals / Demos
- ITAR-certified secure asset solutions

Expert team | Trust guarantee | 100% satisfaction

All trademarks, brand names, and brands appearing herein are the property of their respective owners.

Find the *EIP 1140A-SY12* at our website: *Click HERE*



Models 1140B & 1141B VXIbus Synthesized Signal Generators

Operation Manual

1140B CCN 70601
1141B CCN 70701

Manual Assy. Part Number: 5585495-01
Manual Text Part Number: 5580495-01
Printed in USA, December 2004

Manual Prepared By:
TTW
Rev. 12.15.04

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 to Phase Matrix, Inc. or an authorized service facility. In-warranty units will be returned freight prepaid; out-of-warranty units will be returned freight COLLECT. No warranty other than the 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 United States National Institute of Standards and Technology.

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 an Phase Matrix, Inc. instrument is changed 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.

CONTACTING PHASE MATRIX

Phase Matrix, Inc.
Customer Support
109 Bonaventura Drive
San Jose, CA 95134

Tel: +1 (408) 428-1000
Toll free:+1 (877) 474-2736
Fax: +1 (408) 428-1500
Email:service@phasematrix.com
Web:www.phasematrix.com

SAFETY

The Phase Matrix 1140B & 1141B 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 powered on, its protective earth terminals *must* be in contact with the mainframe's protective conductor. The mainframe *must* be connected to a power source 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

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 power is connected to this instrument. Internal adjustments or servicing that must be done with the power connected must be performed only by qualified personnel.



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.



CAUTION

Some static-sensitive components are used in this instrument. These components can be damaged if handled incorrectly.

This Page Intentionally Left Blank

TABLE OF CONTENTS

WARRANTY iii
CERTIFICATION iii
MANUAL CHANGE INFORMATION iii
CONTACTING PHASE MATRIX iii
SAFETY iv
SAFETY AND HAZARD SYMBOLS iv
OVERALL SAFETY CONSIDERATIONS v

SECTION 1:
GENERAL INFORMATION

INTRODUCTION 1-1
OPERATING CONDITIONS 1-1
STORAGE 1-2
1140B SPECIFICATIONS 1-2
1141B SPECIFICATIONS 1-8

SECTION 2:
INSTALLATION

UNPACKING 2-1
SETTING THE LOGICAL ADDRESS 2-1
POWER AND COOLING 2-2
INSTALLATION 2-2
INCOMING OPERATIONAL CHECK 2-3
IF A PROBLEM OCCURS 2-4
SERVICE INFORMATION 2-4
PERIODIC MAINTENANCE 2-4
SIGNAL-GENERATOR IDENTIFICATION 2-4
FACTORY SERVICE 2-4
SHIPPING INSTRUCTIONS 2-5

SECTION 3:
FRONT PANEL CONNECTORS AND INDICATORS

INTRODUCTION 3-1
FRONT PANEL STATUS LEDS 3-2
FRONT PANEL CONNECTORS 3-2

SECTION 4:
PROGRAMMING

REMOTE PROGRAMMING 4-1
SCPI COMMAND STRUCTURE 4-1
CONVENTIONS 4-2
GENERAL NOTATIONS 4-2
DATA FORMAT NOTATIONS 4-3
COMMON COMMANDS, IEEE 488.2 4-4
INSTRUMENT DEPENDENT COMMANDS 4-9
COMMAND PUNCTUATION SUMMARY 4-10
ABORT SUBSYSTEM 4-11
CALIBRATION SUBSYSTEM 4-11
INITIATE SUBSYSTEM 4-16
MEMORY SUBSYSTEM 4-19
OUTPUT SUBSYSTEM 4-24
SOURCE SUBSYSTEM 4-34
TRIGGER SUBSYSTEM 4-122
UNIT SUBSYSTEM 4-129

SECTION 5:
OPERATION AND PROGRAMMING EXAMPLES

INTRODUCTION 5-1
VXIBUS SYSTEM CONFIGURATIONS 5-1
EXTERNAL COMPUTER 5-1
EMBEDDED COMPUTER 5-3
SYSTEM CONFIGURATION 5-3
SAMPLE PROGRAMS 5-5
TIMED FREQUENCY SWEEP 5-5
TIMED-POWER SWEEP 5-7
FREQUENCY-AND-POWER SWEEP 5-8
TRIGGERED FREQUENCY SWEEP 5-9
TRIGGERED LIST 5-9

APPENDIX A:
VXIbus

INTRODUCTION A-1
VMEbus BACKGROUND A-1
THE VXIbus EXTENSIONS A-2
VXIbus MODULES A-2

APPENDIX A:
VXIbus (Continued)

VXIbus SUBSYSTEMS A-3
P2 CONNECTOR DEFINITION A-3
P3 CONNECTOR DEFINITION A-3
VXIbus SYSTEM ARCHITECTURE A-3

APPENDIX B:
COMMMAND SUMMARY

COMMON COMMANDS, IEEE 488.2 . . . . . B-1
INSTRUMENT DEPENDENT COMMANDS B-1

APPENDIX C:
FUNCTIONAL BLOCK DIAGRAMS

FUNCTIONAL BLOCK DIAGRAMS C-1

APPENDIX D:
COMMMAND TREE

INTRODUCTION D-1

APPENDIX E:
POWER SEARCH

INTRODUCTION E-1
POWER SEARCH THEORY E-2
POWER SEARCH ROUTINE - COMPLEX MODULATION
(Units with neither option 02 or option 05) E-3
POWER SEARCH ROUTINE - DIGITAL (I/Q MODULATION)
(Option 05 Units) E-5

1

GENERAL INFORMATION

INTRODUCTION

The Phase Matrix Models 1140B and 1141B VXIbus Synthesized Signal Generators are message-based VXI modules capable of generating microwave signals. The 1140B has a frequency range of 0.001 to 20 GHz with a typical dynamic range of +10 to –90 dBm. The 1141B has a frequency range of 2 GHz to 20 GHz with a typical dynamic range of +10 to –90 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 or optional, a variety of external modulation inputs including: AM, FM, I/Q, Pulse, and complex modulation. The 1140B and 1141B are VXIbus, C size, 3-wide, plug-in modules that require a VXIbus mainframe for operation.

OPERATING CONDITIONS

The Phase Matrix 1140B and 1141B signal generators are designed to operate at temperatures from 0° C to 50° C at a relative humidity not exceeding 95% (75% over 25° C; 45% over 40° C). The signal generators 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 1140B and 1141B signal generators meet the requirements of MIL-T-28800D, Type III, Class7, Style G, Color R with the following modifications and exceptions:

1. The non-operating temperature requirement is limited to the range of –40° C 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 damage to the signal generators, 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.

1140B SPECIFICATIONS

GENERAL

Operating Temperature	0° C to 50° C (32° F to 122° F)
Non-operating Temperature	–40° C to 70° C (–40° F to 158° F)
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 (8.2 kg)

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	1.2 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	–12	–24
I_{PEAK} (Amperes)	2.0	2.8	2.9	0.2	0.2	0.8	2.0

1140B SPECIFICATIONS (Continued)

FREQUENCY

Range	0.001 GHz 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° C to 50° C
Switching Time	<50 ms to within 500 Hz (Triggered List Mode, 2GHz to 20 GHz step)

PROGRAMMING

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

SPECTRAL PURITY
(At +10 dBm CW Output, Complex Modulation OFF)

	1 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	<–40 dBc	<–40 dBc typ.	<–40 dBc typ.	<–40 dBc typ.
Power-Line Related Spurious	<–45 dBc	<–51 dBc	<–45 dBc	<–39 dBc

Non-Harmonically Related Spurious

Offset Frequency	0.001 to 2 GHz	2 to 4.84 GHz	4.84 to 10 GHz	10 to 20 GHz
<100 kHz	<–60 dBc	<–60 dBc	<–60 dBc	<–60 dBc
100 kHz to 1 MHz	<–60 dBc	<–60 dBc	<–60 dBc	<–60 dBc
1 MHz to 100 MHz	<–60 dBc	<–60 dBc	<–60 dBc	<–60 dBc
>100 MHz	<–60 dBc ①	<–70 dBc	<–70 dBc	<–60 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

1140B SPECIFICATIONS (Continued)

SPECTRAL PURITY (Continued)
(At +10 dBm CW Output, Complex Modulation OFF)

Single-Sideband Phase Noise (dBc/Hz) ①

	Frequency (GHz)									
Offset From Carrier	≤2	4	6	8	10	12	14	16	18	20
30 Hz	–78	–83	–82	–78	–77	–75	–72	–71	–70	–68
100 Hz	–81	–85	–84	–81	–80	–78	–78	–77	–77	–75
1 kHz	–82	–90	–88	–87	–85	–83	–82	–80	–80	–80
10 kHz	–80	–90	–88	–88	–85	–82	–82	–80	–80	–80
100 kHz	–88	–97	–92	–88	–88	–88	–84	–82	–80	–80

① Typical performance is >7 dB lower than specified.

RF OUTPUT

Range (Leveled)	Standard	With Option 06
1 MHz to 2 GHz	+13 dBm to –80 dBm	+16 dBm to –80 dBm
2 GHz to 12 GHz	+13 dBm to –90 dBm	+16 dBm to –90 dBm
12 GHz to 18 GHz	+10 dBm to –90 dBm	+16 dBm to –90 dBm
18 GHz to 20 GHz	+8 dBm to –90 dBm	+16 dBm to –90 dBm

Resolution	0.1 dB
Power Accuracy (In CW mode with attenuator atnd ALC coupled)	
1 MHz to 2 GHz	± 2.0 dB, >–50 dBm; ± 3.0 dB, ≤–50 dBm
2 GHz to 20 GHz	±1.0 dB, >–50 dBm; ± 2.0 dB, –50 dBm to –80 dBm; ± 2.0 dB typical, <–80 dBm
Output Level Switching Time (Triggered List mode)	With attenuator change: <150 ms Without attenuator change: <50 ms
Source Impedence	50 Ω nominal
VSWR	< 2.0:1 (0 dB attenuation) typical
Connector	Standard: APC 3.5 female Option 03: APC 3.5 male
Reverse Power Tolerance	1 watt continuous

1140B SPECIFICATIONS (Continued)

10 MHZ INPUT/OUTPUT

Frequency	10 MHz
Level	0 dBm ± 3 dB
Impedence	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	<10 ns, 10% to 90%
Pulse Overshoot, Ringing	<10% for PRFs <1 MHz typical
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 and open-collector compatible
Input Level Tolerance	–0.5 ≤Vin ≤+7.0 Vdc continuous
Polarity	RF output is OFF with a logic 0 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)
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
Input Impedance10 kΩ ± 10%	General Information
Input Level Tolerance	± 20 Vdc continuous
Connector	BNC female

1140B SPECIFICATIONS (Continued)

FREQUENCY MODULATION (External) (Option 02 or Option 05)

Rate	5 kHz to 10 MHz (3 dB bandwidth)
Deviation	25 MHz p-p, f_o 0.05 to 20 GHz 10 MHz p-p, f_o 10 to 50 MHz
Distortion	<5% typical
Sensitivity	5 MHz/volt (fixed)
Input Impedance	600 Ω nominal (AC coupled)
Input VSWR	<2.0:1 typical
Connector	BNC female

DIGITAL (I/Q) MODULATION (External) (Option 05)

I/Q Bandwidth	DC to > 40 MHz
I/Q Sensitivity	0.5 V_p for 100% modulation drive
Input Impedance	600 Ω nominal
Maximum Input Level	1 V_{rms}
Connectors	BNC female (2 each)

IF INPUT (Complex Modulation) (STANDARD UNIT: Option 03 or Option 05 not installed)

Input Frequency	300 MHz to 1 GHz, programmable	
Input Level	–6 dBm nominal	
Instantaneous 3 dB Bandwidth	>40 MHz typical (ALC OFF)	
Spurious Output (+10 dBm output level, ALC ON, –6 dBm input level), typical	IF Input 300 to <750 MHz 750 MHz to 1 GHz	Level <–30 dBc <–60 dBc
Input Impedance	50 Ω nominal	
Input VSWR	< 2.0:1 typical	
Connector	BNC female	

1140B SPECIFICATIONS (Continued)

OPTIONS AND ACCESSORIES

OPTION	DESCRIPTION
02 ①	FM Modulation (external)
03	APC 3.5 mm Male Connector (replaces standard female type)
05 ①	Digital (I/Q) and FM Modulation (external)
06	High Power Output (+16 dBm)

ACCESSORY	DESCRIPTION
Model 1313B	100 kHz – 26.5 GHz VXIbus Microwave Downconverter
Model 1911A	Four Channel VXIbus Pulse Generator
Model 1355A	1 – 20 GHz Microwave Amplifier

① Adding Option 02 or Option 05 replaces the standard IF Input (Complex Modulation).

1141B SPECIFICATIONS

GENERAL

Operating Temperature	0° C to 50° C (32° F to 122° F)
Non-operating Temperature	–40° C to 70° C (–40° F to 158° F)
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 (8.2 kg)

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	1.0 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	–12	–24
I_{PEAK} (Amperes)	2.0	2.8	2.5	0.2	0.2	0.8	1.7

FREQUENCY

Range	2 GHz 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° C to 50° C
Switching Time	<50 ms to within 500 Hz (Triggered List Mode, 2GHz to 20 GHz step)

1141B SPECIFICATIONS (Continued)

PROGRAMMING

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

SPECTRAL PURITY
(At +10 dBm CW Output, 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	–40 dBc typ.	<–40 dBc typ.	<–40 dBc typ.
Power-Line Related Spurious	<–51 dBc	<–45 dBc	<–39 dcB

Non-Harmonically Related Spurious

Offset Frequency	2 to 4.84 GHz	4.84 to 10 GHz	10 t o20 GHz
<100 kHz	<–60 dBc	<–60 dBc	<–60 dBc
100 kHz to 1 MHz	<–60 dBc	<–60 dBc	<–60 dBc
1 MHz to 100 MHz	<–60 dBc	<–60 dBc	<–60 dBc
>100 MHz	<–70 dBc	<–70 dBc	<–60 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) ①

	Frequency (GHz)									
Offset From Carrier	≤2	4	6	8	10	12	14	16	18	20
30 Hz	–78	–83	–82	–78	–77	–75	–72	–71	–70	–68
100 Hz	–81	–85	–84	–81	–80	–78	–78	–77	–77	–75
1 kHz	–82	–90	–88	–87	–85	–83	–82	–80	–80	–80
10 kHz	–80	–90	–88	–88	–85	–82	–82	–80	–80	–80
100 kHz	–88	–97	–92	–88	–88	–88	–84	–82	–80	–80

① Typical performance is >7 dB lower than specified.

1141B SPECIFICATIONS (Continued)

RF OUTPUT

2 GHz to 12 GHz	+13 dBm to –90 dBm	+16 dBm to –90 dBm
12 GHz to 20 GHz	+10 dBm to –90 dBm	+16 dBm to –90 dBm
Resolution	0.1 dB	
Power Accuracy (In CW mode with attenuator atnd ALC coupled)	±1.0 dB, >–50 dBm ± 2.0 dB, –50 dBm to –80 dBm ±2.0 dB typical, <–80 dBm	
Output Level Switching Time (Triggered List mode)	With attenuator change: <150 ms Without attenuator change: <50 ms	
Source Impedence	50 Ω nominal	
VSWR	<2.0:1 (0 dB attenuation) typical	
Connector	Standard: APC 3.5 female Option 03: APC 3.5 male	
Reverse Power Tolerance	1 watt continuous	

10 MHZ INPUT/OUTPUT

Frequency	10 MHz
Level	0 dBm ± 3 dB
Impedence	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	<10 ns, 10% to 90%
Pulse Overshoot, Ringing	<10% for PRFs <1 MHz typical
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)

1141B SPECIFICATIONS (Continued)

PULSE MODULATION
(External) (At +10 dBm Output Power) (Continued)

Input Level	TTL and open-collector compatible
Input Level Tolerance	−0.5 ≤Vin ≤+7.0 Vdc continuous
Polarity	RF output is OFF with a logic 0 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)
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
Input Impedance	10 kΩ ± 10%
Input Level Tolerance	± 20 Vdc continuous
Connector	BNC female

FREQUENCY MODULATION
(External) (Option 02 or Option 05)

Rate	5 kHz to 10 MHz (3 dB bandwidth)
Deviation	25 MHz p-p, f_o 0.05 to 20 GHz 10 MHz p-p, f_o 10 to 50 MHz
Distortion	<5% typical
Sensitivity	5 MHz/volt (fixed)
Input Impedance	600 Ω nominal (AC coupled)
Input VSWR	<2.0:1 typical
Connector	BNC female

1141B SPECIFICATIONS (Continued)

DIGITAL (I/Q) MODULATION (External) (Option 05)

I/Q Bandwidth	DC to > 40 MHz
I/Q Sensitivity	0.5 V_p for 100% modulation drive
Input Impedance	600 Ω nominal
Maximum Input Level	1 V_{rms}
Connectors	BNC female (2 each)

IF INPUT (Complex Modulation) (STANDARD UNIT: Option 02 or Option 05 not installed)

Input Frequency	300 MHz to 1 GHz, programmable	
Input Level	–6 dBm nominal	
Instantaneous 3 dB Bandwidth	>40 MHz typical (ALC OFF)	
Spurious Output (+10 dBm output level, ALC ON, –6 dBm input level), typical	IF Input 300 to < 750 MHz 750 MHz to 1 GHz	Level < –30 dBc < –60 dBc
Input Impedance	50 Ω nominal	
Input VSWR	< 2.0:1 typical	
Connector	BNC female	

OPTIONS AND ACCESSORIES

OPTION	DESCRIPTION
02 ①	FM Modulation (external)
03	APC 3.5 mm Male Connector (replaces standard female type)
05 ①	Digital (I/Q) and FM Modulation (external)
06	High Power Output (+16 dBm)

ACCESSORY	DESCRIPTION
Model 1313B	100 kHz – 26.5 GHz VXIbus Microwave Downconverter
Model 1911A	Four Channel VXIbus Pulse Generator
Model 1355A	1 – 20 GHz Microwave Amplifier

① Adding Option 02 or Option 05 replaces the standard IF Input (Complex Modulation).

DECLARATION OF CONFORMITY

Application Of Council Directive 89/336/EEC

Standards to which Conformity is Declared:

EMC: EN50082-1

Standards to which Compliance is Declared:

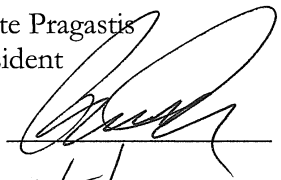
Safety: IEC 1010-1 (1990)

Manufacturer's Name:	Phase Matrix. Inc.
Manufacturer's Address:	109 Bonaventura Drive San Jose, CA 95134
Type of Equipment:	VXI Bus Synthesizer
Model Name(s):	1140A
Tested By:	Rockford Engineering Services, Inc. 9959 Calaveras Road Sunol, CA 94586 USA
Project Engineer:	Mr. Ahmal Safdari
Reviewer:	Mr. Leo Hernandez

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

For: **Phase Matrix**

Name: Pete Pragastis
Title: President

Signature: 

Date: 12/5/02

This Page Intentionally Left Blank

2

INSTALLATION

UNPACKING

The Phase Matrix 1140B and 1141B VXIbus Synthesized Signal Generators arrive ready for operation. Carefully inspect the shipping carton for any sign of damage. If the carton is damaged, immediately notify shipper's agent.

Remove the packing carton and supports, being careful not to mar or damage the instrument. Make a complete visual inspection of the counter, checking for any damage or missing components. Check that all switches and controls operate mechanically. Report any damage to Phase Matrix immediately.

SETTING THE LOGICAL ADDRESS

Before installing the signal generator 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 signal generator is 14 hexadecimal (20 decimal). The logical address of the signal generator is set using the two rotary-type hexadecimal switches located on the bottom of the module (see Figure 2-1 on page 2-2). To set the logical address, set 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 Phase Matrix 1140B and 1141B do not support dynamic configuration.

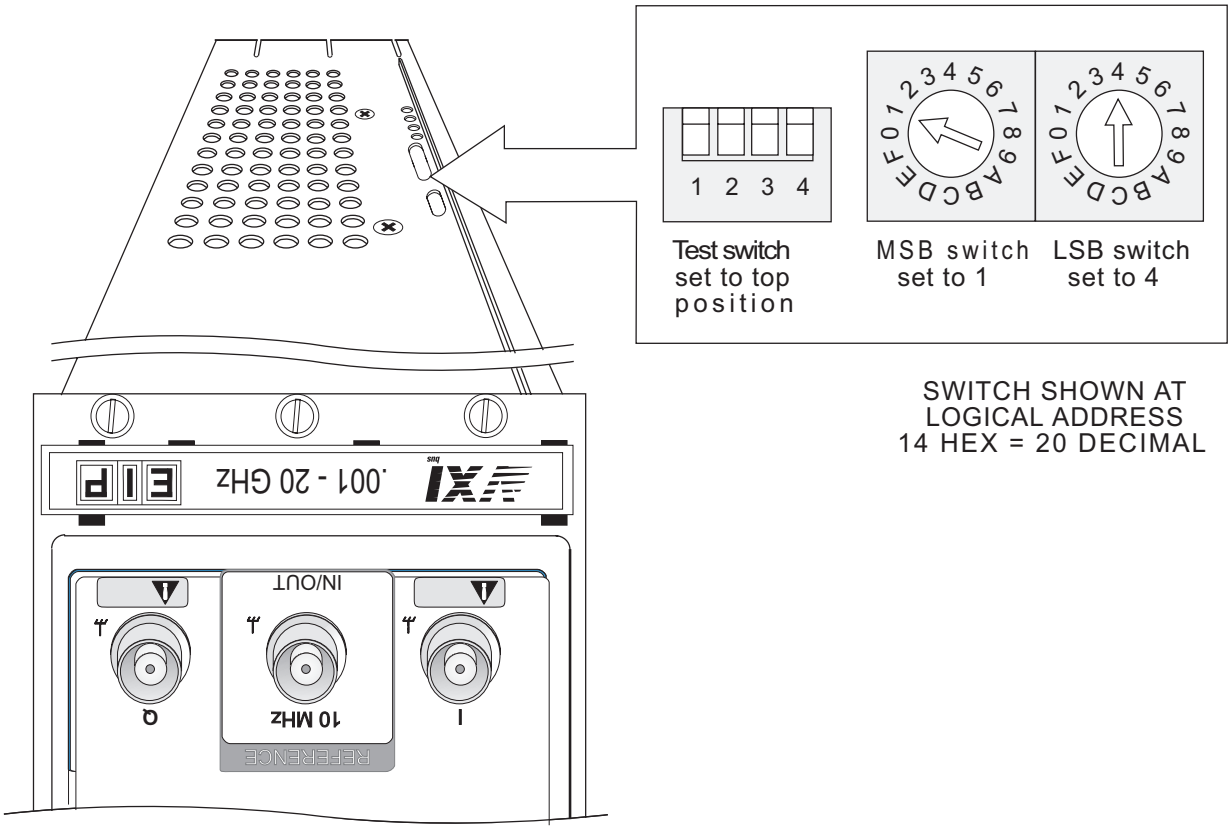


Figure 2-1. Logical Address Switches

POWER AND COOLING

The Phase Matrix 1140B and 1141B signal generators operate over an ambient temperature range from 0° C to 50° C and consume up to 175 watts (see "1140B Specifications" and "1141B Specifications" in Section). When configuring your VXIbus system, make sure that the chassis has sufficient power and cooling capacity for the signal generator. Refer to chassis specifications and cooling capacity curves.

INSTALLATION

The 1140B and 1141B are VXIbus modules designed to be installed in a VXIbus mainframe. Prior to installing the signal generator 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 "1140B Specifications" and "1141B Specifications" in Section).

CAUTION

Never insert the signal generator into a VXIbus mainframe with power applied. Doing so will damage the signal generator.

CAUTION



Verify that all VXI-defined voltages are present and within specification and that the mainframe is capable of supplying the required current before inserting a signal generator in a VXI–bus mainframe.

CAUTION



All unused slots of VXIbus mainframes must have blank panels installed to ensure proper airflow through the installed modules.

The signal generator is a 3-slot, C-size module that can be installed into any slot of a VXIbus mainframe except slot 0, as slot 0 is reserved to the resource manager. Use the following procedure to install the signal generator into the VXIbus mainframe:

1. Turn the mainframe power OFF.
2. Position the signal generator so that its card edges are in the front ends of the mainframe guides (top and bottom).
3. Gently slide the signal generator into the mainframe until you feel resistance as the connectors at the rear of the signal generator begin to mate with the backplane connectors.
4. Firmly seat the signal generator module in the mainframe, making sure that the front panel is flush with the front surface of the mainframe.

CAUTION



Never use the module retaining screws to seat the signal generator in the mainframe. Doing so may damage the module and/or the mainframe chassis.

5. Tighten the retaining screws to ensure the module remains fully seated.

INCOMING OPERATIONAL CHECK

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

1. Observe the state of the Failed LED. Whenever a VXIbus mainframe is energized, the resource manager queries each device checking for proper operation. The signal generator also performs a power-on self-test. The Failed LED lights during the self test or if any problems are detected.
2. Query the instrument regarding its operational status by using a controller to issue the self-test query, ***TST?**. The instrument returns a 0 if all tests pass.

IF A PROBLEM OCCURS

If a problem doeLs occur, make the following checks before returning the instrument for repair:

1. Verify logical address setting on the instrument.
2. Verify that all specified VXI voltages are present.
3. Determine whether the VXIbus AC fail line is asserted. The signal generator monitors this line, and if the line is asserted, the signal generator does not function.
4. If the module has never worked in the particular system, the problem may be in the system rather than in the instrument. If this occurs, contact Phase Matrix Customer Support.

SERVICE INFORMATION

PERIODIC MAINTENANCE

No periodic preventive maintenance is required. However, to maintain accuracy, Phase Matrix recommends that you recalibrate the signal generator every 12 months or whenever a problem is suspected. The specific calibration interval depends upon the accuracy required.

SIGNAL-GENERATOR IDENTIFICATION

The signal generator is identified by three sets of numbers: the model number (Phase Matrix 1140B or 1141B), a serial number, and a configuration-control number (CCN). These numbers, which must be included in any correspondence regarding your signal generator, are on a label affixed to the top of the module.

FACTORY SERVICE

If you are returning the signal generator to Phase Matrix 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 signal generator.
- A complete description of the problem. Be sure to provide enough information so that the problem can be verified:
  - Under what conditions did the problem occur?
  - Did the unit work and then fail?
  - What other equipment was connected to the signal generator when the problem occurred?
- The name and telephone number of someone familiar with the problem who can be contacted by Phase Matrix if any further information is required.
- The shipping address to which the signal generator is to be returned. Include any special shipping instructions.

SHIPPING INSTRUCTIONS

Place the signal generator in an antistatic bag or enclosure, wrap it in heavy plastic or kraft paper, and repack it 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 signal generator and the inner carton. Seal the carton with strong filament tape or strapping. Mark the carton to indicate that it contains a fragile electronic instrument. Ship to Phase Matrix address at the following address:

Phase Matrix, Inc.
109 E. Bonaventura Dr.
San Jose, CA 95134

This Page Intentionally Left Blank

3

FRONT PANEL CONNECTORS AND INDICATORS

INTRODUCTION

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

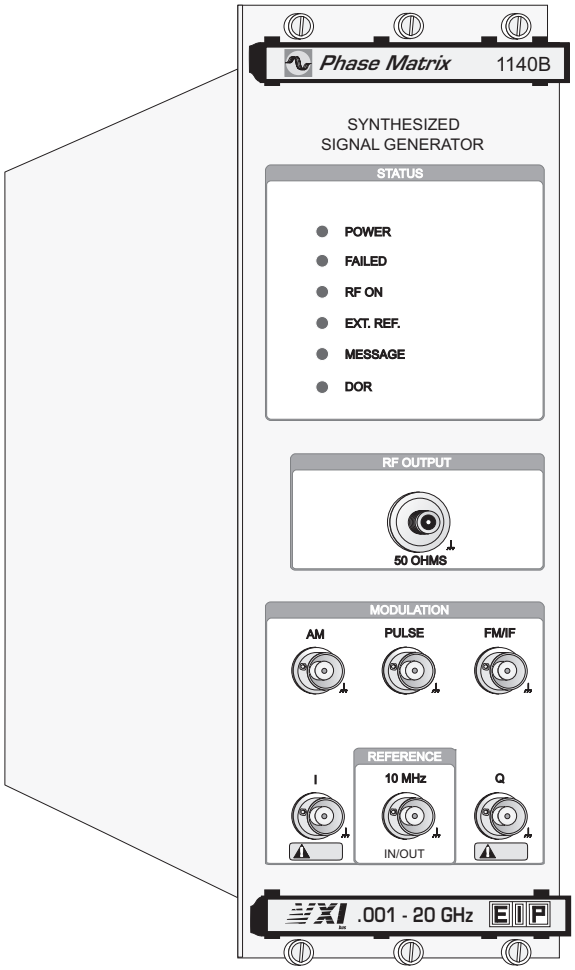


Figure 3-1. 1140B/1141B Front Panel

FRONT PANEL STATUS LEDS

The LEDs on the front panel of the 1140B and 1141B signal generators indicate its status as described below.

- POWER — This LED lights to indicate that power is being supplied to the signal generator. The signal generator senses the voltages on all lines that are used and checks for sufficient voltage. If the required voltages are not present or are not within specification, the LED does not light.
- FAILED — This LED lights to indicate either that the signal generator 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 remains lit even if SYSFAIL is inhibited by the commander.
- RF ON — This LED lights to indicate that microwave power is present on the front panel RF output connector.
- EXT. REF. — This LED lights to indicate that the signal generator is set up to input an external 10 MHz frequency reference from the front panel connector.
- MESSAGE — This LED lights to indicate that the signal generator is either sending or receiving messages or data over the VXIbus. The MESSAGE LED also illuminates when a commander accesses the signal generator's VXIbus registers.
- DOR — This LED lights to indicate that the signal generator is ready to output data.

FRONT PANEL CONNECTORS

This section describes the functions of the front-panel connectors.

- RF OUTPUT — This is the main output connector for the signal generator. The 1140B and 1141B have an APC 3.5 female connector with a nominal 50 Ω impedance.
- MODULATION
  - AM — This is the amplitude-modulation input connector. It is a BNC female connector with 10 kΩ input impedance and a frequency range of dc to 100 kHz. The optimum input level is 2 V p-p, and the Maximum input tolerance is ±20 Vdc continuous.
  - PULSE — This BNC female connector is the pulse-modulation input connector. It is a TTL- or open-collector-compatible input with an input impedance of less than 1 standard TTL load. RF output is OFF with logical 0 input. The frequency range is dc to 10 MHz. Maximum input tolerance is –0.5 to +7 Vdc.
  - FM/IF — This is a BNC female connector that serves as the FM input connector in signal generators with Option 02 or Option 05 installed. The nominal impedance of the FM input is 600 Ω In units with neither option, the input is a complex-modulation IF input. The input impedance is 50 Ω, and the IF frequency can be programmed from 300 Hz to 1 GHz. The optimum input level is –6dBm.

– I and Q — The I and Q inputs are active in Option 05 units when Digital Modulation (DM) mode is selected. These inputs are BNC female connectors with 600 Ω impedance.



CAUTION

To prevent damage to the signal generator, do not exceed 1 V_{rms} on the I and Q inputs.

– INPUT/OUTPUT 10 MHz — This connector is programmable as a reference input or an output. It uses a BNC female connector with a nominal input impedance of 50 Ω . The optimum input level is 0 dBm ±3 dB. The output level is 0 dBm ±3 dB. The maximum input tolerance is +10 dBm continuous.

This Page Intentionally Left Blank

4

PROGRAMMING

REMOTE PROGRAMMING

SCPI COMMAND STRUCTURE

The programming interface for the Phase Matrix 1140B and 1141B signal generators is designed to conform to the requirements of 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 insturmentation. As with any progrmmaing language, exact command syntax must be used. Unrecognised commands are ignored and errors are recorded in memory. It is good programming technique to routinely check the error queue in the signal generator while developing software.

The actual SCPI commands are constructed from individual keywords. The keywords can be a mixture of upper- and lowercase characters. Each keyword has both a long and a short form. The 1140B and 1141B accept only the exact long or short form of the keyword. Sending commands which include keywords that are not the exact long or short version generates an error. The short-form keyword is an abbreviation of the long-form and is usually the first four characters of the long form. However, there is one exception to this rule: when the long form comprises 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 in uppercase letters and is 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 separates alternative parameter options. For example, ON|OFF is the same as ON or OFF.

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

→ Arrows show the path to follow when constructing a signal-generator command.

Boxes enclose required elements of signal-generator commands, such as punctuation.

sp Boxes with sp inside implies whitespace.

Encloses keywords with their acceptable short form in uppercase letters.

Encloses 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|OFFCan 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.

<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

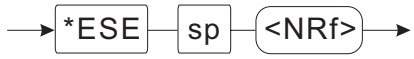
***CLS** (Clear Status Command)



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

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

Default: 0
Range: 0 through 255



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

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

The 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



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

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

Range: 0 through 255



This query returns the value of the Standard Event Status register and clears the value after being read. Refer to the "Status Subsystem" section of this manual for more information.

***IDN?** (Identification query)



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

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

Example:

PHASE MATRIX,1141B,00122,V1.1

***OPC** (Operation Complete command)



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)



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

***OPT?**

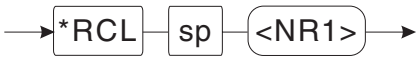


This query returns a list of the options installed in the signal generator. The available options are:

0	No options installed
02	Frequency Modulation
05	Digital (I/Q) Modulation and Frequency Modulation
06	High Power
SY12	SY12 version of the 1140B (Includes FM)

***RCL** (Recall command)

Range: 0



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

For example, to recall the signal-generator setting stored in register 0, send the command:

***RCL 0**

Related Commands:

***SAV** (Save command)

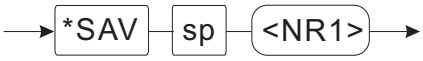
***RST** (Reset command)



This command sets all user programmable variables to default values. It does not affect the VXI interface, output queue, status register, standard event enable register, or calibration data.

***SAV** (Save command)

Range: 0



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

For example, to save the current signal-generator setup parameters in register 0, send this command:

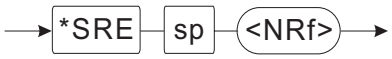
***SAV 0**

Related Command:

***RCL** (Recall command)

***SRE** (Service Request Enable command)

Range: 0 through 255
Default: 0



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

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

Range: 0 through 255



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 this manual for more information.

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

Range: 0 through 255



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

***TRG** (Trigger command)



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

***TST?** (Self-Test query)



This query causes the signal generator 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 (1s) in the binary bit pattern per the following list (on page 4-8).

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 indicates a 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 the *Phase Matrix 1140B & 1141B Service Manual* for service information.

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



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 signal generator 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 signal generator is shown below. The keywords listed under the root are known as subsystems.

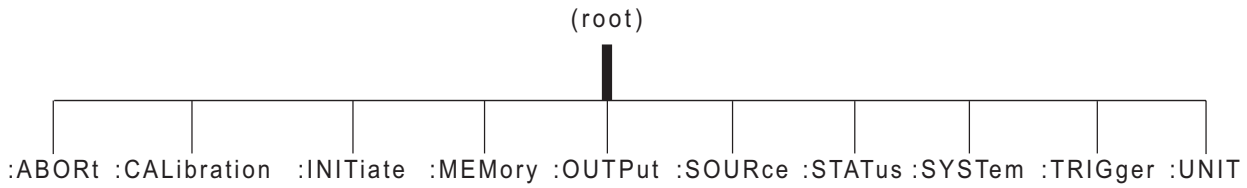


Figure 4-1. Root Command Tree

In developing specific commands for the signal generator, the first element of the command must be one of the sub-system keywords shown above. The keyword is 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 signal generator. Since this operation deals with an output from the signal generator, it would be listed under the **Output** subsystem. A partial command tree for the **Output** subsystem is shown below.

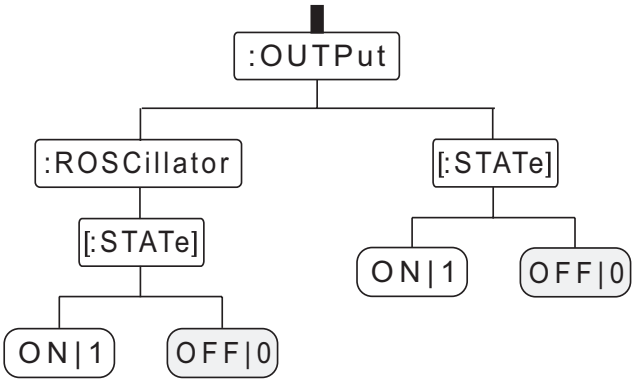


Figure 4-2. Sample Subsystem Command Tree

To build the command to turn on the RF output power from the signal generator, 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 signal generator, 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 this command:

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

The colons shown preceding each of the keywords are not actually part of the command. They tell the signal-generator 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 signal-generator 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 **OUTP**ut keyword. If it is left out, as shown below, an error is generated.

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

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

COMMAND PUNCTUATION SUMMARY

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

SEMICOLON — A semicolon flags the signal-generator firmware that a new command follows.

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 separates multiple command arguments.

ABORT SUBSYSTEM

The Abort Subsystem provides control over the Trigger subsystem. It comprises only one command that forces each of the four trigger sequences back to the Idle state upon completion of the current **:SOURce:LIST** or **:SWEep :STEP** command.



Figure 4-3. Abort Subsystem Command Tree

Example:

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

:ABORt

Related Commands:

See "Trigger Subsystem."

CALIBRATION SUBSYSTEM

The Calibration Subsystem includes routines for developing and controlling signal-generator 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.

Phase Matrix recommends that you calibrate the YIG-tuned oscillator and YIG-tuned filter when the signal generator is first installed in the mainframe, or whenever it is removed and reinstalled.

The signal generator defaults to using the calibration tables for both the leveling loop and the microwave attenuator. During normal operation, these tables must not be disabled.

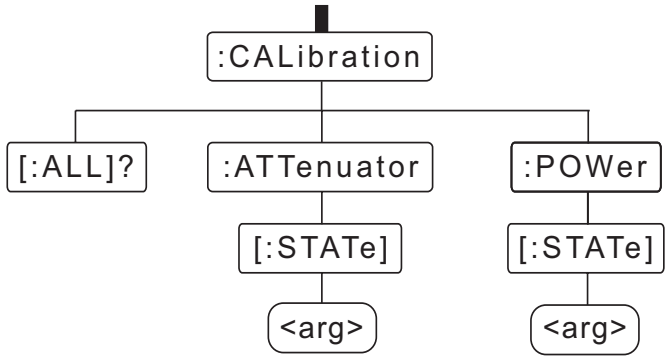
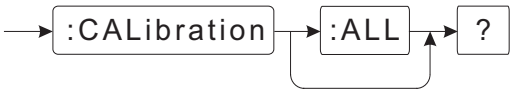


Figure 4-4. Calibration Subsystem Command Tree

CALibration**:ALL?**



This is a fully automatic calibration routine that requires no special setup. However, ***DO NOT*** run the calibration routine until the signal generator is allowed to warm up for at least 15 minutes. Phase Matrix recommends that this calibration be performed when the signal generator is first installed in the mainframe and 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 is returned. If the calibration routine fails, a number is 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 (1s) in the binary bit pattern per the following list.

Weight

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

2 bit 1: Invalid RAM YIG calibration constants. This indicates that calibration was attempted, but the data returned are 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, this 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 the *Phase Matrix 1140B & 1141B Service Manual* for service information.

Example:

To start self-calibration, send this command:

:CAL:ALL?

NOTE

Use of the **:ALL** parameter is optional.

CAUTION

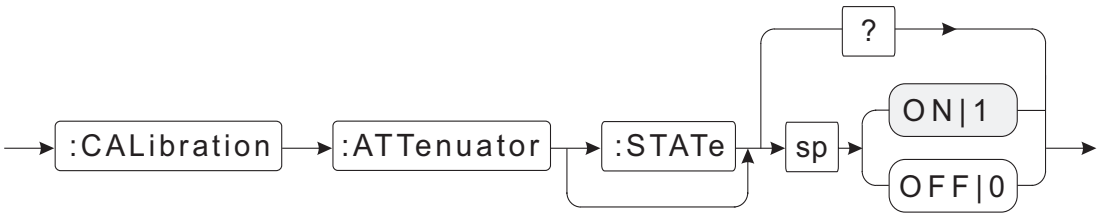
Do not attempt to calibrate the signal generator until it is allowed to warm up for fully 15 minutes.

NOTE

It takes approximately one minute to perform this calibration.

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

Default: ON



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

Examples:

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

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

   or

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

   NOTE

   Use of the **:STATE** parameter is optional.

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

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

   or

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

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

   :CALibration**:ATT**enuator?

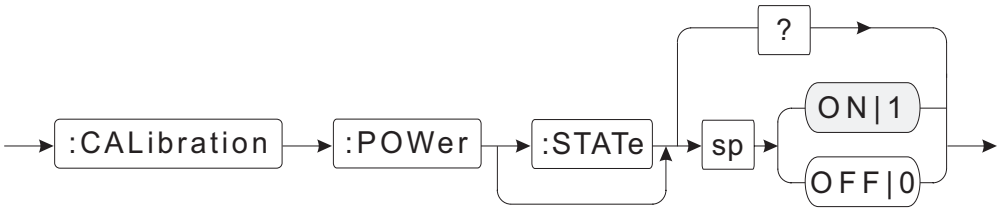
After the signal generator receives the query command, a read command returns the current state of the attenuator power calibration table in <bNR1> format. For example, if the calibration table is enabled and the signal generator is queried for the current state, it returns **1,** indicating that the attenuator calibration table is enabled. If **0** is returned, 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 are applied. The power calibration tables comprise a table for the power leveling loop and a table for the internal attenuator. Turning off power calibration disables both of these tables. The use of the calibration tables defaults to ON (enabled); they are turned off only during calibration and in some other special circumstances.

Examples:

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

   :CALibration:POWer:STATe ON

   or

   :CALibration:POWer:STATe 1

   NOTE

   Use of the **:STATE** parameter is optional.

2. To disable the calibration tables, send this command:

   :CALibration:POWer OFF

   or

   :CALibration:POWer 0

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

   :CALibration:POWer?

After the signal generator receives the query command, a read command returns the current state of the power calibration tables in <bNR> format. For example, if the calibration tables are enabled and the signal generator is queried for the current state, it returns **1**, indicating that the power calibration tables are enabled. If a **0** is returned, the power calibration tables are disabled.

Related Command:

:CALibration:ATTenuator:STATe

NOTE

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

INITIATE SUBSYSTEM

The signal generator 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 signal generator is simply waiting to be commanded to the Armed state. The Armed state is where trigger detection occurs. Once triggered, the signal generator 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 signal generator to move from the Idle state to the Armed state.

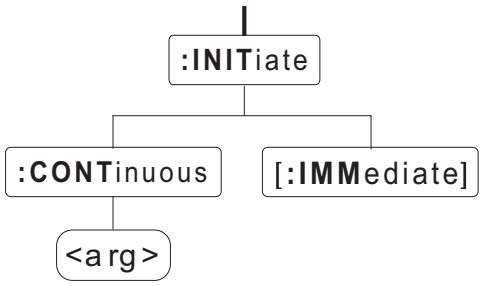
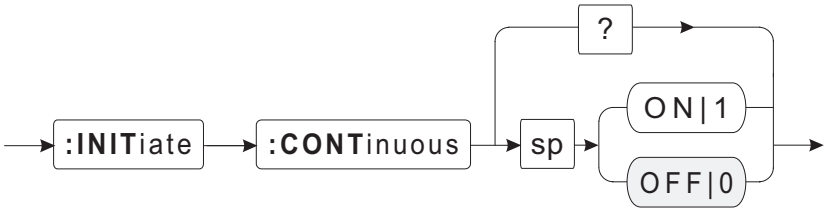


Figure 4-5. Initiate Subsystem Command Tree

Related Commands: See “Trigger Subsystem.”

INITiate**:CONT**inuous

Default: OFF



The signal generator uses a three-layer trigger model consisting of Idle, Armed, and Triggered states. This command controls the movement of the signal generator from the Idle state to the Armed state. It defaults to OFF, but when turned on, the signal generator bypasses the Idle state and moves 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 signal generator to automatically move through the Idle state to the Armed state, send this command:

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

2. To command the signal generator back to the default mode, send this command:

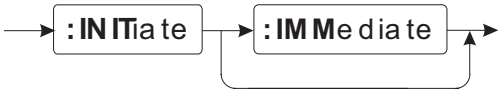
   :INITiate**:CONT**inuous **OFF**

To query the signal generator for the current state, send this command:

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

3. After the signal generator receives the query command, a read command returns the state in <bNR1> format. For example, if it is currently in the default state, and the signal generator is queried for the current state, it returns **0,** indicating that the signal generator is set to remain in the Idle state until specifically commanded to the Armed state.

INITiate**:IMM**ediate



This command immediately moves the signal generator out of the Idle state and into the Armed state. If the signal generator is not idle or if **:INIT:CONT** is ON, it is not valid and the signal generator returns error **–213**.

Example:

To command the signal generator to move immediately from the Idle state to the Armed state, send this command:

:INITiate**:IMM**ediate

or just

:INIT

Related Commands: See "Trigger Subsystem."

MEMORY SUBSYSTEM

The Phase Matrix 1140B and 1141B signal generators include internal memory that stores a signal-generator 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 can be reallocated to increase the number of possible Trigger List data points.

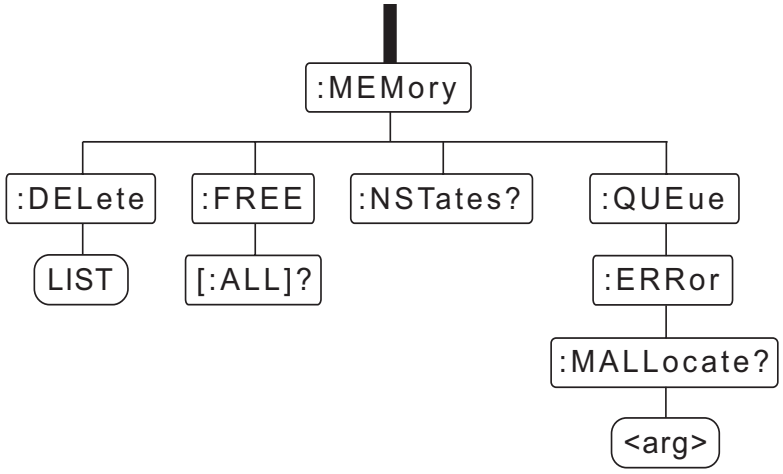


Figure 4-6. Memory Subsystem Command Tree

Related Commands: See “Trigger List.”

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



The signal generator provides a Triggered List feature that allows it to step through a list of up to 201 points. Each of these points can 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 must be executed prior to entering new Triggered List data.

Example:

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

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

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

Related Command:

MEMory **FREE:ALL?**

MEMory**:FREE:ALL?**



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

Example:

To query the signal generator the number of bytes of RAM currently available, send this command:

:MEMory**:FREE:ALL?**

NOTE

Use of the :ALL parameter is optional.

After the signal generator receives the query command, a read command returns the number of bytes available in <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**:NST**ates



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

Example:

To query the signal generator for the number of signal-generator setup states available, send this command:

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

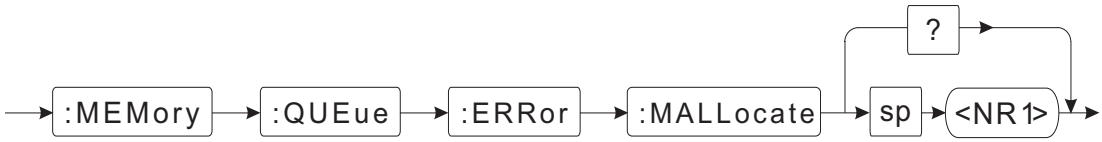
After the signal generator receives the query command, a read command returns **1** in <NR1> format.

Related Commands:

***SAV**
***RCL**

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

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 signal-generator reset.

Examples:

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

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

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

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

3. To query the signal generator for the number of messages in the error queue, send this command:

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

After the signal generator receives the query command, a read command returns the depth of the error queue in <NR1> format. For example, a typical response to this query is **30**, indicating that the SCPI error queue can hold up to 30 messages.

Related Command:

:MEMory:**FREE?**

OUTPUT SUBSYSTEM

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

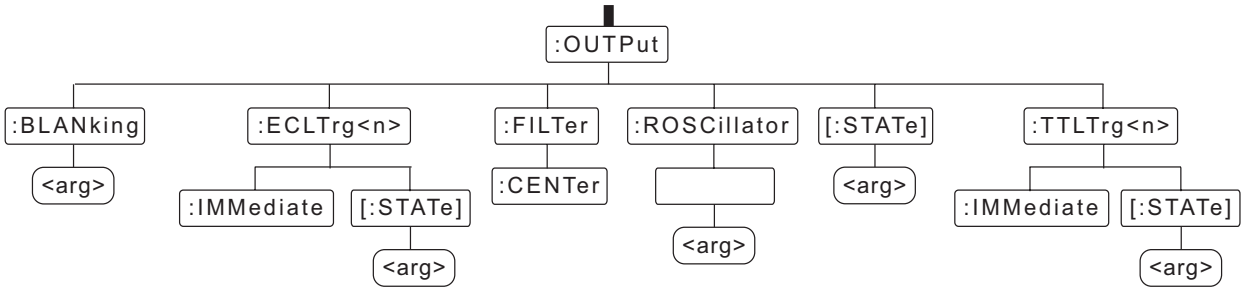
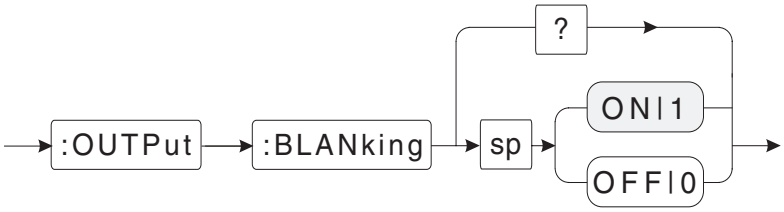


Figure 4-7. Output Subsystem Command Tree

OUTPut**:BLAN**king

Default: ON



The RF output from the signal generator is normally turned off (blanked) during frequency and power changes. The **:OUTP**ut**:BLAN**king command controls output blanking.

Examples:

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

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

2. To enable output signal blanking, send this command:

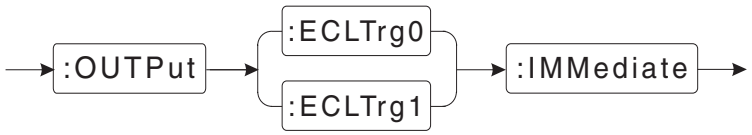
   :OUTPut**:BLAN**king **ON**

3. To query the signal generator for the current state of output blanking, send this command:

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

After the signal generator receives the query command, a read command returns the state of output blanking in <bNR1> format. For example, if blanking is enabled and the signal generator is queried for the current state of blanking, it returns **1**, indicating that blanking is enabled (ON).

OUTPut:ECLTrg<n>:IMMediate



This command causes the signal generator 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 this command:

   :OUTPut:ECLTrg0:IMMediate

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

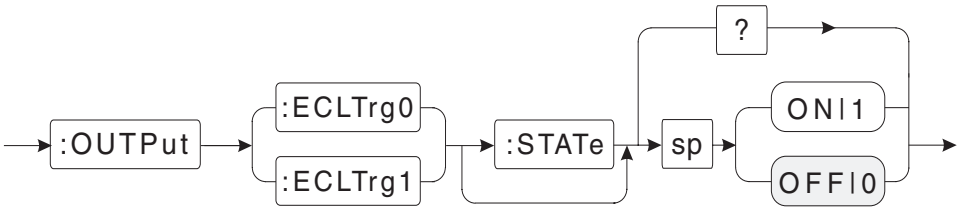
   :OUTPut:ECLTrg1:IMMediate

Related Command:

:OUTPut:ECLTrg<n>:STATe

OUTPut:ECLTrg<n>:STATe

Default: OFF



This command enables one of the two ECL triggers (0...1). The synchronous protocol is implemented on the signal generator. The synchronous trigger protocol is a single-line broadcast trigger that does not require any acknowledgment from the acceptors. The signal generator 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 this command:

   :OUTPut:9:ECLTrg0:STATe ON

   or

   :OUTPut:ECLTrg0 1

   NOTE

   Use of the :STATe parameter is optional.

2. To enable ECLTrg1, send this command:

   :OUTPut:ECLTrg1:STATe ON

3. To query the signal generator for the current state of ECL trigger 1, send this command:

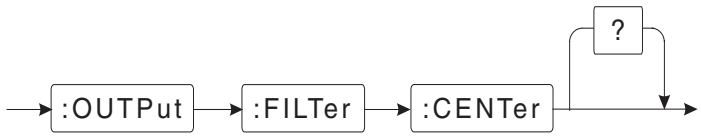
   :OUTPut:ECLTrg1:STATe?

After the signal generator receives the query command, a read operatoin returns the state of the selected ECL trigger line in <bNR1> format. For example, if ECLTrg was previously enabled, and the signal generator is queried for the current state of ECLTrg1, it returns **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 signal generator to center on the output signal. The YTF eliminates (filters 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 that 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 the signal generator receives 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 signal generator 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 the signal generator receives the filter-center command, sending the query form of the command causes the signal generator to output the frequency shift that occurred as a result of the filter -entering routine.

Examples:

1. To command the signal generator to manually center the YTF on the output signal, send this command:

   :OUTPut:FILTer:CENTer

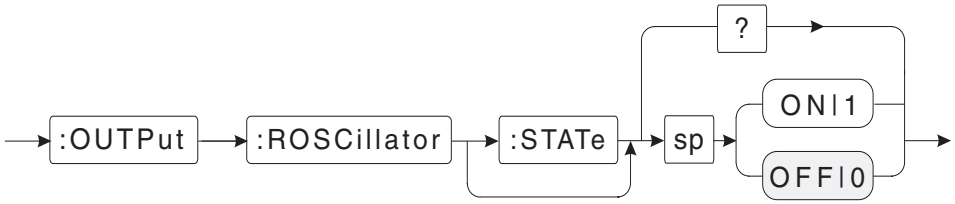
2. To find out how far the filter was moved in response to the filter center command, send this query command:

   :OUTPut:FILTer:CENTer?

   After the signal generator receives the query command, a read command returns the frequency shiftin <NR1> format. For example, if the filter center command caused the center frequency of the filter to increase by 1 MHz, the query returns **1000000**, indicating a positive shift of 1 MHz in the YTF center frequency. If the new center frequency is less than the original center frequency, the frequency shift is 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 this command:

   :OUTPut:ROSCillator:STATe ON

   NOTE

   Use of the **:STATe** parameter is optional.

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

   :OUTPut:ROSCillator OFF

3. To query the signal generator for the status of the the front panel 10 MHz connector, send this command:

   :OUTPut:ROSCillator?

After the signal generator receives the query command, a read command returns the state of the reference oscillator in <bNR1> format. For example, if the reference oscillator was previously set to ON, and the signal generator is queried for the current state, it returns **1**, indicating that 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 that the reference-oscillator source is set to internal **(:SOUR:ROSC:SOUR INT)**. If this is not done, a settings conflict error occurs.

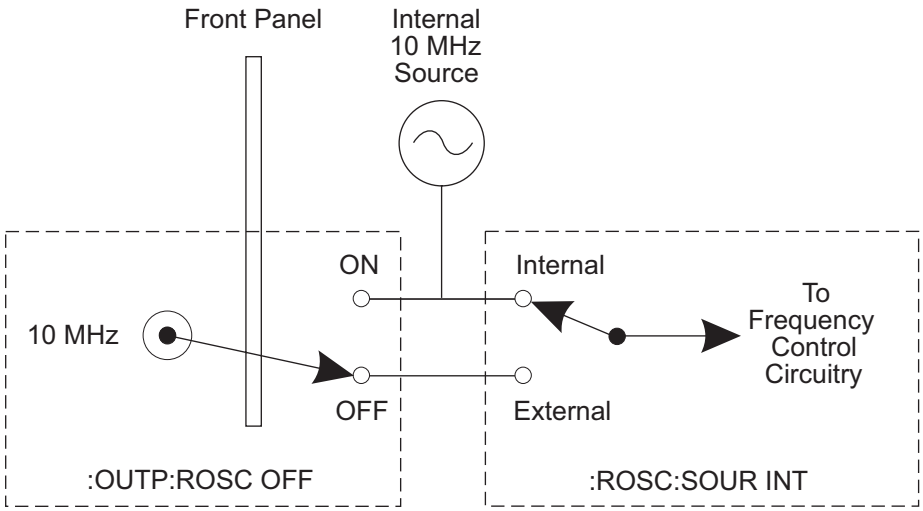
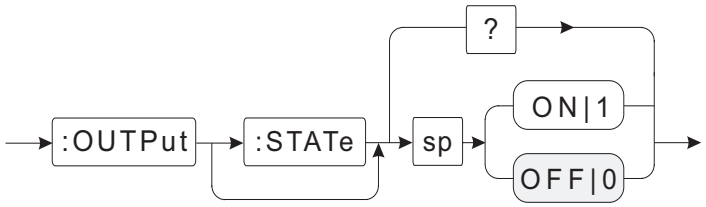


Figure 4-8. Reference Oscillator Selection Circuit

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 this command:

   :OUTPut:STATe ON

   or

   :OUTP ON

   NOTE

   Use of the **:STAT**e parameter is optional.

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

   :OUTPut:STATe OFF

   or

   :OUTP OFF

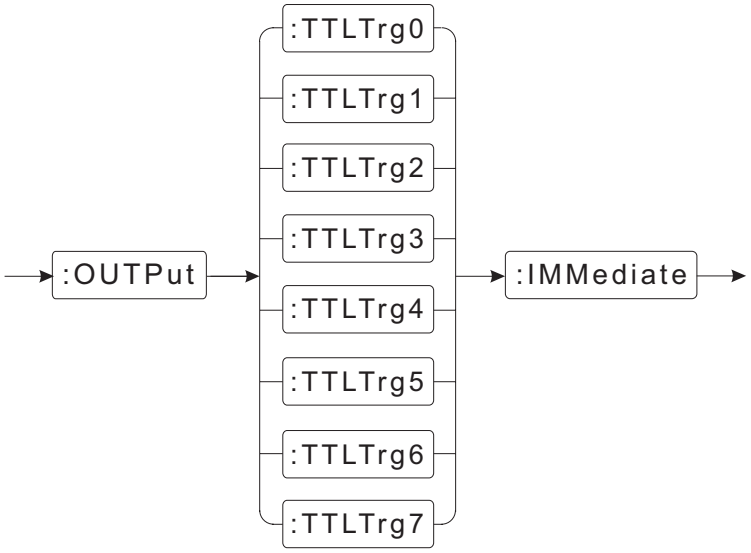
3. To query the signal generator for the current state of the RF output, send this command:

   :OUTPut:STATe?

After the signal generator receives the query command, a read command returns the state of the front panel RF OUTPUT connector in <bNR1> format.

For example, if RF output was previously enabled, and the signal generator is queried for the current state, it returns **1**, indicating that the RF output is enabled (ON).

OUTPut**:TTL**Trg<n>**:IMM**ediate



This command causes the signal generator to immediately output an active high TTL pulse that is 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 **:OUTPut:TTLTrg<n>:STATe** command.

Examples:

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

   :OUTPut:TTLTrg0:IMMediate

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

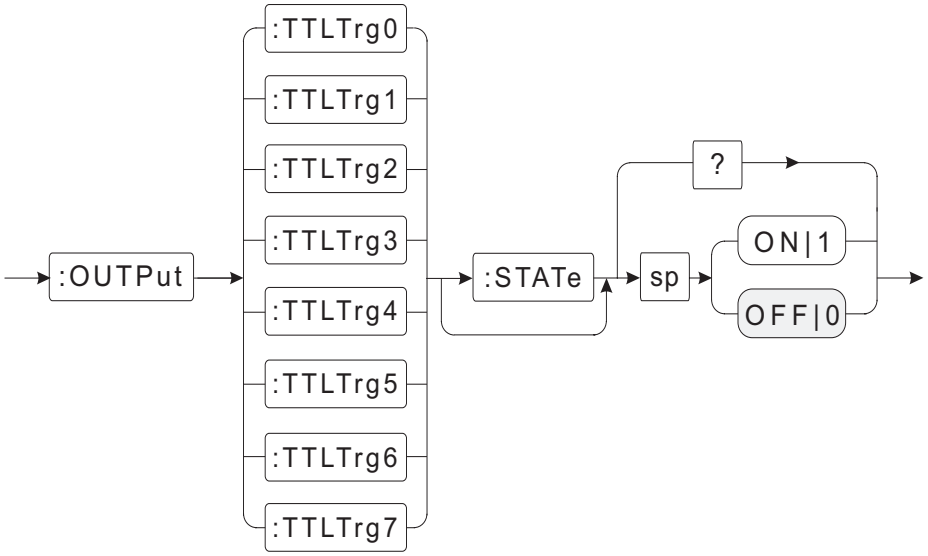
   :OUTPut:TTLTrg7:IMMediate

Related Command:

:OUTPut:TTLTrg<n>:STATe

OUTPut:TTLTrg<n>:STATe

Default: OFF



This command enables one of the eight TTL triggers (TTLTrg0...TTLTrg7). The synchronous protocol is implemented on the signal generator. The synchronous trigger protocol is a single-line broadcast trigger that does not require an acknowledge from any acceptors. The signal generator 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 this command:

   :OUTPut:TTLTrg0:STATe ON

   or

   :OUTP:TTLTrg0 ON

   NOTE

   Use of the **:STAT**e parameter is optional.

2. To enable TTLTrg1, send this command:

   :OUTPut:TTLTrg1:STATe ON

3. To query the signal generator for the current state of TTL trigger 1, send this command:

   :OUTPut:TTLTrg1:STATe?

After the signal generator receives the query command, a read command returns the state of the selected TTL trigger line in <bNR1> format. For example, if TTLTrg was previously enabled, and the signal generator is queried for the current state of TTLTrg1, it returns **1**, indicating that TTLTrg1 is enabled (ON). This does not mean the trigger line is active — only that it is enabled.

Related Command:

:OUTPut**:TTLT**rg<n>**:IMM**ediate

SOURCE SUBSYSTEM

The Source Subsystem provides control of a variety of signal-generator source settings: sdpower, modulation, and selecting the source of the reference oscillator used by the signal generator.

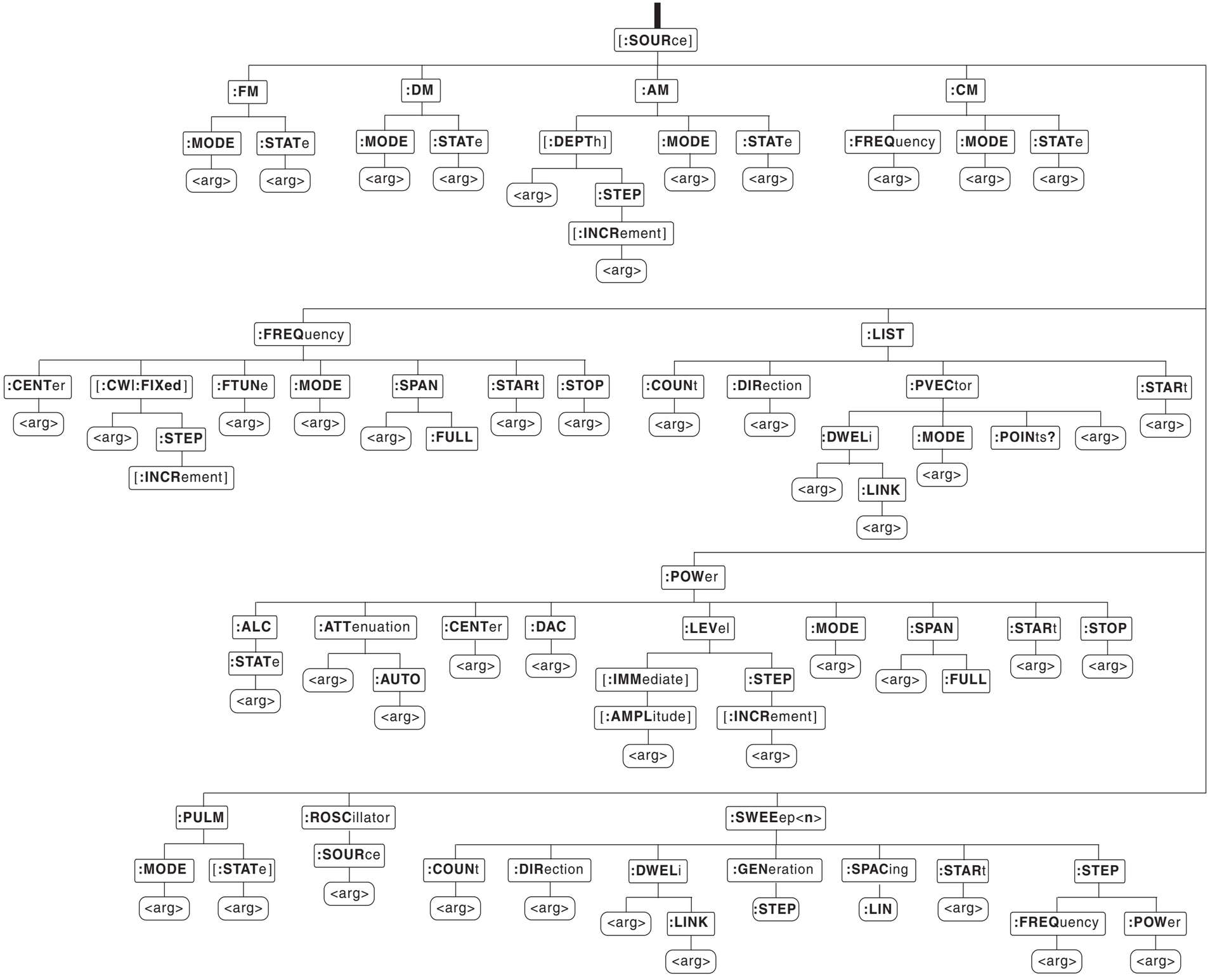
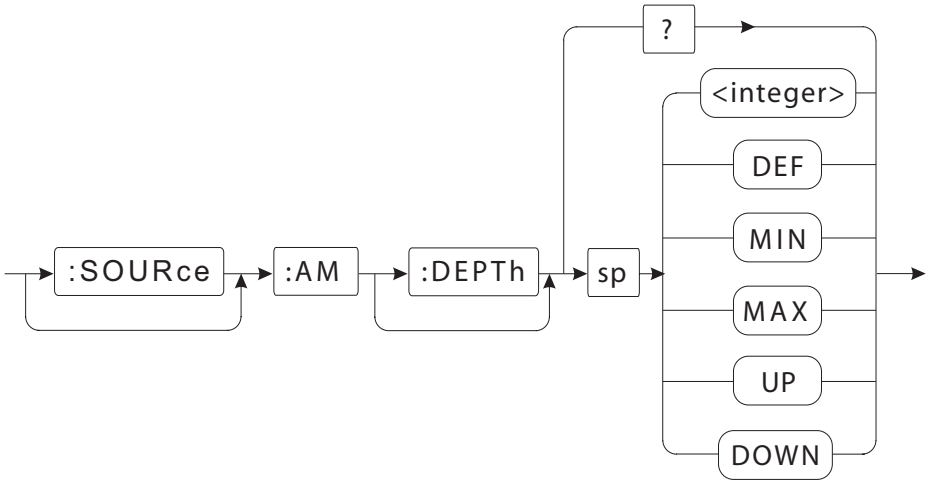


Figure 4-1. Source Subsystem Command Tree

This Page Intentionally Left Blank

SOURce:AM:DEPTh

Default: 0
Range: 0 to 100%
Resolution: 1%



This command sets the depth of an amplitude-modulated signal. Depth is specified in percent (%) in increments of 0.1% (the percent sign is optional). Specification of tenths of a percent is optional, and depths can be specified simply as an integer (55% instead of 55.0%, for example). Additional arguments to this command are **MIN**imum (0.0%), **MAX**imum (100.0%), and **DEF**ault (also 0.0%). You can also use the arguments **UP** and **DOWN** to increment or decrement the depth of modulation by a quantity set by the command **:SOURce:AM:DEPTh:STEP:INCR**ement.

NOTE

Use of the percentage sign (%) and the parameters **:SOUR**ce and **:DEPT**h are optional.

Examples:

1. To set the amplitude-modulation depth to 50 percent, send this command:

   :SOURce:AM:DEPTh 50.0

   or

   :AM 50%

2. To set the amplitude-modulation depth to 0.0% (effectively turning amplitude modulation off), send this command:

   :SOURce:AM:DEPTh MIN

   or

   :AM DEF

3. To set the amplitude modulation to a depth of 100.0%, send this command:

   :SOURce:AM:DEPTh 100

   or

   :AM MAX

4. To increment the depth of amplitude modulation by a quantity set by the **:SOURce:AM:DEPTh:STEP:INCRement** command, send this command:

   :AM UP

5. To decrement the depth of amplitude modulation by a quantity set by the **:SOURce:AM:DEPTh:STEP:INCRement** command, send this command:

   :SOURce:AM:DEPTh: DOWN

6. To query the signal generator for the current setting of the amplitude modulation depth, send this command:

   :SOURce:AM:DEPTh?

   or

   :AM?

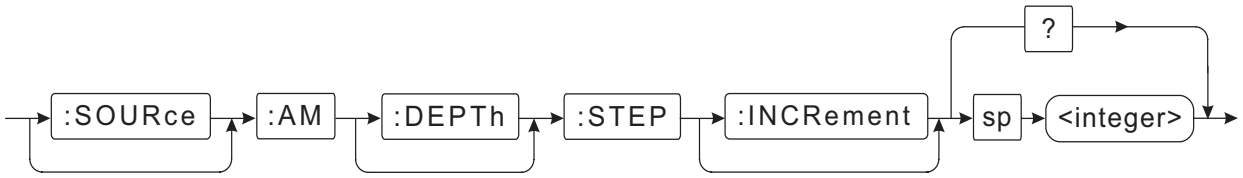
After the signal generator receives the query command, a read command returns the current amplitude-modulation depth setting in <NR1> format.

For example, if the amplitude modulation depth is set to 40% and the signal generator is queried for the current amplitude modulation depth, it returns **40**, indicating that 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: 0.1 to 100%
Resolution: 0.1 dB

This command sets the step size for incrementing or decrementing the depth of an amplitude-modulated signal. The step size is specified in percent (%) in increments of 0.1% (the percent sign is optional). Specification of tenths of a percent is optional, and step size can be specified simply as an integer (5% instead of 5.0%, for example). The increment value can be either positive or negative, and no range checking is performed.

NOTE

Use of the percentage sign (%) and the parameters **:SOURce**, **:DEPTh**, and **:INCRement** are optional.

Examples:

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

   :SOURce:AM:DEPTh:STEP:INCRement

   or

   :AM:STEP 10%

2. To query the signal generator for the current amplitude-modulation step size, send this command:

   :SOURce:AM:DEPTh:STEP:INCRement?

   or

   :AM:STEP?

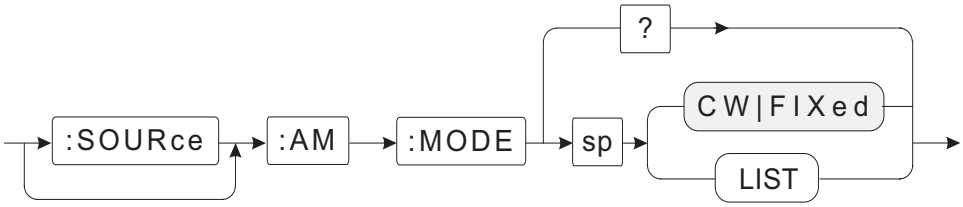
After the signal generator receives the query command, a read command returns the current amplitude-modulation step setting in <NR1> format. For example, if the amplitude modulation step is set to 40% and the signal generator is queried for the current amplitude-modulation step size, it returns **40**, indicating that the amplitude modulation step is set to 40%.

Related Commands:

:SOURce:AM:DEPTh
:SOURce:AM:STATe

SOURce:AM:MODE

Default: CW|FIXed



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

The **:SOURce:LIST** commands are included to provide compatibility with SCPI-defined syntax, and they are fully functional, but they are typically not used. The **:SOURce:LIST:PVECtor:MODE LIST** command is normally used for coupling to **LIST** mode, because it couples AM as well as CM, DM, FM, FREQ, POWer, and PULM (pulse modulation).

> **NOTE**
>
> Use of the **:SOURce** parameter is optional.

Examples:

1. To couple amplitude modulation to **LIST** mode, send this command:

   :SOURce:AM:MODE LIST

2. To query the signal generator for the current mode setting, send this command:

   :SOURce:AM:MODE?

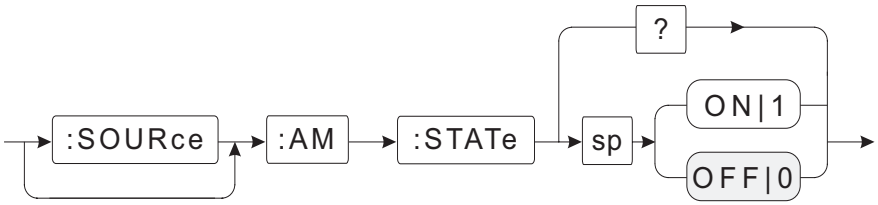
After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if the AM mode was previously set to **LIST** mode, and the signal generator is queried for the current mode, it returns **LIST**, indicating that the state of amplitude modulation is currently being controlled by the triggered-list function.

Related Commands:

:SOURce:AM:STATe
:SOURce:LIST:PVECtor:MODE LIST

SOURce:AM:STATe

Default: OFF



This command turns amplitude modulation on or off.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To enable the amplitude-modulation input, send this command:

   :SOURce:AM:STATe ON

   or

   :SOURce:AM:STATe 1

2. To disable the amplitude-modulation input, send this command:

   :AM:STATe OFF

   or

   :AM:STATe 0

3. To query the signal generator for the current amplitude-modulation state, send this command:

   :AM:STATe?

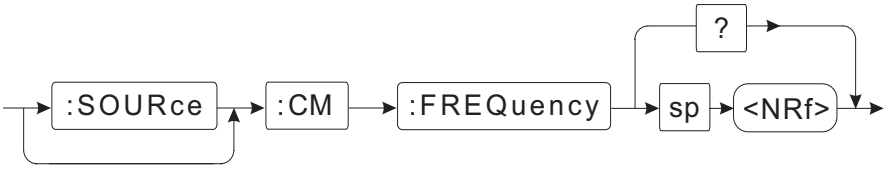
After the signal generator receives the query command, a read command returns the state of amplitude modulation in <bNR1> format. For example, if the amplitude modulation was previously enabled, and the signal generator is queried for the current state, it returns **1**, indicating that amplitude modulation is enabled (ON).

Related Commands:

:SOURce:AM:DEPTh
:SOURce:AM:DEPTh:STEP:INCRement

SOURce:CM:FREQuency
(Option 02 or Option 05 not installed)

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



The signal generator has an IF input that is 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 signal generators in Appendix C. The microwave filter is a YIG-tuned bandpass 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 signal generator.

To output the proper frequency and to correctly set the YIG-tuned bandpass filter, the center frequency of the externally supplied IF must be entered into the generator. The **:SOURce :CM:FREQuency** command accomplishes this. The generator adds the supplied IF to a synthesized carrier and outputs the sum. For example, if the generator is set to 6.0 GHz, and a 700 MHz IF is supplied, the generator synthesizes a 5.3 GHz signal and sets the bandpass filter to 6.0 GHz.

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

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:CM:FREQuency 500 MHZ

2. To query the signal generator for the current frequency expected at the IF input, send this command:

   :SOURce:CM:FREQuency?

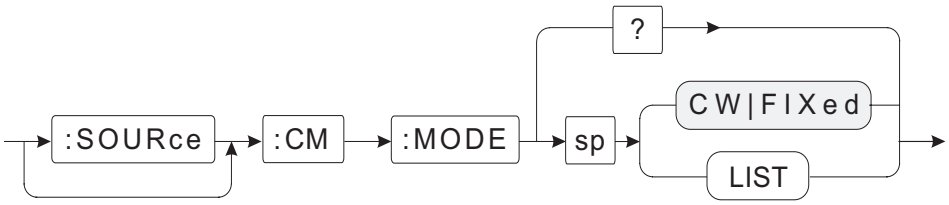
After the signal generator receives the query command, a read command returns the expected center frequency of the signal applied to the IF input in <NR1> format. For example, if the complex modulation frequency is set to 500 MHz, and the signal generator is queried for the current expected center frequency, it returns **500000000**, indicating that the complex modulation frequency is set to 500 MHz.

Related Command: **:SOURce:CM:STATe**

SOURce:CM:MODE

(Option 02 or Option 05 not installed)

Default: CW|FIXed



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

The **:SOURce:LIST** commands are included to provide compatibility with SCPI-defined syntax and are fully functional, but they are 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**, **DM**, **FM**, **FREQ**, **POWER**, and **PULM** (pulse modulation).

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:CM:MODE LIST

2. To query the signal generator for the current mode setting, send this command:

   :SOURce:CM:MODE?

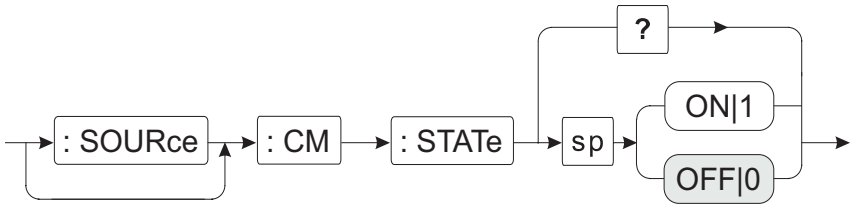
   After the signal generator receives the query command, a read command returns the current modesetting in <char> format. For example, if the **CM** mode was previously set to the **LIST** mode, and the signal generator is queried for the current mode, it returns **LIST**, indicating that the state of 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 turns 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 **:SOURce:CM:FREQuency** command. If it is not, the signal generator uses the current setting of the complex modulation frequency for adjusting the center frequency of the bandpass filter.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To enable complex modulation, send this command:

   :SOURce:CM:STATe ON

   or

   :SOURce:CM:STATe 1

2. To disable complex modulation, send this command:

   :CM:STATe OFF

   or

   :CM:STATe 0

3. To query the signal generator for the current state of complex modulation, send this command:

   :SOURce:CM:STATe?

   After the signal generator receives the query command, a read command returns the current state of complex modulation in <bNR1> format. For example, if complex modulation is enabled, and the signal generator is queried for the current state, the read returns **1**, indicating complex modulation is enabled.

NOTE

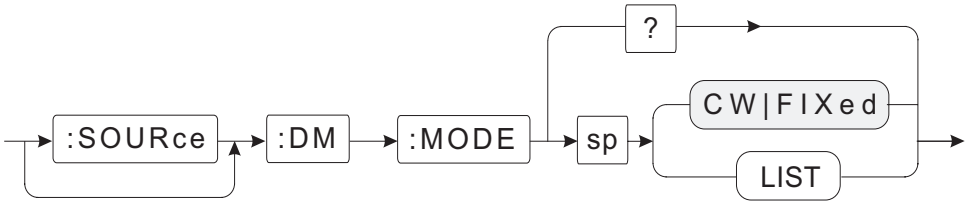
When operating the 114XB in the Complex Modulation mode, especially if the modulating signal has a varying signal level (amplitude) or low frequency amplitude modulation component, it is necessary to disable the Automatic Level Control (ALC) system and control the output power using the DAC command. Otherwise, the ALC will counter the amplitude component of the modulated signal. (See Appendix E.)

Related Commands:

:SOURce**:CM:FREQ**uency
:SOURce**:POW**er**:ALC:STAT**e
:SOURce**:POW**er**:DAC**

SOURce:DM:MODE
(Option 05 installed)

Default: CW|FIXed



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

The **:SOURce:LIST** commands are included to provide compatibility with SCPI-defined syntax and are fully functional, but they are not typically used. The command **:SOURce:LIST:PVECtor:MODE LIST** is normally used for coupling to the **LIST** mode, because it couples **DM** as well as **FM**, **AM**, **FREQ**, **POWER**, and **PULM** (pulse modulation).

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To couple digital (I/Q) modulation to the **LIST** mode, send this command:

   :SOURce:DM:MODE LIST

2. To query the signal generator for the current mode setting, send this command:

   :SOURce:DM:MODE?

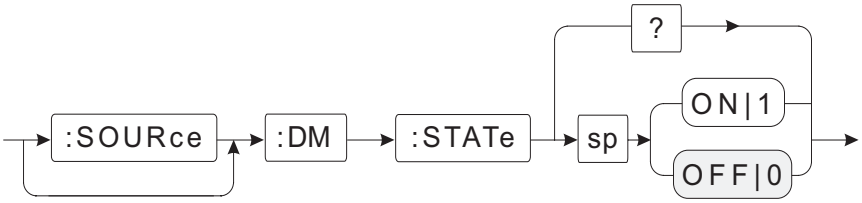
After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if the **DM** mode was previously set to the **LIST** mode, and the signal generator is queried for the current mode, the read operation returns **LIST**, indicating that the state of digital modulation is being controlled by the triggered-list function.

Related commands:

:SOURce:DM:STATe
:SOURce:LIST:PVECtor:MODE LIST

:SOURce:DM:STATe
(Option 05 installed)

Default: OFF



This command turns the digital (I/Q) modulation input on and off.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To enable digital modulation , send this command:

   :SOURce:DM:STATe ON

   or

   :SOURce:DM:STATe 1

2. To disable digital modulation, send this command:

   :DM:STATe OFF

   or

   :DM:STATe 0

3. To query the signal generator for the current state of digital modulation, send this command:

   :SOURce:DM:STATe?

   After the syntheszer receives the query command, a read command returns the current state of digital modulation in <bNR1> format. For example, if digital modulation is enabled, and the signal generator is queried for the current state, the read operation returns **1**, indicating that digital modulation is enabled.

NOTE

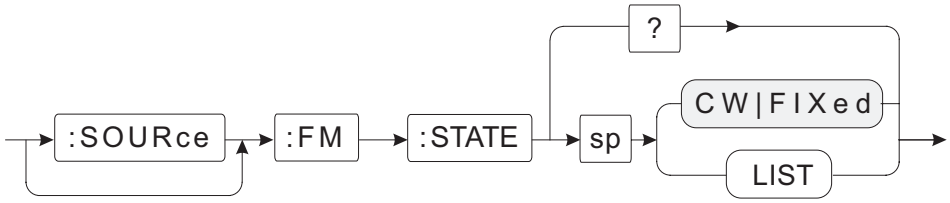
When operating the 114XB in Digital Modulation mode, especially at low modulating frequencies, or if the modulation has a DC component, it is necessary to disable the Automatic Level Control (ALC) system and control the output power using the DAC command. If this is not done, the ALC will counter the amplitude component of the modulated signal. (See Appendix E.)

Related Commands: **:SOURce:POWer:ALC:STATe** **:SOURce:POWer:DAC**

SOURce:FM:MODE

(Option 02 or Option 05 installed, or SY-12 version)

Default: CW|FIXed



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

The **:SOURce:LIST** commands are included to provide compatibility with SCPI-defined syntax, and they are fully functional, but they are typically not used. The **:SOURce:LIST:PVECtor:MODE LIST** command is normally used for coupling to **LIST** mode, because it couples FM as well as: DM, AM, FREQ, POWer, and PULM (pulse modulation).

> **NOTE**
>
> Use of the **:SOUR**ce parameter is optional.

Examples:

1. To couple frequency modulation to **LIST** mode, send this command:

   :SOURce:FM:MODE LIST

2. To query the signal generator for the current mode setting, send this command:

   :SOURce:FM:MODE?

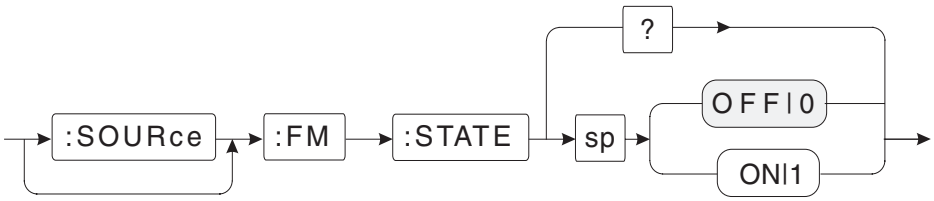
After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if the **FM** mode was previously set to **LIST** mode, and the signal generator is queried for the current mode, it returns **LIST**, indicating that the state of frequency modulation is currently being controlled by the triggered-list function.

Related Commands:

:SOURce:FM:STATe
:SOURce:LIST:PVECtor:MODE LIST

SOURce:FM:STATe
(Option 02 or Option 05 installed, or SY-12 version)

Default: OFF



This command turns the frequency-modulation input on or off.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To enable the frequency modulation input, send this command:

   :SOURce:FM:STATe ON

   or

   :SOURce:FM:STATe 1

2. To disable the frequency modulation input, send this command:

   :FM:STATe OFF

   or

   :FM:STATe 0

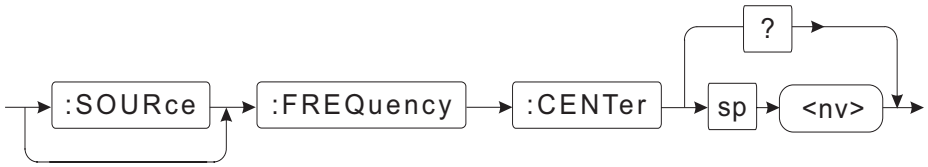
3. To query the signal generator for the current frequency-modulation state, send this command:

   :SOURCe:FM:STATe?

After the signal generator receives the query command, a read command returns the state of frequency modulation in <bNR1> format. For example, if the frequency modulation input was previously enabled, and the signal generator is queried for the current state, it returns **1**, indicating that frequency modulation is enabled (ON).

SOURce:FREQuency:CENTer

Default:(MIN+MAX)/2
1140B: 10.005 GHz
1141B:11 GHz



This command sets the center frequency of the signal generator when it is operated in the **SWEEP** mode. The command's default value is the midpoint of the generator's total range. When the generator is commanded to a new center frequency, it calculates new start and stop frequencies based on the frequency span currently set. If a computed start or stop frequency falls outside the generator's frequency range, an error message is generated. It is, therefore, good practice to send a new frequency span along with the new center frequency, choosing a span that keeps the range swept within the capabilities of the generator.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:FREQ*uency*:CENTer 7 GHz;:SOURce:FREQuency:SPAN 6 GHz

   When the signal generator receives the new center frequency and frequency span, it recomputes the corresponding start and stop frequencies. For this example, the new start frequency is 4 GHz and the new stop frequency is 10 GHz.

2. To query the signal generator for the current center frequency, send this command:

   :SOURce:FREQuency:CENT?

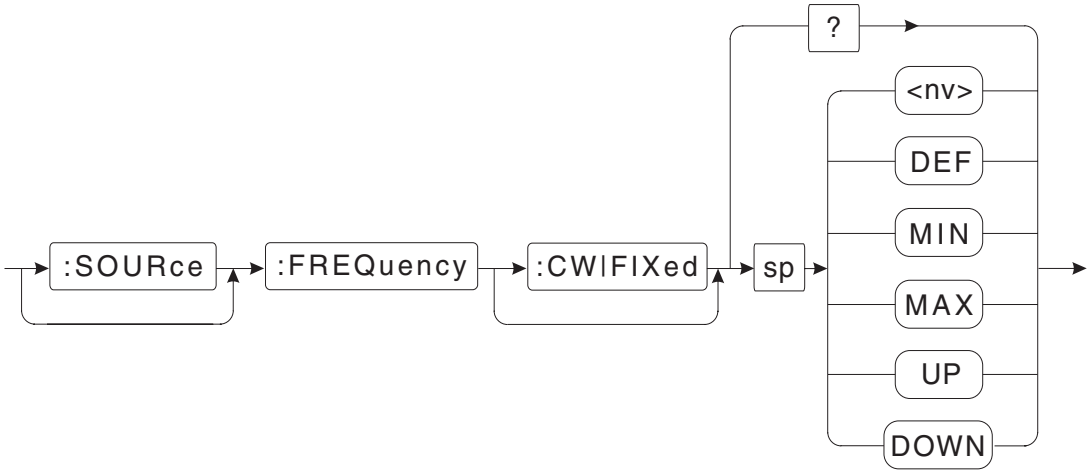
   When the signal generator receives the query command, a read command returns the current mode setting in <NR1> format. For example, if the 1141B is at the default center frequency, and the signal generator is queried for the current value, it returns **11000000000**, indicating that the current center frequency is 11.0 GHz.

Related Commands:

:SOURce:FREQuency:SPAN
:SOURce:FREQuency:STARt
:SOURce:FREQuency:STOP
:SOURce:SWEep1:COUNt
:SOURce:SWEep1:DWELL
:SOURce:SWEep1:STARt
:SOURce:SWEep1:STEP:FREQuency

SOURce:FREQuency:CW|FIXed

Default:(MIN + MAX)/2
1140B: 10.005 GHz
1141B:11 GHz
Resolution: 1 Hz
Range:
1140B: 0.001 GHz to 20 GHz
1141B:2 GHz to 20 GHz



This command sets the generator's output frequency when it is operated in the **CW|FIXed** mode. Frequency resolution is 1 Hz. The arguments **MIN** and **MAX** set the output frequency to the limits of the generator's range. The argument **DEF** set it to the midpoint of its range. The arguments **UP** and **DOWN** increment or decrement the frequency by the amount set by the **:SOURce:FREQuency:CW:STEP:INCRement** command.

NOTE

Use of the **:SOURce**, **:CW**, and **:FIXed** parameters is optional.

Examples:

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

   :SOURce:FREQuency:CW 7E9

   or

   :FREQuency:FIXed 7 GHZ

2. To query the signal generator for the current frequency setting, send this command:

   :SOURce**:FREQ**uency**:CW?**

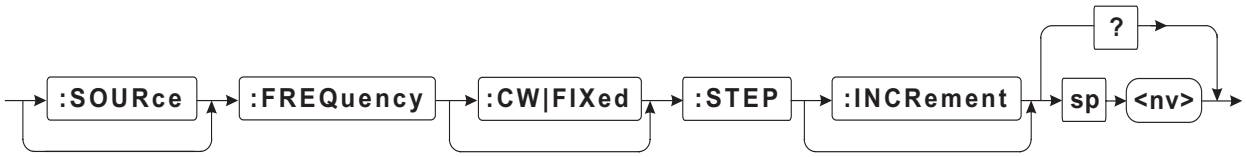
   After the signal generator receives the query command, a read command returns the current frequency setting in <NR1> format. For example, if the frequency is set to 11 GHz, and the signal generator is queried for the current frequency, it returns **11000000000**, indicating that the frequency is set to 11 GHz.

Related Commands:

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

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 signal generator when it is in the **CW|FIXed** mode. The incremental step size can be either positive or negative.

NOTE

Use of the **:SOURce**, **:CW** or **:FIXed**, and **:INCRement** parameters is optional.

Examples:

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

   :SOURce:FREQuency:CW:STEP:INCRement 10 KHZ

   or

   :FREQuency:FIXed:STEP 10 KHZ

2. To query the signal generator for the current frequency step size, send this command:

   :SOURce:FREQuency:CW:STEP:INCRement?

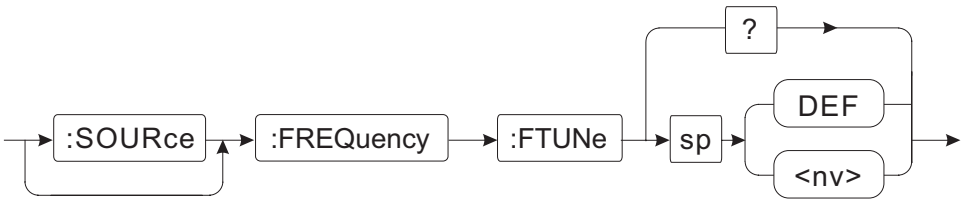
After the signal generator receives the query command, a read command returns the current frequency step setting in <NR1> format. For example, if the frequency step is set to 1 kHz, and the signal generator is queried for the current frequency step, it returns **1000**, indicating that the frequency step is set to 1000 Hz (1 kHz).

Related Command:

:SOURce:FREQuency:CW|FIXed

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 signal generator. When output blanking is off, this command can change the frequency over a limited bandwidth without the large frequency transient and power loss that normally occurs during changes in output frequency.

Commanding the signal generator to a new frequency, either directly or via triggered list, resets the fine-tune frequency to zero.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To fine-tune the signal generator 500 kHz above the current CW frequency, send this command:

   :SOURce:FREQuency:FTUNe 500 KHZ

   When the signal generator receives the fine tune command, it commands the DDS to a new frequency, causing the output frequency from the signal generator to increase by 500 kHz.

2. To query the signal generator for the current center frequency, send this command:

   :SOURce:FREQuency:FTUNEe?

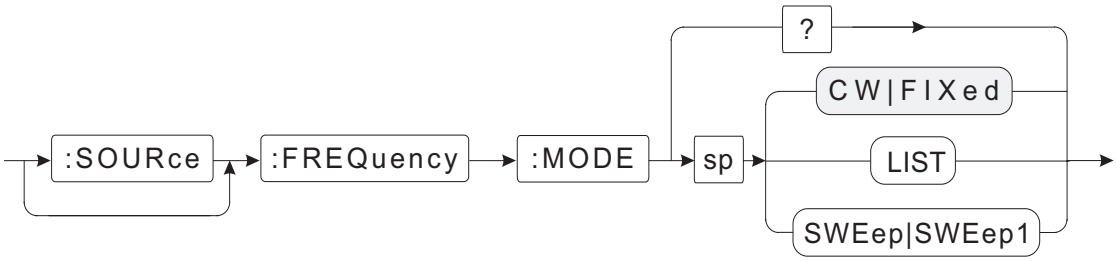
After the signal generator receives the query command, a read command returns the current mode setting in <NR1> format. For example, if the fine-tune command was used to increase the output frequency from the signal generator by 500 kHz, and the signal generator is queried for the current value, it returns **500000**, indicating that current fine-tune frequency is 500 kHz.

Related Command:

:SOURce:FREQuency:CW

SOURce:**FREQ**uency:**MODE**

Default: CW|FIXed



This command couples the signal generator's frequency control to the **FIX**ed, **LIST**, or **SWE**ep**1** mode. It is normally used to couple control of the signal generator to the **SWE**ep**1** mode. It can also be used to couple control of the source frequency to the **LIST** mode, but it is not typically used for this function. The **:SOUR**ce**:LIST:PVEC**tor**:MODE LIST** command is a better choice for coupling frequency control to the **LIST** mode, because it couples frequency as well as: AM, CM, DM, FM, POWER, and PULM (pulse modulation).

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

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

   :SOURce**:FREQ**uency**:MODE SWE**ep**1**

2. To query the signal generator for the current mode setting, send this command:

   :SOURce**:FREQ**uency**:MODE?**

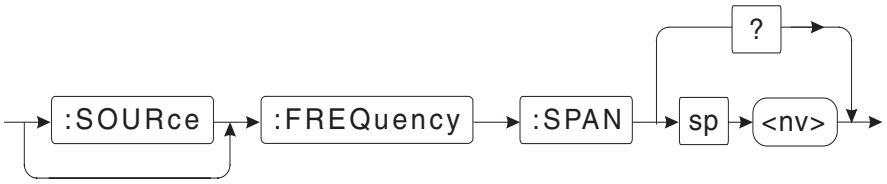
After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if the frequency mode was previously set to **SWE**ep**1**, and the signal generator is queried for the current mode, it returns **SWE** indicating that the output frequency from the signal generator is currently controlled by **SWE**ep**1**.

Related Command:

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

SOURce:FREQuency:SPAN

Default:
1140B: 19.999 GHz
1141B:18 GHz



This command sets the frequency span for the signal generator. The frequency span is centered around the current center frequency. It defaults to the maximum range of the signal generator. Whenever a new frequency span is received, the signal generator 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 is 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 1141B is set to 6 GHz, the center frequency can be any value between 5 GHz and 17 GHz. In mathematical terms, the allowable range of the center frequency can be calculated using the following formula:

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

For the 1141B, the formula can be rewritten as:

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

For the 1140B, the formula can be rewritten as:

$$Fcenter = 10.0005\ GHz \pm \frac{19.999\ GHz - Span}{2}$$

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To set the frequency span, send the 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 this command:

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

   Upon receiving the new center frequency and span, the signal generator recomputes the corresponding start and stop frequency. For this example, the new start frequency is 4 GHz and the new stop frequency is 10 GHz.

2. To query the signal generator for the current span, send this command:

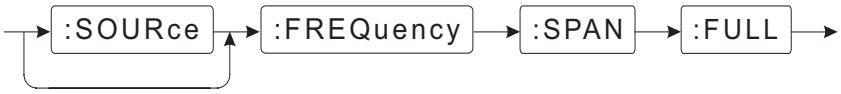
   :SOURce:FREQuency:SPAN?

   After the signal generator receives the query command, a read command returns the current mode setting in <NR1> format. For example, if the 1142A is at the default frequency span and the signal generator is queried for the current value, it returns **8400000000**, indicating that the current frequency span is 8.4 GHz.

Related Commands:

:SOURce:FREQuency:CENTer
:SOURce:FREQuency:STARt
:SOURce:FREQuency:STOP

SOURce:**FREQ**uency:**SPAN:FULL**



This command sets the start frequency, stop frequency, center frequency, and frequency span back to the default values listed below in Table 4-1. Although this command changes signal-generator parameters, it is an event rather than a state, so it has no associated query.

Table 4-1. **SOUR**ce:**FREQ**uency:**SPAN:FULL** *Default Values*

	1140B	**1141B**
Start	0.001 GHz	2 GHz
Stop	20 GHz	20 GHz
Center	10.0005 GHz	11 GHz
Span	19.999 GHz	18 GHz

NOTE

Use of the **:SOUR**ce parameter is optional.

Example:

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

:SOURce:**FREQ**uency:**SPAN:FULL**

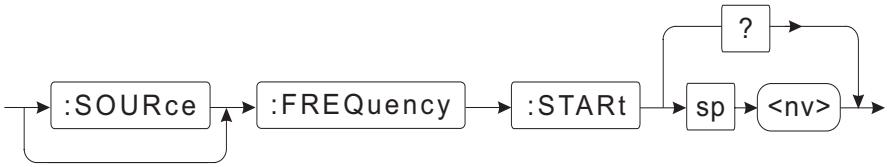
Related Commands:

:SOURce:**FREQ**uency:**CENTer**
:SOURce:**FREQ**uency:**SPAN**
:SOURce:**FREQ**uency:**STAR**t
:SOURce:**FREQ**uency:**STOP**
:SOURce:**SWE**ep1:**COUN**t
:SOURce:**SWE**ep1:**DWEL**L
:SOURce:**SWE**ep1:**STAR**t
:SOURce:**SWE**ep1:**STEP:FREQ**uency

SOURce:**FREQ**uency:**STAR**t

Default:

1140B: 0.001 GHz
1141B: 2.0 GHz



This command sets the start frequency for a frequency sweep of the signal generator. It defaults to the generator's minimum frequency. Whenever a new start frequency is received, the signal generator calculates a new center frequency and frequency span based on the current stop frequency. These values are returned in response to a **:SOURce:FREQ:CENT?** or **:SOURce:FREQ:SPAN?** command.

Normally, a start frequency is less than a stop frequency, but it is valid to set a start frequency higher than the stop frequency. If the start frequency is higher than the stop frequency, the computed frequency span is negative, and the generator sweeps from the high to the low frequency.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

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

   :SOURce:**FREQ**uency:**STAR**t **6 GHz**

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**uency:**STAR**t **6 GHz;:SOUR**ce:**FREQ**uency:**STOP 8 GHz**

3. To query the signal generator for the current start frequency, send this command:

   :SOURce:**FREQ**uency:**STAR**t**?**

   After the signal generator receives the query command, a read command returns the current start frequency in <NR1> format. For example, if the 1141B is at the default start frequency, and the signal generator is queried for the current value, it returns **4000000000**, indicating that the current start frequency is 4 GHz.

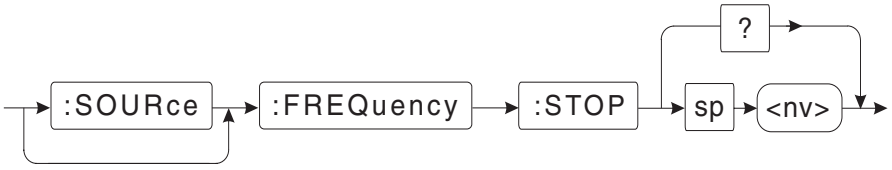
Related Commands:

:SOURce:**FREQ**uency:**CENTer**
:SOURce:**FREQ**uency:**SPAN**
:SOURce:**FREQ**uency:**STOP**
:SOURce:**SWE**ep1:**COUN**t
:SOURce:**SWE**ep1:**DWEL**L
:SOURce:**SWE**ep1:**STAR**t
:SOURce:**SWE**ep1:**STEP:FREQ**uency

SOURce:FREQuency:STOP

Default:

1140B:20 GHz
1141B:20 GHz



This command sets the stop frequency for a frequency sweep of the signal generator. It defaults to the generator's maximum frequency. Whenever the generator receives a new stop frequency, it calculates a new center frequency and frequency span based on the current start frequency. These values are returned in response to a **:SOURce:FREQ:CENT?** or **:SOURce:FREQ:SPAN?** command.

Normally, a stop frequency is greater than a start frequency, but it is valid to set a stop frequency lower than the start frequency. If the stop frequency is lower than the start frequency, the computed frequency span is negative, and the gernerator sweeps from the high to the low frequency.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:FREQuency:STOP 10 GHz

   Upon receiving the new stop frequency, the signal generator changes both the center frequency and the frequency span based on the new stop frequency and the current start frequency.

2. Both the new start and stop frequencies can be sent on the same line, as shown below.

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

3. To query the signal generator for the current stop frequency, send this command:

   :SOURce:FREQuency:STOP?

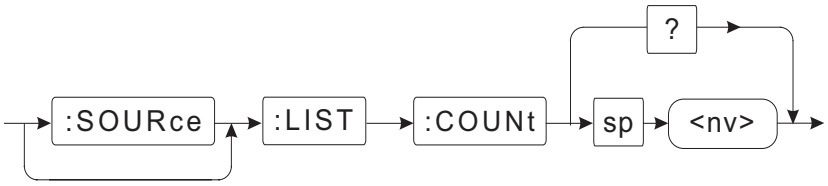
   After the signal generator receives the query command, a read command returns the current stop frequency in <NR1> format. For example, if the 1141B is at the default stop frequency, and the signal generator is queried for the current value, it returns **20000000000,** indicating that the current stop frequency is 20.0 GHz.

Related Commands:

:SOURce:FREQuency:SPAN
:SOURce:FREQuency:STARt
:SOURce:FREQuency:CENTer
:SOURce:SWEep1:COUNt
:SOURce:SWEep1:DWELL
:SOURce:SWEep1:STARt
:SOURce:SWEep1:STEP:FREQuency

SOURce:LIST:COUNt

Default:1
Range:1 to 32000



This command sets the number of times the signal generator cycles through a list for each start event.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To command the signal generator to cycle twice through the triggered list after detecting the trigger, send this command:

   :SOURce:LIST:COUNt 2
2. To query the signal generator for the current list count, send this command:

   :SOURce:LIST:COUNt?

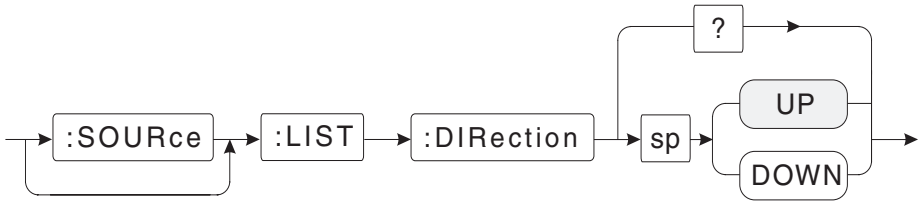
   After the signal generator receives the query command, a read command returns the current list count in <NR1> format. For example, if the list count was previously set to 10, and the signal generator is queried for the current list count value, it returns **10**, indicating that the signal generator 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 signal generator sequences through the list. Normally, the signal generator steps from the lowest point vector (PVEC1) toward the highest point vector.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

1. To command the signal generator to sequence through the list beginning at the highest point vector, send this command:

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

2. To query the signal generator for the current list direction, send this command:

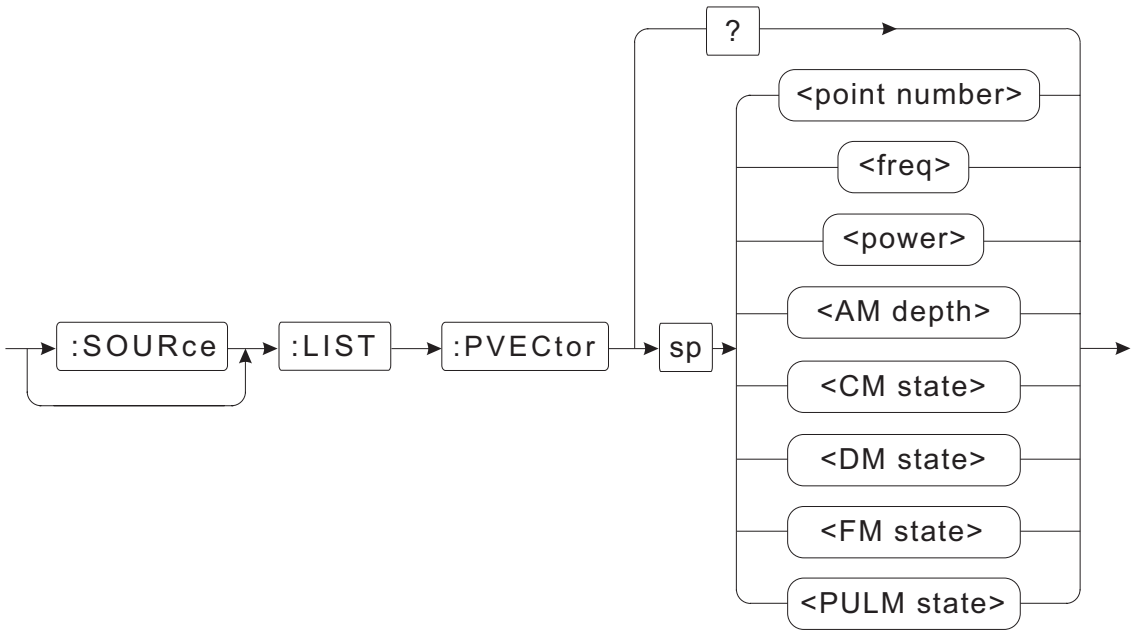
   :SOURce**:LIST:DIR**ection**?**

   After the signal generator receives the query command, a read command returns the current list direction. For example, if the list direction is set to the default direction, and the signal generator is queried for the current list direction, it returns **UP**, indicating that the signal generator 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 builds 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, DM state, FM state, and PULM (pulse modulation) state. Once the list is constructed, the signal generator 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 **PVECtor 1**. The last possible point vector is called **PVECtor 201**.

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

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

For example, to program point 1 of the list to 5 GHz at 0 dBm with AM depth at 0, and all other modulation turned off, send the following command:

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

A missing data item in an individual row is replaced with the current instrument setting. The previous command sets 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, OFF, OFF

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

:LIST:PVEC 2, 5 GHz, 0 dBm, 0, OFF, OFF, 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 is filled in as shown below.

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

When specifying a new list table, identical **LIST:PVEC**tors can be created without specifying all of them. Using a point vector number higher than any previously received causes the signal generator to fill all point 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 signal generator:

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

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

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

CONTROLLING LIST SEQUENCING

In 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 signal generator remains at each point vector in the list.

:LIST:STARt

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

:LIST:DIRection

In trigger-list mode, a trigger is required to cause the signal generator 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 these commands:

:LIST:PVECtor**:DWEL**L

:LIST:PVECtor**:DWEL**L**:LINK DWEL**L

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

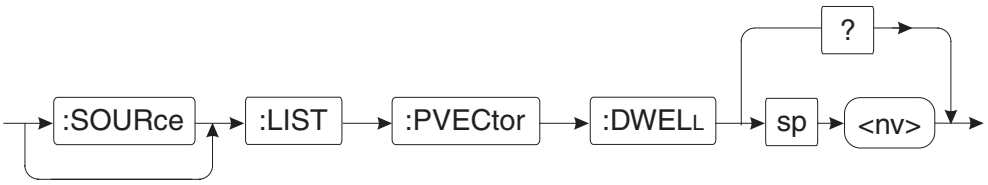
:LIST:COUNt

After a hard reset or **MEM**ory**:DELete LIST** command, the first **LIST:PVEC**tor command allocates 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 signal-generator hardware. The associated point vector commands are not saved, and therefore, point vectors cannot be read back from the signal generator. 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 signal generator.

For complete programming examples of the triggered-list feature, see Section 5.

SOURce:LIST:PVECtor:DWELL

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



This command controls the amount of time the signal generator remains at each point in a trigger list. This command is used to set the dwell time. Dwell time is set in units of 1 microsecond in the range of 5 milliseconds to 32 seconds and can be entered as SEC (seconds) or USEC (microseconds). If no label is entered, the entry is assumed to be in seconds. The signal generator can also be programmed to dwell at each point until an external event is detected using the **:SOURce:LIST:PVECtor:DWELL:LINK** command.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:LIST:PVECtor:DWELL 5

2. To query the signal generator for the dwell time, send this command:

   :SOURce:LIST:PVECtor:DWELL?

   After the signal generator receives the query command, a read command returns the current dwell time in seconds. For example, if the signal generator was just turned on and it is queried for the current (default) dwell time, it returns **1.000** indicating a dwell time of 1 second.

NOTE

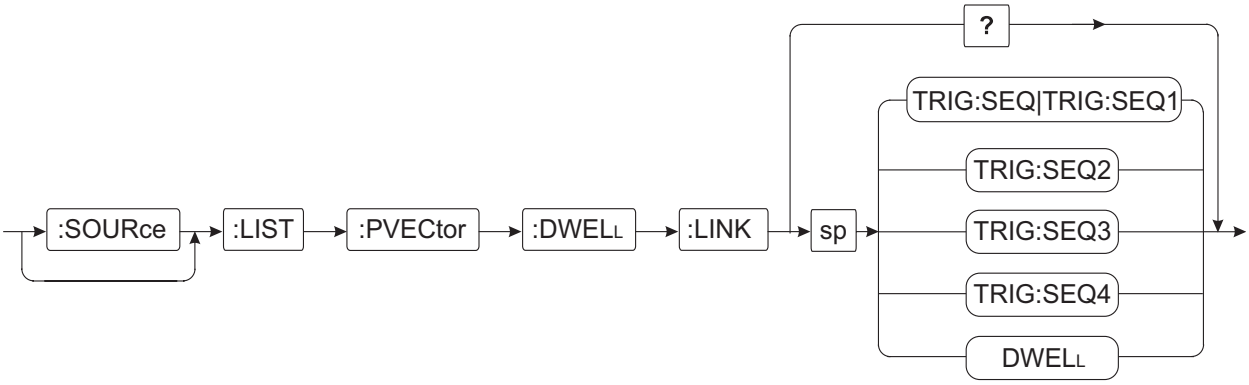
The switching overhead is 35 ms, so the minimum repetition rate is 40 ms (<dwell time> + 35ms).

Related Command:

:SOURce:LIST:PVECtor:DWELL:LINK

SOURce:LIST:PVECtor:DWELL:LINK

Default: DWELL



This command links dwell to the event that triggers the signal generator to move to the next point in a triggered list. The event 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.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To command the signal generator to step between points of a triggered list based on trigger sequence2 (**TRG:SEQ2**), send this command:

   :SOURce:LIST:PVECtor:DWELL:LINK TRIG:SEQ2

2. To command the signal generator to step between points of a triggered list every 5 seconds, send these commands:

   :SOURce:LIST:PVECtor:DWELL 5

   :SOURce:LIST:PVECtor:DWELL:LINK DWELL

3. To query the signal generator for the current link to dwell, send this command:

   :SOURce:LIST:PVECtor:DWELL:LINK?

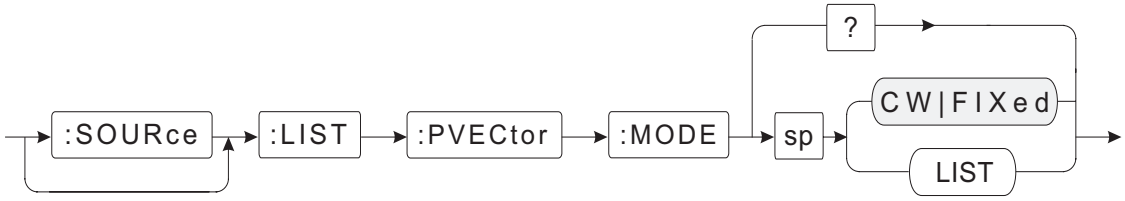
   After the signal generator receives the query command, a read command returns the current link to dwell. For example, if the signal generator was just turned on, and it is queried for the current (default) dwell link, it returns **DWEL** indicating dwell is linked to the dwell timer.

Related Commands:

:SOURce:LIST:PVECtor:DWELL
:SOURce:TRIG:SEQ<n>**:SOURce**

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

Default: CW|FIXed



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

NOTE

Use of the **:SOUR**ce parameter is optional.

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 this command:

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

2. To query the signal generator for the current mode, send this command:

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

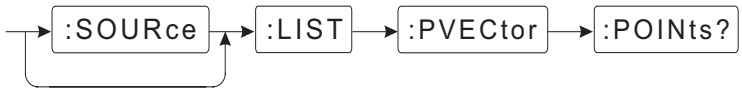
   After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if the mode was previously set to **LIST**, and the signal generator is queried for the current mode, it returns **LIST**, indicating that control of output frequency and power, along with the state of amplitude modulation (AM), pulse modulation (PULM), frequency modulation (FM), digital modulation (DM), and complex modulation (CM) is coupled to triggered list.

Related Commands:

:SOURce**:AM:MODE**
:SOURce**:CM:MODE**
:SOURce**:DM:MODE**
:SOURce**:FM:MODE**
:SOURce**:FREQ**uency**:MODE**
:SOURce**:POW**er**:MODE**
:SOURce**:PULM:MODE**

SOURce**:LIST:PVEC**tor**:POIN**ts**?**

Default: 0
Range: 0 to 201



This is a query-only command that returns the current number of trigger-list point vectors. For example, to query the signal generator for the current number of point vectors, send this command:

:SOURce**:LIST:PVEC**tor**:POIN**ts**?**

or send the shortened version:

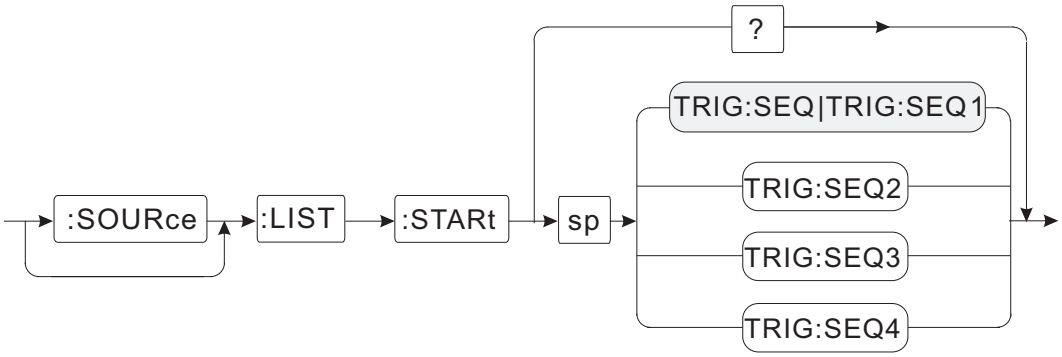
:LIST:PVEC:POINts**?**

> **NOTE**
>
> Use of the **:SOUR**ce command is optional.

After the signal generator receives the query command, a read command returns the current number of trigger-list point vectors in <NR1> format. For example, if the signal generator was just turned on, and it is queried for the number of point vectors, it returns **0**, indicating that no point vectors have been created.

SOURce:LIST:STARt

Default: TRIG:SEQ1



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

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To command the signal generator to start sequencing through the trigger list upon receiving a trigger from trigger sequence 1, send this command:

   :SOURce:LIST:STARt TRIG:SEQ1

   or the shortened version:

   :LIST:STAR TRIG:SEQ

   In the shortened version, the number 1 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 signal generator for the current trigger sequence, send this command:

   :SOURce:LIST:STARt?

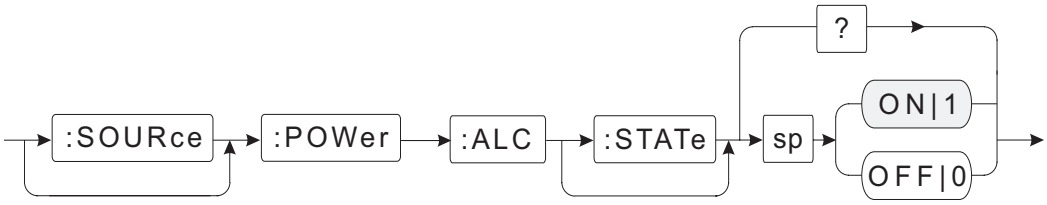
   After the signal generator receives the query command, a read command returns the current trigger sequence in <char> format. For example, if triggered-list is using the default trigger sequence, a query returns **TRIG:SEQ**, indicating that trigger sequence 1 will start the signal generator sequencing through the triggered list.

Related Commands:

:SOURce:LIST:COUNt
:SOURce:LIST:DIRection

SOURce:POWer:ALC:STATe

Default: ON



This command turns the automatic leveling control (ALC) on the signal generator on and off. It is normally 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 the Caution below).

Since the ALC controls the output-power level, turning it off disables normal level control and causes bit 0 in the Questionable Power event register to be set to **1**.

NOTE

Use of the **:SOURce** and **:STATe** parameters is optional.

Examples:

1. To turn off the ALC, send this command:

   :SOURce:POWer:ALC:STATe OFF

   or the shortened version:

   :POWer:ALC 0

2. To query the signal generator for the state of the ALC, send this command:

   :SOURce:POWer:ALC:STATe?

   After the signal generator receives the query command, a read command returns the ALC state in <bNR1> format. For example, if the ALC was previously turned off, and the signal generator is queried for the current state, it returns **0**, indicating that the ALC is turned off.

CAUTION

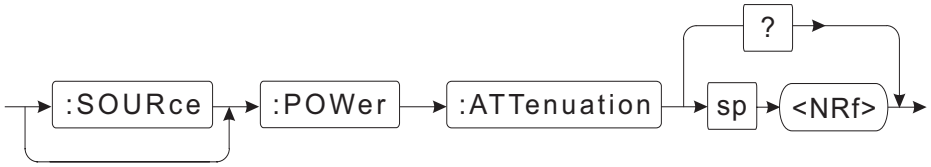
When the ALC in the signal generator is off, the output power at the front-panel connector can exceed the maximum specified output. Generally, turning off the ALC causes the output power to drop, but if this command is used in conjunction with the :POWer:DAC command, the power at the front panel could increase up to +20 dBm.

Related Commands:

:SOURce:POWer:ATTenuation
:SOURce:POWer:ATTenuation:AUTO

SOURce:POWer:ATTenuation

Default: 0 dB
Resolution: 10 dB
Range:
1140B:0 to 90 dB
1141B:0 to 90 dB



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

NOTE

Use of the **:SOURce** parameter and the **dB** argument is optional.

Examples:

1. To set RF output attenuation to 10 dB, send this command:

   :SOURce:POWer:ATTenuation 10
2. To set RF output attenuation to 20 dB, send this command:

   :POWer:ATTenuation 20

3. To query the signal generator for the status of RF output, send this command:

   :POWer:ATTenuation?

   After the signal generator receives the query command, a read command returns the current attenuation setting in <NR2> format. For example, if the attenuation was previously set to 10 dB, and the signal generator is queried for the current attenuation setting, it returns **10.0**, indicating that the attenuation is set to 10 dB.

NOTE

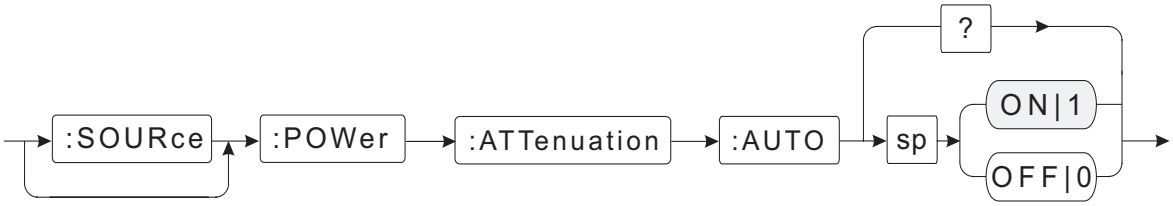
Use of this command automatically decouples the attenuator from the output-power level.

Related Commands:

:SOURce:POWer:ALC:STATe
:SOURce:POWer:ATTenuation:AUTO
:SOURce:POWer:LEVel

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

Default: ON



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

> **NOTE**
>
> Use of the **:SOUR**ce parameter is optional.

Examples:

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

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

   or

   :POWer:**ATT**enuation:**AUTO 1**

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

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

   or

   :POWer:**ATT**enuation:**AUTO 0**

3. To query the signal generator for the current state of the attenuator, send this command:

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

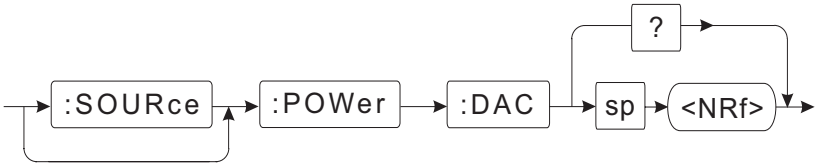
   After the signal generator receives the query command, a read command returns the current attenuation setting in <NR2> format. For example, if attenuation coupling was previously set to default and the signal generator is queried for the current attenuation coupling, it returns **1**, indicating that the ALC and step attenuator are coupled.

Related Commands:

:OUTPut:**STAT**e
:SOURce:**POW**er:**LEV**el:**IMMediate:AMPL**itude

SOURce:POW:DAC

Range: 0 to 100%
Resolution: 0.1%



This command sets the RF power control DAC directly in percent-full-scale. It is primarily used for troubleshooting the signal generator and in some special applications. While it does provide control of the output power, the **:SOUR:POW:LEV:IMM:AMP** command is normally 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 the Caution on below).

NOTE

Use of the **:SOUR**ce parameter and the **%** argument is optional.

Examples:

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

   :SOURce**:POW**er**:DAC 60%**

2. To query the signal generator for the current setting of the RF power control DAC, send this command:

   :POWer**:DAC?**

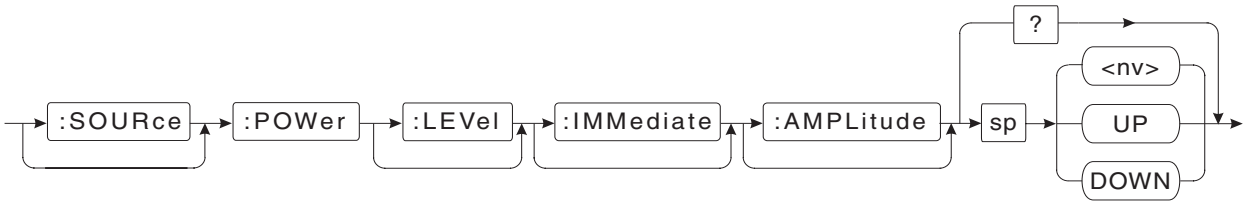
   After the signal generator receives the query command, a read command returns the current setting of the RF power-control DAC in <NR2> format. For example, if the RF power-control DAC was previously set to 60%, and the signal generator is queried for the current setting, it returns **60**, indicating that the RF power-control DAC is at 60 percent of full scale.

CAUTION

When the ALC in the signal generator is turned off, the output power at the front-panel connector can exceed the maximum specified output. Generally,turning off the ALC causes 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:
1140B:
Coupled see 1140B power-range specification
Uncoupled ≤18 GHz: +10 to –3 dBm
>18 GHz: +5 to –5 dBm
1141B:
Coupled see 1141B power-range specification
Uncoupled +10 to –3 dBm



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

NOTE

Use of the **:SOURce**, **:LEVel**, **:IMMediate**, and **:AMPLitude** parameters and the **dB** argument is optional.

Examples:

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

   :SOURce:POWer:LEVel:IMMediate:AMPLitude 5

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

   :POW 10 DBM

3. To query the signal generator for the current setting of the output power, send this command:

   :SOURce:POWer:LEVel:IMMediate:AMPLitude?

   After the signal generator receives the query command, a read command returns the current setting of the output power in <NR2> format. For example, if the output power was previously set to the default value, and the signal generator is queried for the current setting, it returns **0.0**, indicating that 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 signal generator. 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 of the **:POW**er**:LEVEL** command is then –3 to +10 dBm. The generator output power is then equal to that commanded by the **:POW:LEVEL** command minus the output attenuator setting commanded by **:POW**er**:ATT**enuation.

4. To step the power level of the signal generator by the current step size, send this command:

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

   or

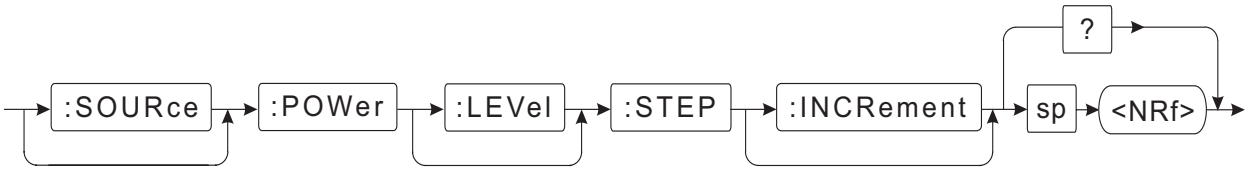
   :POWer **DOWN**

Related Commands:

:OUTPut**:STAT**e
:SOURce**:POW**er**:STEP:INCR**ement

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

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 signal generator.

NOTE

Use of the **:SOUR**ce, **:LEV**el, and **:INCR**ement parameters and the **dB** argument is optional.

Examples:

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

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

   or

   :POWer:**STEP 10 dB**

2. To query the signal generator for the current power level step size, send this command:

   :POWer:**STEP?**

   After the signal generator receives the query command, a read command returns the current power-level step size in <NR2> format. For example, if the power-level step size was previously set to 2 dB, and the signal generator is queried for the current setting, it returns **2.0**, indicating that the step size for power sweep is 2 dB.

Related Commands:

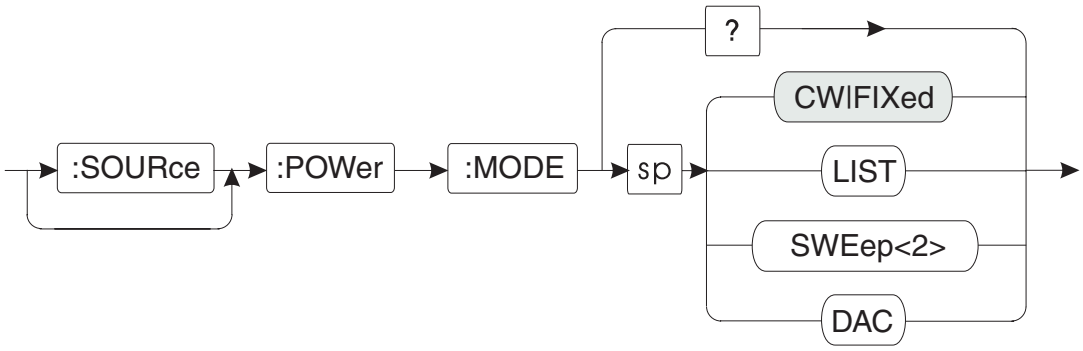
:OUTPut:**STAT**e
:SOURce:**POW**er:**LEV**el:**IMM**ediate:**AMPL**itude

NOTE

When the attenuator is coupled to output level, setting output power to minimum generates an out-of-range error.

SOURce:POWer:MODE

Default: CW|FIXed



This command couples control of the output-signal power to either the **FIX**ed or **LIST** mode. It is normally used to select either the **FIX**ed or **SWE**ep**2** mode, which couples output power to control the family of source-power commands.

The **SOUR**ce**:POW**er**:MODE** command can be used to couple power control to Triggered List, but it is not typically used for this purpose. The command :**SOUR**ce**:LIST:PVEC**tor**:MODE LIST** is normally used for coupling to the **LIST** mode, because it couples POWer as well as AM, CM, DM, FM, FREQ, and PULM (pulse modulation).

The DAC mode is used for special cases where the user wants complete control over the power level. It is used in conjunction with the **SOUR**ce**:POW**er**:DAC** commands. When in DAC mode, only the **SOUR**ce**:POW**er**:DAC** commands can be used to update the power level. Using **POW**er**:LEV**el results in an error 221. When in DAC mode, changing the frequency does not result in application of new corrections for losses at the new frequency.

NOTE

While AM and CM can be used while in DAC mode, they must be set before entering DAC mode. The **ALC:STAT**e, **AM:STAT**e, **AM:DEPTH**, and **CM:STATe** commands are not available while in DAC mode. This mode is also incompatible with the power-sweep and list modes, but It can be used in frequency-sweep mode.

NOTE

When the frequency is changed while in DAC mode, the power/frequency validity is checked using the last value set by the **POWer:LEVel** command. For this reason, set a value for **POWer:LEVel** that is valid for all frequencies to be used before entering DAC mode. To exit DAC mode, select one of the other modes. The power level is reset to the last value of **POW**er**:LEV**el.

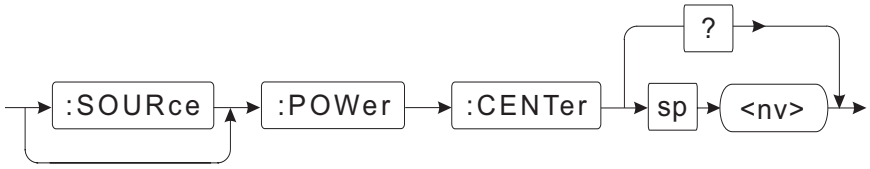
Related Commands:

:SOURce**:LIST:PVEC**tor**:MODE LIST**
:SOURce**:POW**er**:LEV**el**:IMM**ediate**:AMP**litude

SOURce:POWer:CENTer

Default:+5 dBm
Range:

Minimum generator power level plus ½ the currently selected power span to maximum generator power level minus ½ the currently selected power span



This command sets a reference-center-power for the signal generator that establishes a midpoint for power sweeps. It defaults to 5 dBm, which is the midpoint between the default start and the stop power for the signal generator. Whenever a new center power is received, the signal generator 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 is generated.

Examples:

1. To set a center power of –10 dBm, send this command:

   :SOURce:POWer:CENTer –10

   NOTE

   Use of the **:SOURce** parameter and the **dB** argument is optional.

2. If the power limits are less than ½ of the power span from the entered center power, an error message is generated. To prevent this from occurring, when sending a new center power, also send a power span. This prevents the calculation of start and stop powers that are 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 a power-span command chosen to prevent error messages from being generated, as shown below.

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

   Upon receiving the new center power and span, the signal generator recomputes the corresponding start and stop powers. For this example, the new start power is 10 dBm and the new stop power is 12 dBm.

3. To query the signal generator for the current center power, send this command:

   :SOURce:POWer:CENT?

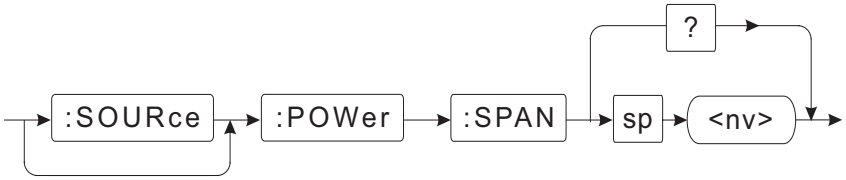
   After the signal generator receives the query command, a read command returns the current center power in <NR2> format. For example, if the signal generator is at the default center power, and it is queried for the current value, it returns **5.0**, indicating that the current center power is 5.0 dBm.

Related Commands:

:SOURce:POWer:SPAN **:SOURce:POWer:STARt** **:SOURce:POWer:STOP**

SOURce:POWer:SPAN

Default: 10 dB
Resolution: 0.1



This command sets a power span for the signal generator that establishes a range for power sweeps. It defaults to 10 dB, which is the difference between the default start and stop power levels for the signal generator.

When the signal generator is performing a power sweep, 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, select the center power and sweep span to avoid step-attenuator switching points.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:POWer:SPAN 2

   Whenever the signal generator receives a power-span command, it recomputes the start and stop powers levels based on the center power and the new power span. If these computed power levels are out of range, an error message is generated. To prevent this from occurring, a power span can be set along with the required center power. For example, to set a center power of 11 dBm combined with a power span of 2 dB, send this command:

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

   Upon receiving the new center power and power span, the signal generator recomputes the corresponding start and stop powers. For this example, the new start power is 10 dBm and the new stop power is 12 dBm.

2. To query the signal generator for the current power span, send this command:

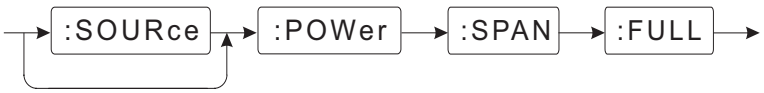
   :SOURce:POWer:SPAN?

After the signal generator receives the query command, a read command returns the current power span in <NR2> format. For example, if the signal generator is at the default power span, and it is queried for the current value, it returns **10.0**, indicating that the current power span is 10 dB.

Related Commands:

:SOURce:POWer:CENTer
:SOURce:POWer:STARt
:SOURce:POWer:STOP

SOURce:POWer:SPAN:FULL



This command sets a power span to cover the full dynamic range of the signal generator. This command causes the signal generator 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. However, a power span query can be used.

NOTE

Use of the **:SOUR**ce parameter and the **dB** label is optional.

Examples:

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

   :SOURce**:POW**er**:SPAN:FULL**

2. To query the signal generator for the current power span, send this command:

   :SOURce**:POW**er**:SPAN?**

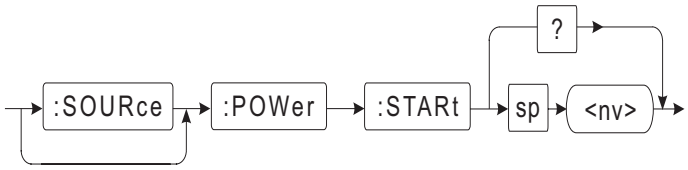
   After the signal generator receives the query command, a read command returns the current power span in <NR2> format. For example, if the 1141B signal generator was commanded to full power span and then queried for the current power span, it returns **103.0**, indicating a full power span of 103 dB.

Related Commands:

:SOURce**:POW**er**:SPAN**
:SOURce**:POW**er**:STAR**t
:SOURce**:POW**er**:STOP**

SOURce:POWer:STARt

Default: 0 dBm
Resolution: 0.1 dB
Limit: –90 dBm



This command sets the start power the signal generator uses when performing power sweeps. It defaults to 0 dBm but can be set anywhere within the dynamic range of the signal generator. On receiving a new start power, the signal generator computes a new center power and span based on the current stop power. These values are returned in response to a :**SOURce:POWer:CENT?** or **:SOURce:POWer:SPAN?** command.

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 is negative.

NOTE

Use of the **:SOURce** parameter and the **dBm** label is optional.

Examples:

1. To set the start power to –20 dBm, send this command:

   :SOURce:POWer:STARt –20 dBm

   or the shortened version:

   :POW:STAR –20

   Upon receiving the new center frequency and span, the signal generator changes both the center frequency and frequency span based on the new start frequency and the current stop frequency.

2. To query the signal generator for the current start power, send this command:

   :SOURce:POWer:STARt?

   After the signal generator receives the query command, a read command returns the current start power in <NR2> format. For example, if the signal generator is at the default start power and it is queried for the current value, it returns **0.0**, indicating that the current start power is 0 dBm.

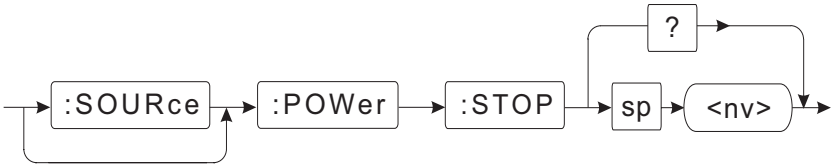
Related Commands:

:SOURce:POWer:CENTer
:SOURce:POWer:SPAN
:SOURce:POWer:STOP

SOURce:POWer:STOP

Default: 10 dBm
Resolution: 0.1 dB
Limit:
1140B: +10 dBm (1 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)

1141B: +13 dBm (2 GHz to <10 GHz)
+10 dBm (10 GHz to 20 GHz)
Option 06: +16 dBm



This command sets the stop power used by the signal generator when performing power sweeps. It defaults to 10 dBm but can be set anywhere within the dynamic range of the signal generator. On receiving a new stop power, the signal generator computes a new center power and span based on the current start power. These values are returned in response to a **:SOURce:POWer:CENT?** or **:SOURce:POWer:SPAN?** command.

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 is negative.

NOTE

Use of the **:SOURce** parameter and the **dBm** label is optional.

Examples:

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

   :SOURce:POWer:STOP 5 dBm

   or the shortened version:

   :POW:STOP 5

2. To query the signal generator for the current stop power, send this command:

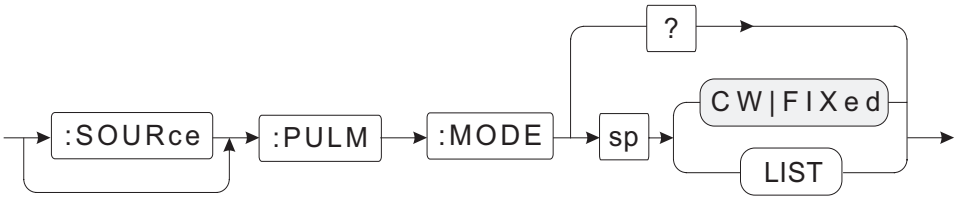
   :SOURce:POWer:STOP?

   After the signal generator receives the query command, a read command returns the current stop power in <NR2> format. For example, if the signal generator is at the default stop power and it is queried for the current value, it returns **10.0**, indicating that 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 **:SOURce:PULM:STATe** command. In the **LIST** mode, the state of pulse modulation is controlled by parameters set using the **:SOURce:LIST** commands.

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

NOTE

Use of the **:SOURce** parameter and the **dBm** label is optional.

Examples:

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

   :SOURce:PULM:MODE LIST

2. To query the signal generator for the current mode setting, send this command:

   :SOURce:PULM:MODE?

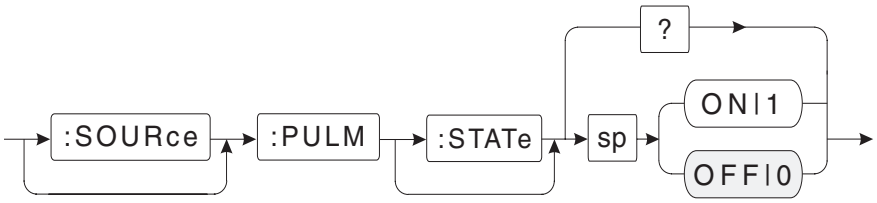
After the signal generator receives the query command, a read command returns the current mode setting in <char> format. For example, if pulse modulation was previously set to **LIST** mode and the signal generator is queried for the current mode, it returns **LIST**, indicating that the state of pulse modulation is currently being controlled by the triggered-list function.

Related Commands:

:SOURce:LIST:PVECtor:MODE LIST
:SOURce:PULM:STATe

SOURce:PULM:STATe

Default: OFF



This command turns the internal pulse modulator on or off.

NOTE

Use of the **:SOURce** and **:STATe** parameters is optional.

Examples:

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

   :SOURce:PULM:STATe ON

   or

   :SOURce:PULM:STATe 1

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

   :PULM:STATe OFF

   or

   :PULM:STATe 0

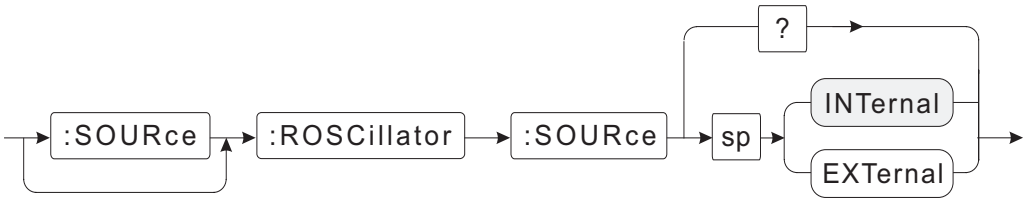
3. To query the signal generator for the current state of the pulse modulation input, send this command:

   :PULM:STATe?

   After the signal generator receives the query command, a read command returns the state of amplitude modulation in <bNR1> format. For example, if the pulse modulation was previously enabled and the signal generator is queried for the current state, it returns **1**, indicating that pulse modulation is enabled (ON).

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

Default: INTernal



This command selects the source of the reference oscillator for the signal generator. **EXT**ernal causes the signal generator to use an external 10 MHz signal from the front-panel connector.

> **NOTE**
>
> Use of the **:SOUR**ce parameter at the beginning of the command is optional.

Examples:

1. To select an external reference, send this command:

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

   In this example, two commands are sent, separated by a semicolon. The first command turns off the 10 MHz output connector on the front panel, which configures the connector 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, the second command would normally be the only one necessary unless the output has previously been enabled.

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

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

3. To query the signal generator for the reference oscillator source, send this command:

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

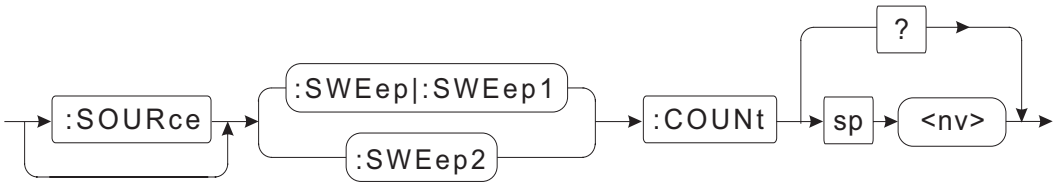
   After the signal generator receives the query command, a read command returns the source of the reference oscillator in <char> format. For example, if the reference oscillator was previously set to internal, and the signal generator is queried for the current state, it returns **INT**, indicating that the reference-oscillator source is set to internal.

Related Command:

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

SOURce:SWEep<n>:COUNt

Default: 1
Range: 1 to 32000



The sweep count determines how many times the signal generator repeats 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 the variable blank denotes a frequency sweep and using a 2 denotes a power sweep.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

To set sweep count for a frequency sweep to 4, send this command:

:SOURce:SWEep1:COUNt 4

or

:SOURce:SWEep:COUNt 4

NOTE

As described in the example, **SWEep** and **SWEep 1** are equivalent.

4. To set sweep count for a power sweep to 3, send this command:

   :SOURce:SWEep2:COUNt 3

5. To query the signal generator for the current value of **SWEep2** count, send this command:

   :SOURce:SWEep2:COUNt?

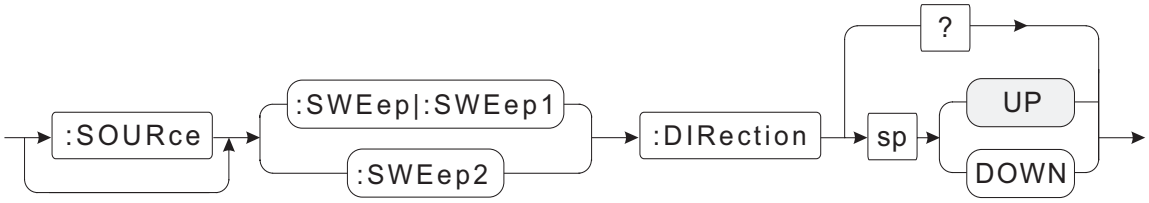
   After the signal generator receives the query command, a read command returns the current **SWEep2** count in <NR1> format. For example, if the **SWEep2** count was previously set to the default value and the signal generator is queried for the current setting, it returns **1**, indicating that for power sweeps, the signal generator will perform a single sweep in response to a trigger.

Related Commands:

:SOURce:FREQuency:CENTer	**:SOURce:POWer:SPAN**
:SOURce:FREQuency:SPAN	**:SOURce:POWer:STARt**
:SOURce:FREQuency:STARt	**:SOURce:POWer:STOP**
:SOURce:FREQuency:STOP	**:SOURce:SWEep<n>:DIRection**
:SOURce:POWer:CENTer	**:SOURce:SWEep<n>:DWELL**

SOURce:**SWE**ep<n>:**DIR**ection

Default: UP



The sweep-direction command determines which direction the signal generator will step for either frequency or power sweeps. The **SWEep** variable "n" selects between frequency and power sweeps. Using either a 1 or leaving the variable blank denotes a frequency sweep, and using a 2 denotes a power sweep.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

1. To command the signal generator to step down when performing a power sweep, send this command:

   :SOURce:**SWE**ep**2:DIR**ection **DOWN**

   or

   :SWEep**2:DIR DOWN**

2. To set up the signal generator to step down when performing a frequency sweep, send this command:

   :SOURce:**SWE**ep**1:DIR**ection **DOWN**

   or

   :SWEep:**DIR DOWN**

   NOTE

   As demonstrated in this example, **SWE**ep and **SWE**ep**1** are equivalent.

3. To query the signal generator for the current direction of **SWEep2**, send this command:

   :SOURce:**SWE**ep**2:DIR?**

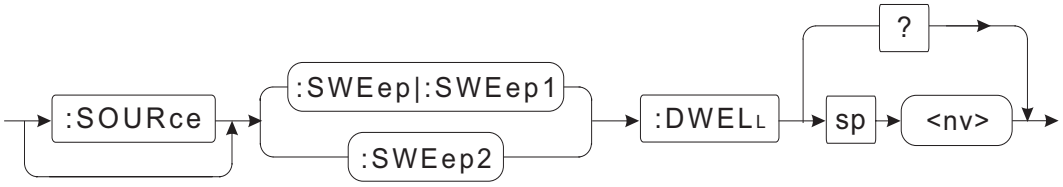
   After the signal generator receives the query command, a read command returns the current sweep direction in <char> format. For example, if the sweep direction was previously set to down and the signal generator is queried for the current setting, it returns **DOWN**, indicating that the sweep direction is set to down.

Related Commands:

:SOURce:**FREQ**uency:**CENT**er	**:SOUR**ce:**POW**er:**SPAN**
:SOURce:**FREQ**uency:**SPAN**	**:SOUR**ce:**POW**er:**STAR**t
:SOURce:**FREQ**uency:**STAR**t	**:SOUR**ce:**POW**er:**STOP**
:SOURce:**FREQ**uency:**STOP**	**:SOUR**ce:**SWE**ep<n>:**COUN**t
:SOURce:**POW**er:**CENT**er	**:SOUR**ce:**SWE**ep<n>:**DWEL**L

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 sweep mode, the signal generator moves between adjacent points in steps. The length of time the signal generator remains at each point is controlled by setting of the dwell-time variable. Dwell time does not include switching time, which is the time it takes for the signal-generator hardware to change power or to settle to the new frequency. Typically, the switching time is 35 ms. Dwell time controls only the length of time the signal generator remains at each point once the signal is settled. The total time required to complete a sweep is determined by this 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 the variable blank denotes a frequency sweep, and using a 2 denotes a power sweep.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To set the amount of time the signal generator waits at each frequency point when performing a frequency sweep to 0.3 seconds, send this 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 signal generator waits at each power point when performing a power sweep to 1 second, send this command:

   :SOURce:SWEep2:DWELL 1

3. To query the signal generator for the current dwell time of **SWEep1**, send this command:

   :SOURce:SWEep1:DWELL?

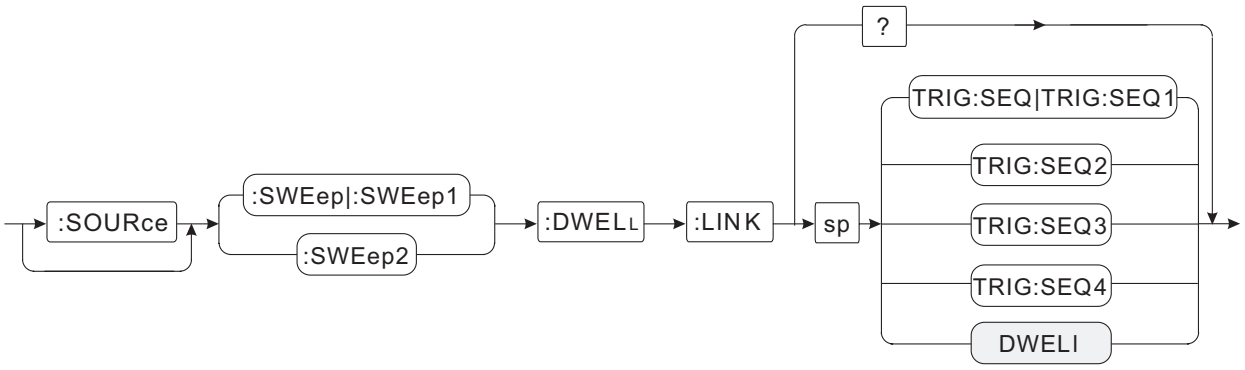
   After the signal generator receives the query command, a read command returns the current dwell time of **SWEep1** in <NR3> format. For example, if the **SWEep1** dwell time was previously set to 2 seconds, and the signal generator is queried for the current setting, it returns **2.000**, indicating that the signal generator will dwell (wait) at each step in a frequency sweep for 2 seconds.

Related Commands:

:SOURce:FREQuency:CENTer
:SOURce:FREQuency:SPAN
:SOURce:FREQuency:STARt
:SOURce:FREQuency:STOP
:SOURce:POWer:CENTer
:SOURce:POWer:SPAN
:SOURce:POWer:STARt
:SOURce:POWer:STOP
:SOURce:SWEep<n>:COUNt
:SOURce:SWEep<n>:DIRection
:SOURce:SWEep<n>:DWELL:LINK

SOURce:SWEep<n>:DWELL:LINK

Default:DWELL



In sweep mode, the signal generator moves between adjacent points in steps. The length of time the signal generator remains at each point is controlled by dwell. Dwell can be a fixed time; it can be 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.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

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

   :SOURce:SWEep1:DWELL:LINK DWELL

   To link dwell to a trigger from trigger sequence 2, send this command:

   :SOURce:SWEep1:DWELL:LINK TRIG:SEQ2

   NOTE

   The arguments **:SEQ** and **:SEQ1** are equivalent.

2. To query the signal generator for the current dwell linkage for **SWEep1**, send this command:

   :SOURce:SWEep1:DWELL:LINK?

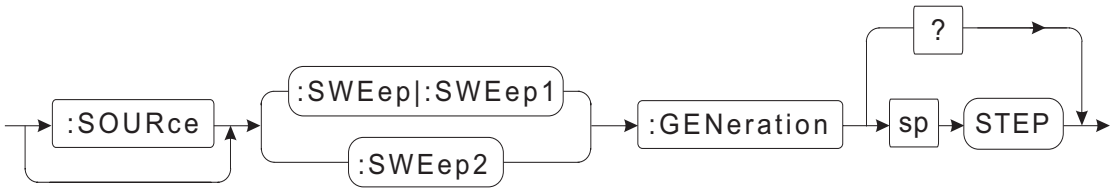
   After the signal generator receives the query command, a read command returns the current **SWEep1** dwell linkage in <NR3> format. For example, if the signal generator is using the default linkage for **SWEep1,** and it is queried for the current linkage, it returns **DWEL**, indicating that the internal dwell timer is providing the trigger to move between steps in a frequency sweep.

Note that **SWEep1** is a sweep of frequency only, and **SWEep2** is a sweep of power only.

Related Command: **:SOURce:SWEep<n>:DWELL**

SOURce:**SWE**ep<n>:**GEN**eration

Default: STEP



This command is defined in SCPI for selecting between analog and stepped sweep modes. This command has no real purpose for the 1140B and 1141B signal generators, because these signal generators only support step mode. However, the command was included to maintain compatibility with the SCPI language.

The **SWE**ep variable "n" selects between frequency and power sweeps. Using either a 1 or leaving the variable blank denotes a frequency sweep, and using a 2 denotes a power sweep.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

1. To set up the signal generator to use the step mode for frequency sweeps, send this command:

   :SOURce:**SWE**ep**1:GEN STEP** or **:SOUR**ce:**SWE**ep**:GEN STEP**

   NOTE

   As demonstrated in this example, **SWE**ep and **SWE**ep**1** are equivalent.

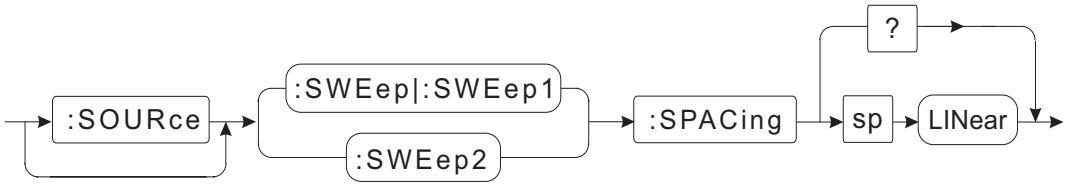
2. To query the signal generator for the current sweep mode of **SWE**ep**2**, send this command:

   :SOURce:**SWE**ep**2:GEN?**

   After the signal generator receives the query command, a read command returns the current sweep mode in <char> format. The 1140B and 1141B signal generators support only the step mode, so if one of these signal generators is queried for the current sweep mode, it should returns **STEP**, indicating that the signal generator is in step mode.

SOURce:SWEep<n>:SPACing

Default: LINear



This command is defined in SCPI for selecting between linear and logarithmic modes for sweep functions. The command has no real purpose for the 1140B and 1141B signal generators, because these signal generators support only the linear mode for both frequency and power sweeps. However, the command 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 the variable blank denotes a frequency sweep, and using a 2 denotes a power sweep.

> **NOTE**
>
> Use of the **:SOURce** parameter is optional.

Examples:

1. To command the signal generator to use linear mode for frequency sweeps, send this command:

   :SOURce:SWEep1:SPACing LINear

   or

   :SOURce:SWEep:SPACing LINear

   > **NOTE**
   >
   > As demonstrated in this example, **SWEep** and **SWEep1** are equivalent.

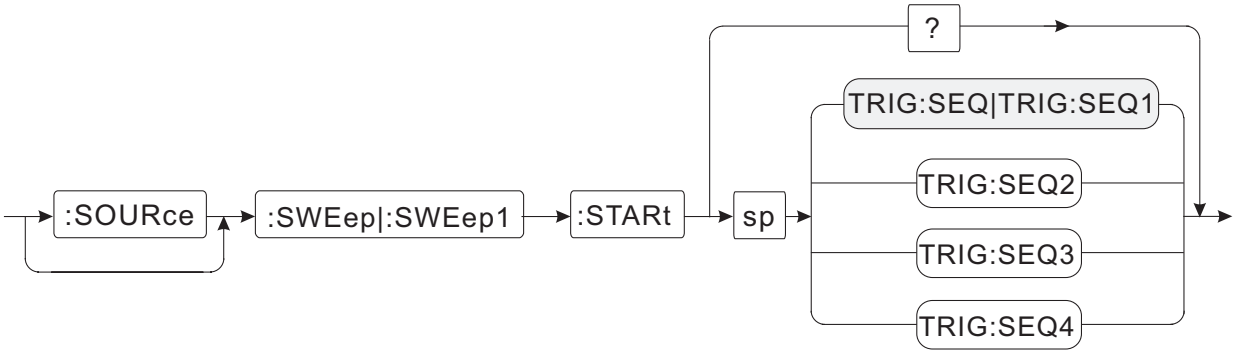
2. To query the signal generator for the current sweep-spacing mode of **SWEep2**, send this command:

   :SOURce:SWEep2:SPAC?

   After the signal generator receives the query command, a read command returns the current sweep spacing mode in <char> format. The signal generator supports only the linear mode, so if it is queried for the current sweep-spacing mode, it returns **LIN**, indicating that the linear mode is current.

SOURce**:SWE**ep**1:STAR**t

Default: TRIG:SEQ1



The signal generator provides four trigger sequences. This command is used to select which trigger sequence initiates a frequency sweep.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

1. To command the signal generator to initiate a frequency sweep based on trigger-sequence 2, send this command:

   :SOURce**:SWE**ep**1:STAR**t **TRIG:SEQ2**

   or

   :SOURce**:SWE**ep**:STAR**t **TRIG:SEQ2**

2. To query the signal generator for the trigger sequence being used for **SWE**ep**1**, send this command:

   :SOURce**:SWE**ep**1:STAR**t**?**

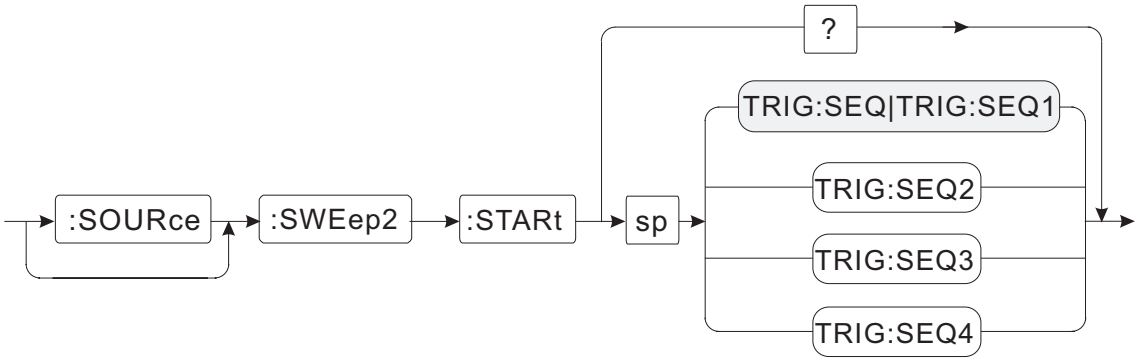
   After the signal generator receives the query command, a read command returns the current trigger sequence in <char> format. For example, if the **SWE**ep**1** trigger sequence was previously set to the default, and the signal generator is queried for the current setting, it returns **TRIG:SEQ1**, indicating that the **SWE**ep**1** trigger sequence is set to trigger from sequence 1.

NOTE

The arguments :**SEQ** and :**SEQ1** are equivalent.

SOURce:SWEep2:STARt

Default: TRIG:SEQ1



The signal generator provides four trigger sequences. This command selects the trigger sequence that initiates a power sweep.

NOTE

Use of the **:SOUR**ce parameter is optional.

Examples:

1. To command the signal generator to initiate a power sweep based on trigger sequence 3, send this command:

   :SOURce**:SWE**ep**2:STAR**t **TRIG:SEQ3**

2. To query the signal generator for the trigger sequence being used for **SWE**ep**2**, send this command:

   :SOURce**:SWE**ep**2:STAR**t**?**

   After the signal generator receives the query command, a read command returns the current trigger sequence in <char> format. For example, if the **SWEep1** trigger sequence was previously set to the default, and the signal generator is queried for the current setting, it returns **TRIG:SEQ1**, indicating that the **SWEep2** trigger sequence is set to trigger from sequence 1.

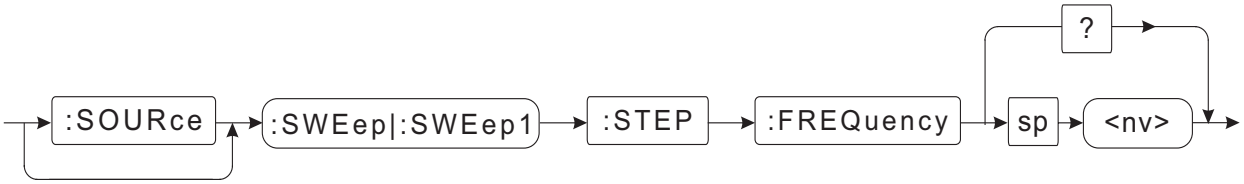
   sequence 1.

NOTE

The arguments :**SEQ** and :**SEQ1** are equivalent.

SOURce:SWEep1:STEP:FREQuency

Default: 1 GHZ
Resolution: 1 Hz



In the frequency-sweep mode, the signal generator moves between adjacent points in steps. This command controls the 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 is generated.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

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

   :SOURce:SWEep1:STEP:FREQuency 1MHz

   If the frequency units (MHz) were omitted, the units selected by the units subsystem are 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 required, the following shortened version of the command is also valid.

   :SWE:STEP:FREQ 1000000

2. To query the signal generator for the current frequency step size, send this command:

   :SOURce:SWEep1:STEP:FREQuency?

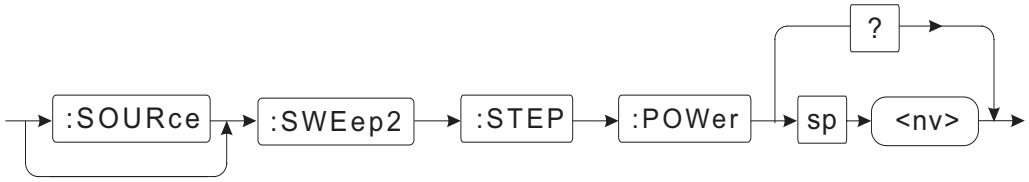
   After the signal generator receives the query command, a read command returns the current frequency step size in <NR2> format. For example, if the **SWEep1** step size was previously set to the default value and the signal generator is queried for the current setting, it returns **1000000000**, indicating that the **SWEep1** frequency step size is set to 1 GHz.

Related Commands:

:SOURce:FREQuency:CENTer
:SOURce:FREQuency:SPAN
:SOURce:FREQuency:STARt
:SOURce:FREQuency:STOP
:SOURce:SWEep1<n>**:COUNt**
:SOURce:SWEep1<n>**:DIRection**
:SOURce:SWEep1<n>**:DWELL**

SOURce:SWEep2:STEP:POWer

Default: 1 dB
Resolution: 0.1 dB



In power-sweep mode, the signal generator moves between adjacent points in steps. This command controls the size of the power stepa. 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 is generated.

NOTE

Use of the **:SOURce** parameter is optional.

Examples:

1. To set the power step size to 0.5 dB, send this 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 signal generator for the current power step size, send this command:

   :SOURce:SWEep2:STEP:POWer?

   After the signal generator receives the query command, a read command returns the current power step size in <NR2> format. For example, if the **SWEep2** step size was previously set to the default value, and the signal generator is queried for the current setting, it returns **1.0**, indicating that the **SWEep2** power step size is set to 1 dB.

Related Commands:

:SOURce:POWer:CENTer
:SOURce:POWer:SPAN
:SOURce:POWer:STARt
:SOURce:POWer:STOP
:SOURce:SWEep2<n>:COUNt
:SOURce:SWEep2<n>:DIRection
:SOURce:SWEep2<n>:DWELL

STATUS SUBSYSTEM

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 signal generator 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 in "Common Commands, IEEE 488.2" on page 4-4.

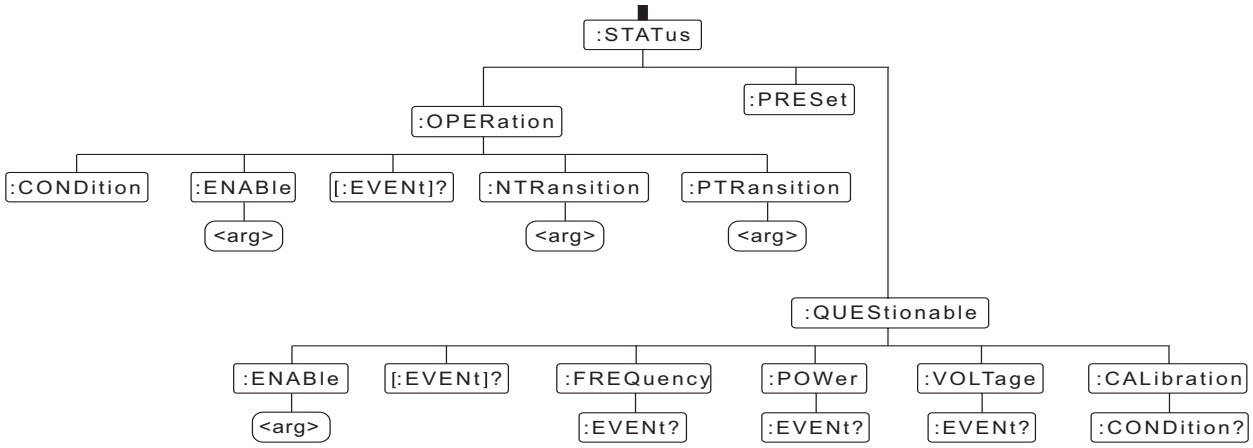


Figure 4-2. Status Subsystem Command Tree

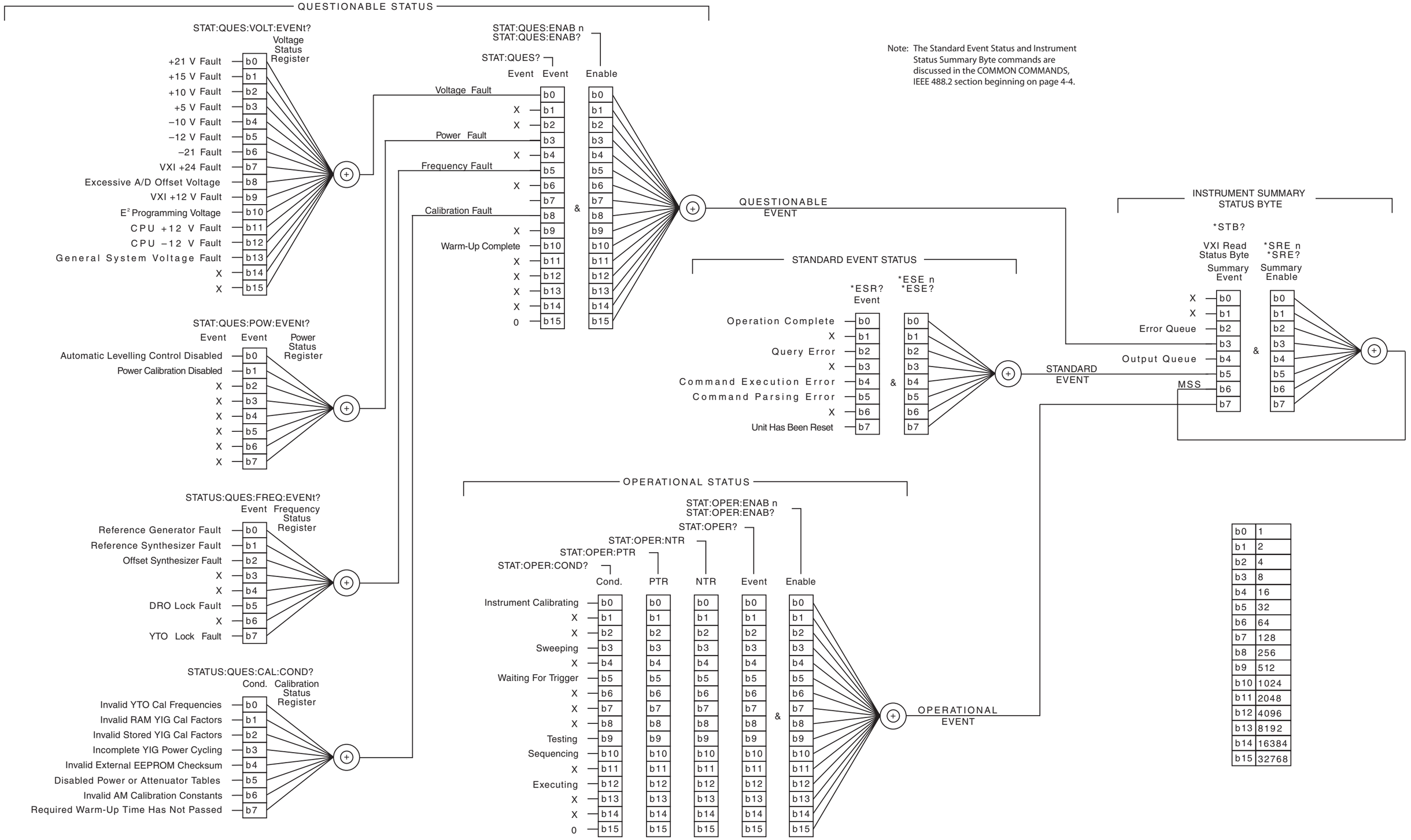
There are three types of registers contained in the status-reporting structures: condition, event, and enable. The condition registers are continuously updated to reflect the current status of the signal generator. Reading the contents of a condition register does not change its contents. The event registers are all initially cleared to zero. Whenever an event occurs that corresponds 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 signal generator. 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 signal generator. 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 4-3.

This Page Intentionally Left Blank



Note: The Standard Event Status and Instrument Status Summary Byte commands are discussed in the COMMON COMMANDS, IEEE 488.2 section beginning on page 4-4.

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

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

NOTE: TESTING (Bit 9) is true during a self test initiated by a *TST command. SEQUENCING (Bit 10) is active only in the LIST mode.

Figure 4-3. Status Structure

This Page Intentionally Left Blank

STATus:OPERation:CONDition?



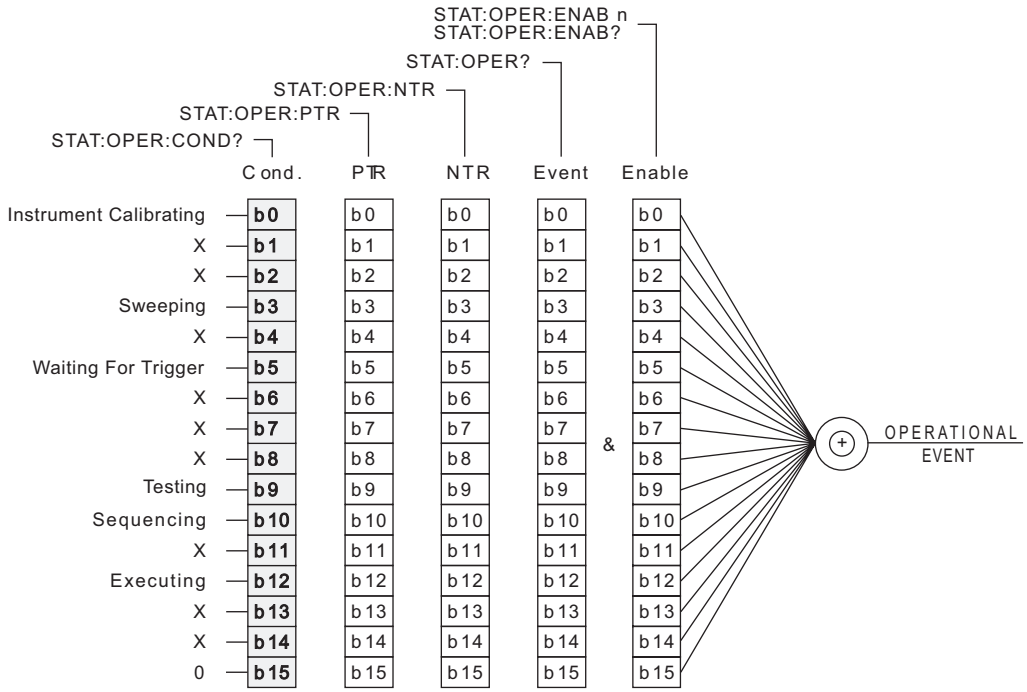
This query returns the contents of the Operational-condition register. Reading a condition register does not change its contents. Bit 15 of this register is always zero. All undefined bits are always cleared to zero.

Example:

To query the Operational condition register, send this command:

:STATus:OPERation:CONDition?

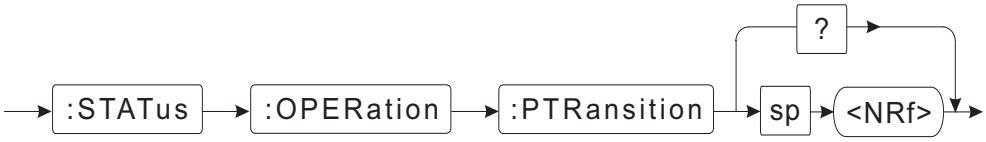
This query will return a numeric string in the <NR1> format. To determine which bits are set in the register, convert the returned string to its numeric value and then to the corresponding binary bit pattern.



NOTE

X denotes unused bits, which are always cleared to 0.

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



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 is set.

Examples:

1. To cause bit 0 of the Operational-event-status register to be set whenever the signal generator is calibrating, send this 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 signal generator is calibrating, a 1 is written to bit 0 of the Operational-event-status register.

2. To query the signal generator for the current contents of the positive-transition filter, send this command:

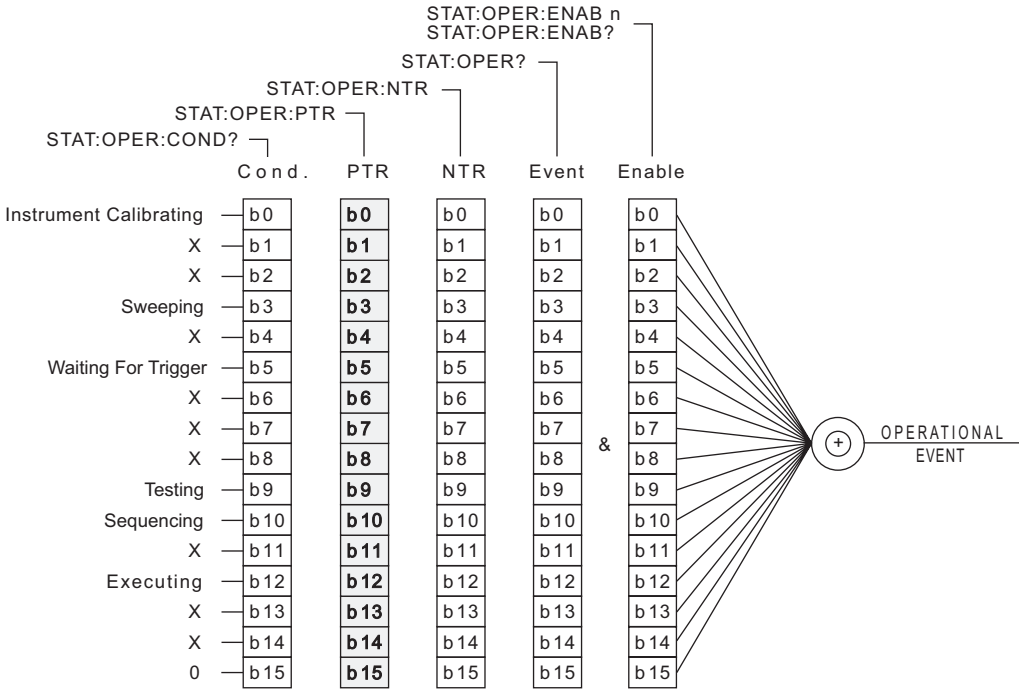
   :STATus:**OPER**ation:**PTR**ansition**?**

   The signal generator returns a numeric string, in <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 **4096**, this indicates that 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

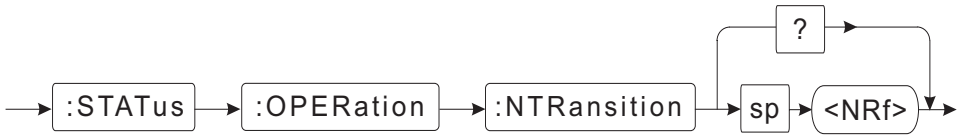
Sending the common commands ***CLS** and ***RST** does not change the contents of a transition filter.



NOTE

X denotes unused bits, which are always cleared to 0.

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



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 is set.

Examples:

1. To cause bit 12 of the Operational-event-status register to be set whenever the signal generator is finished executing a command, send this 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 that the signal generator is finished executing a command, a **1** is written to bit 12 of the Operational-event-status register.

2. To query the signal generator for the current contents of the negative-transition filter, send this command:

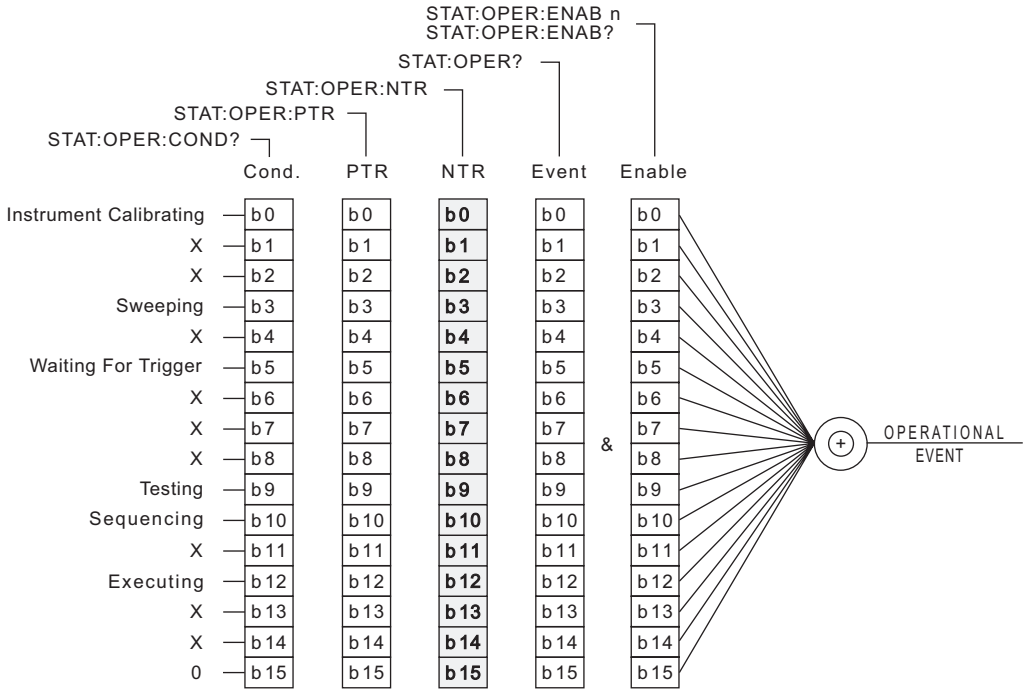
   :STATus**:OPER**ation**:NTR**ansition**?**

   The signal generator returns a numeric string in <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

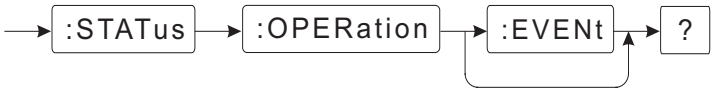
Sending the common commands ***CLS** and ***RST** do not change the contents of a transition filter.



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 <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 clears the register.

For example, if you set all the bits in the positive-transition filter to True and all the bits in the negative-transition filter to False, and if you send a query at this point and it returns **4096**, it indicates a False-to-True transition of bit 12 in the Operational-condition register occurred since the last time the Operational-event register was queried.

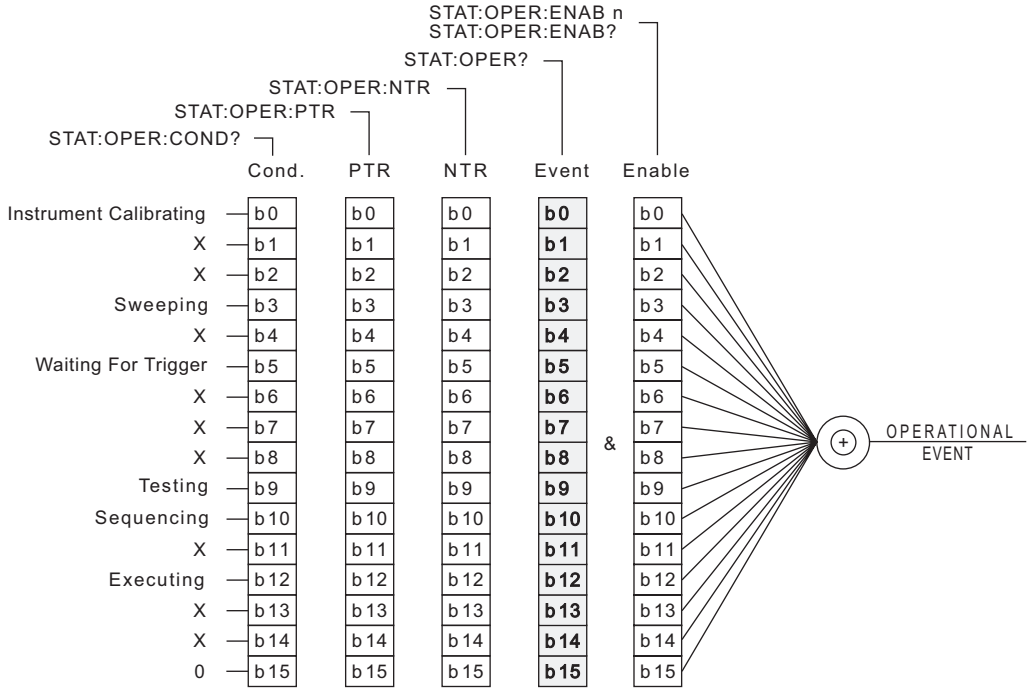
To query the signal generator for the contents of the Operational event register, send this command (use of the :**EVENt** parameter is optional):

:STATus:OPERation:EVENt?

or

:STATus:OPERation?

The signal generator returns a numeric string, in <NR1> format corresponding to the bits set. To decode the string, first convert it to its numeric value and then to the corresponding binary bit pattern.

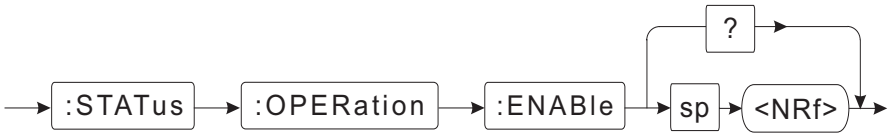


NOTE

X denotes unused bits, which are always cleared to 0.

STATus**:OPER**ation**:ENAB**le

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** is written in bit 7 of the Summary-event register. Querying an enable register does 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 this command:

   :STATus**:OPER**ation**:ENAB**le **4096**

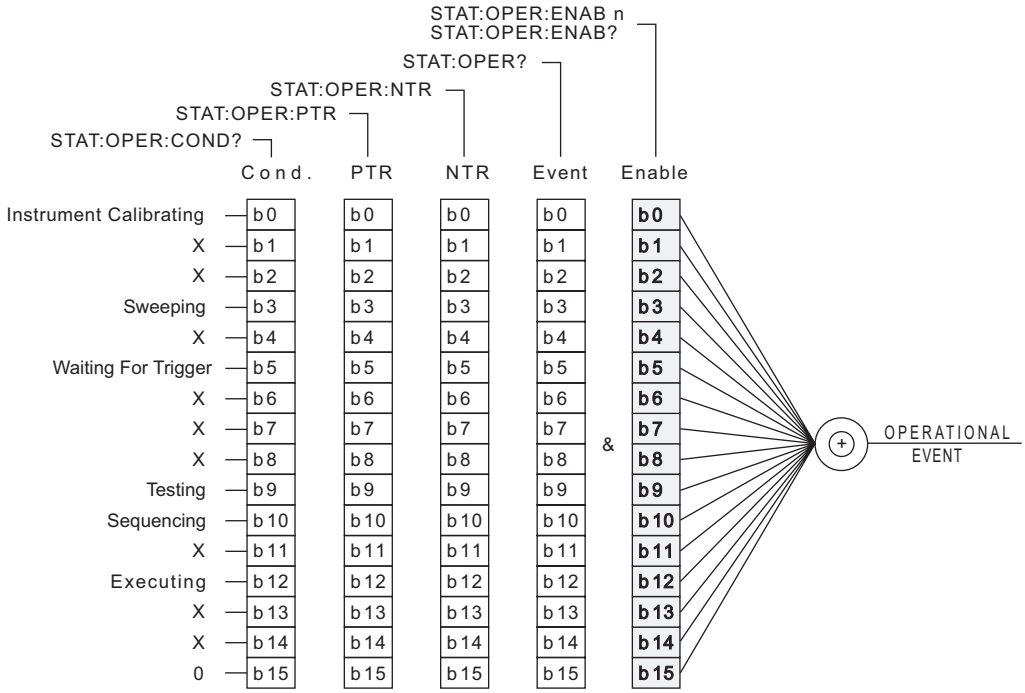
   This sets bit 12 in the Operational-enable register, which causes a **1** to be written in bit 7 of the Summary-event register, if bit 12 of the Operational-event register becomes True.

2. To query the signal generator for the current contents of the Operational-enable register, send this command:

   :STATus**:OPER**ation**:ENAB**le**?**

   The signal generator returna a numeric string, in <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** is written to bit 7 of the Summary-event register.



NOTE

X denotes unused bits, which are always cleared to 0.

STATus**:PRES**et



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

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

:STATus**:QUEStionab**le**:ENAB**le **0**
:STATus**:OPERation:ENAB**le **0**
:STATus**:OPERation:NTR**ansition **0**, or
:STATus**:OPER**ation**:PTR**ansition **65535**

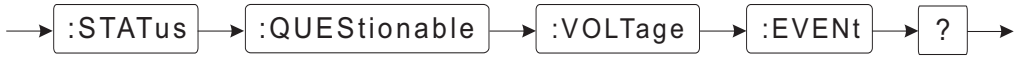
This command does not affect either the status byte or the standard-event status and does not clear any of the event registers.

Example:

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

:STATus**:PRES**et

STATus:**QUES**tionable:**VOLT**age:**EVEN**t?



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 that the unit needs to be serviced.

Example:

To query the signal generator for the current contents of the Questionable-Voltage-event register, send this command:

:STATus:**QUES**tionable:**VOLT**age:**EVEN**t?

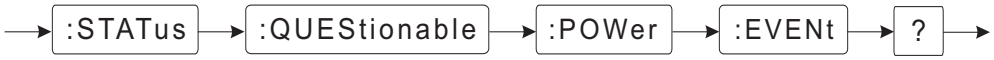
The signal generator returns a numeric string, in <NR1> format corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.

+21 V Fault	b 0
+15 V Fault	b 1
+10 V Fault	b 2
+5V Fault	b 3
–10 V Fault	b 4
–12 V Fault	b 5
–21 V Fault	b 6
VXI +24 V Fault	b 7
Excessive A/D Offset Voltage	b 8
VXI +12 V Fault	b 9
^{2}E Programming Voltage	b 10
CPU +12 V Fault	b 11
CPU –12 V Fault	b 12
General System Voltage Fault	b 13
X	b 14
X	b 15

NOTE

X denotes unused bits, which are always cleared to 0.

STATus:**QUES**tionable:**POW**er:**EVEN**t?



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

Bit 0 of this register is set when automatic leveling for the signal generator is disabled. Sending the command **:SOURce:POWer:ALC OFF** disables automatic power leveling for the signal generator and causes bit 0 to be set. Once set, bit 0 remains set until the register is queried.

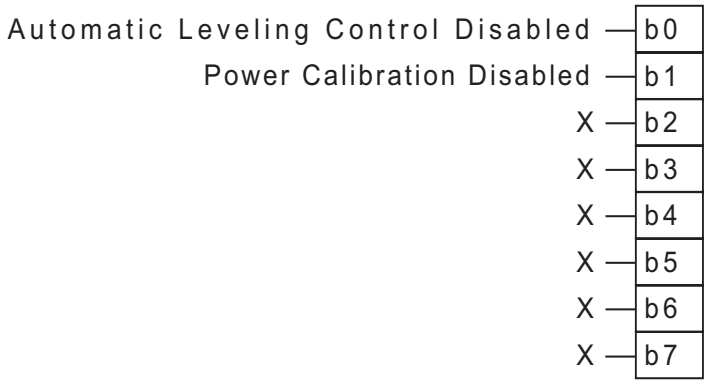
Bit 1 of this register is set when power calibration is disabled. Sending the command **:CAL**ibration:**POW**er:**STAT**e **OFF** disables power calibration for the signal generator and causes bit 0 to be set. Once set, bit 0 remains set until the register is queried.

Example:

To query the signal generator for the current contents of the Questionable-Power-event register, send this command:

:STATus:**QUES**tionable:**POW**er:**EVEN**t?

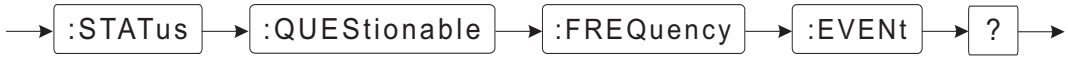
The signal generator returns a numeric string, in <NR1> format corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.



NOTE

X denotes unused bits, which are always cleared to 0.

STATus:**QUES**tionable:**FREQ**uency:**EVEN**t?



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

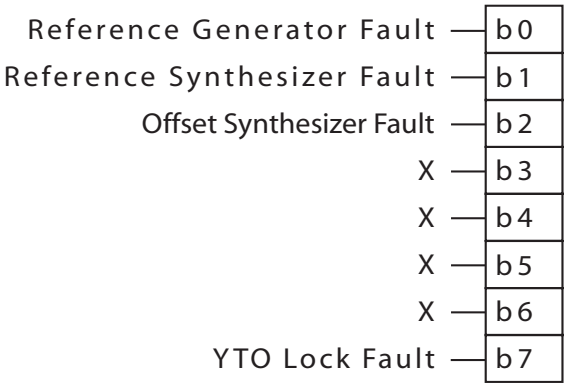
If a fault is detected, perform a query to clear the register. If the fault reappears, it indicates the unit needs to be serviced.

Example:

To query the signal generator for the current contents of the Questionable-Frequency-event register, send this command:

:STATus:**QUES**tionable:**FREQ**uency:**EVEN**t?

The signal generator returns a numeric string, in <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:**QUES**tionable:**CAL**ibration:**COND**ition?



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

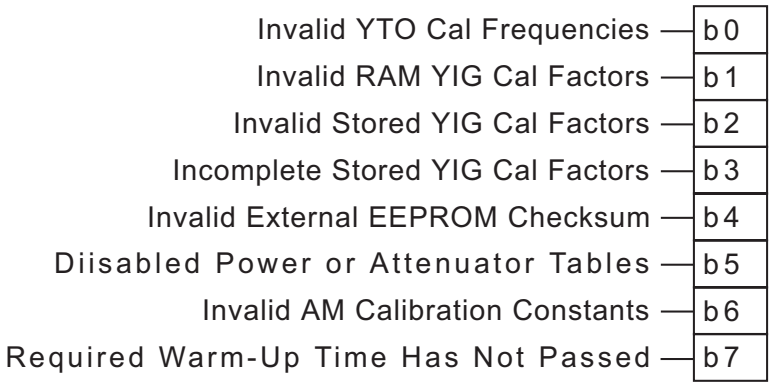
If a fault is detected, performing a query does not clear the fault, since it is a condition register. Cycling signal-generator power may clear the fault. If it does not, the unit needs to be serviced.

Example:

To query the signal generator for the current contents of the Questionable-Calibration-condition register, send this command:

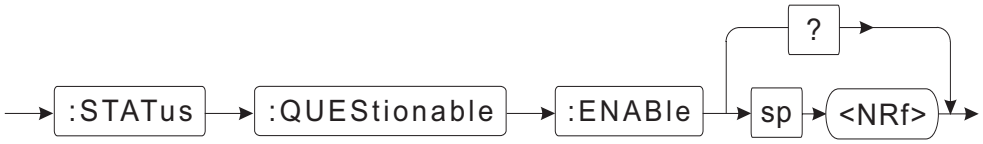
:STATus:**QUES**tionable:**CAL**ibration:**COND**ition?

The signal generator returns a numeric string, in <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.



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

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** is written in bit 3 of the Summary-event register. Querying an enable register does 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 this command:

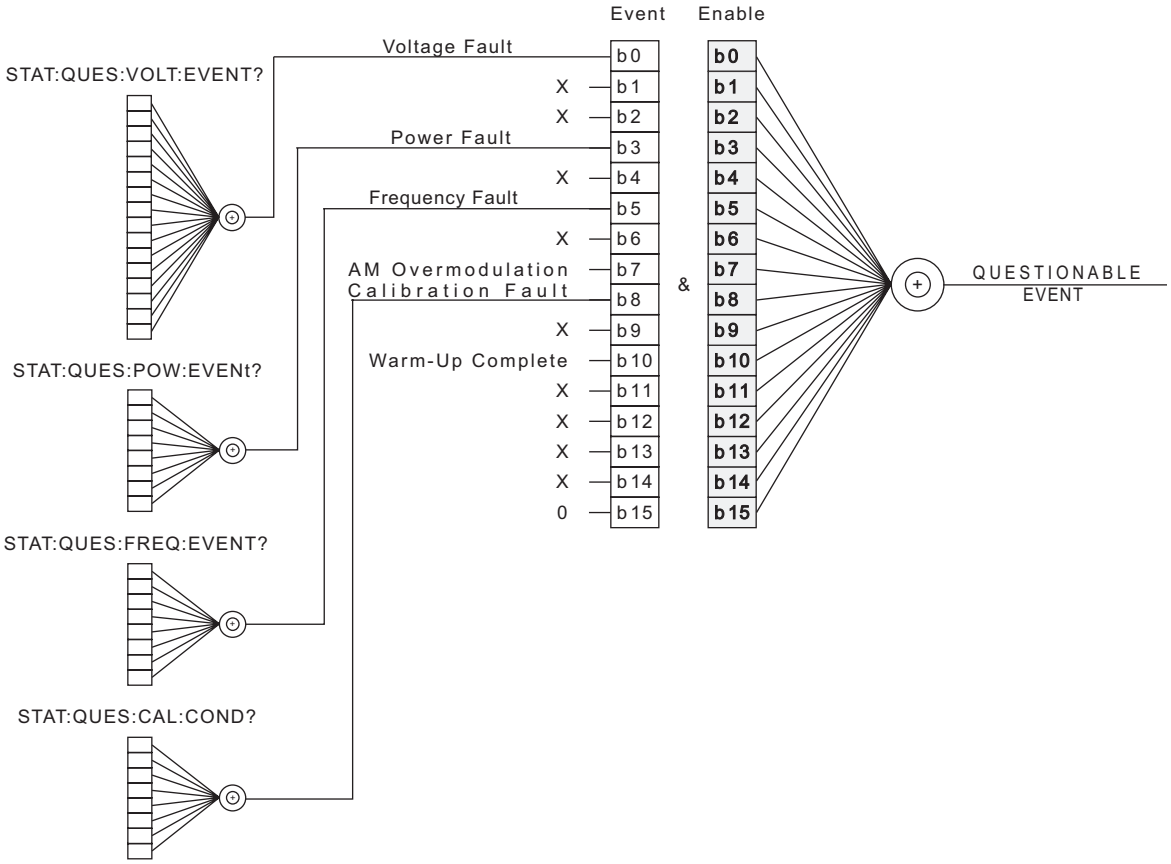
   :STATus:**QUES**tionable:**ENAB**le **425**

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

2. To query the signal generator for the current contents of the Questionable enable register, send this command:

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

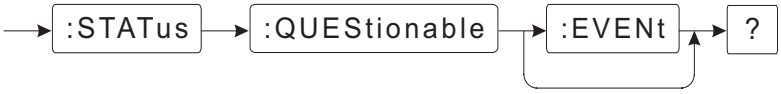
The signal generator returns a numeric string in <NR1> format corresponding to the bits set. To decode the string, first convert the string to its numeric value and then to the corresponding binary bit pattern.



NOTE

X denotes unused bits, which are always cleared to 0.

STATus**:QUES**tionable**:EVEN**t**?**



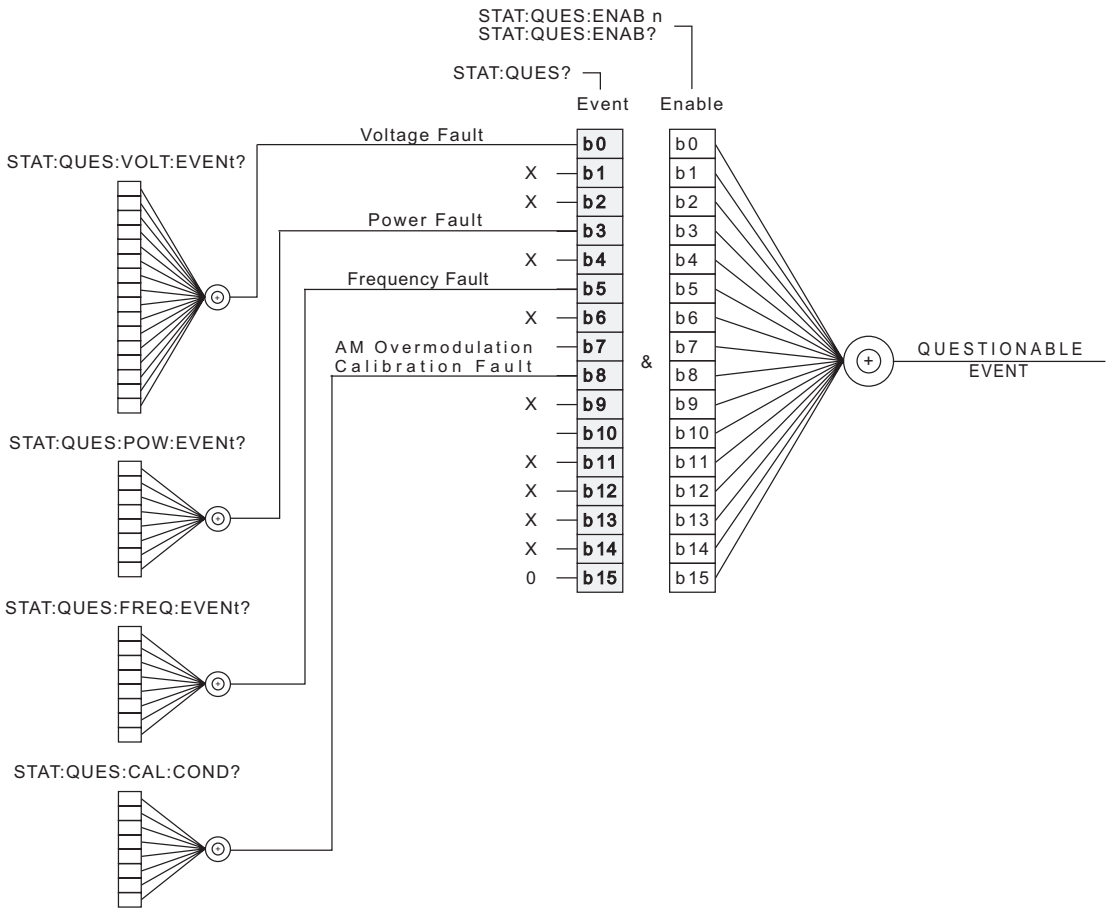
This query returns the contents of the Questionable-event register in <NR1> format. Its contents are determined by the four registers reporting to it and by the two signal-generator conditions of AM Overmodulation and Warm-up Complete, as shown below. To decode the string, first convert the string to 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 preceeded by an implied negative-transition filter. It is True only once each time the signal-generator power is cycled after the required warm-up time (15 minutes) has passed. Reading the Questionable-event register clears bit 10 and it does not become True unless the signal-generator power is cycled.

Example:

To query the signal generator for the current contents of the Questionable-event register, send this command:

:STATus**:QUES**tionable**:EVEN**t**?**



Note that X denotes unused bits, which are always cleared to 0.

SYSTEM SUBSYSTEM

The System Subsystem comprises the functions that are not related to signal-generator performance. These are error query and version query.

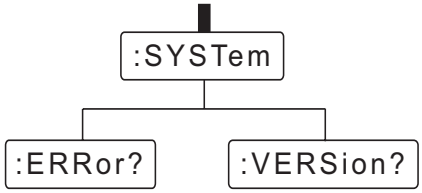


Figure 4-4. System Subsystem Command Tree

SYSTem:ERRor?



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

Example:

To query the system error, send this command:

:SYSTem:ERRor?

The signal generator responds 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 is incorrectly entered, the error message might read

–131,"Invalid suffix"

and sending **:SYST:ERRor** without the **?** would result in this 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 this error:

–350,"Queue overflow"

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

0,"No error"

The following is a list of possible error messages:

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

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



This query determines the SCPI version number with which the signal generator complies. The response is in the form **YYYY.V**, where the Ys indicate the year-version and the V indicates the approved revision number for that year.

Example:

To determine signal generator's system version, send this command:

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

After the signal generator receives the query command, a read command returns the current system version in <char> format. Specifically, an example of the string returned is:

1993.0

which indicates the signal generator complies to the 1993 version of SCPI.

TRIGGER SUBSYSTEM

The Phase Matrix 1140B and 1141B implement 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 signal generator. The Phase Matrix 1140B and 1141B signal generators not only respond to triggers from the VXI backplane but can also source these triggers to provide synchronization between a variety of instrument modules.

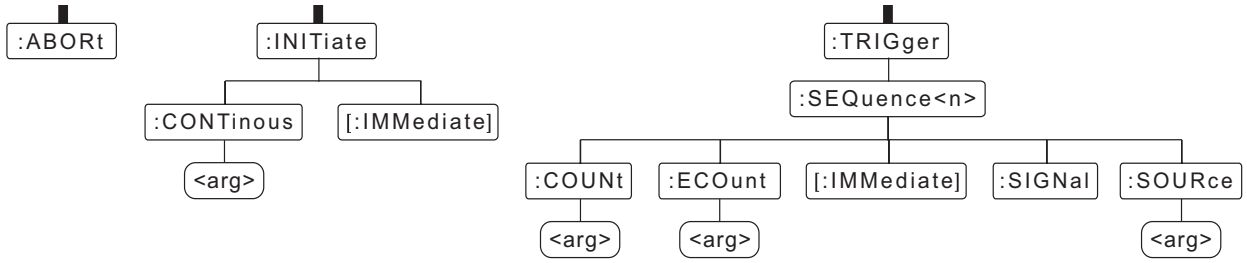


Figure 4-5. Trigger Subsystem Command Tree

In general, the trigger subsystem comprises commands used to synchronize an action in the signal generator with an event. Changing the output frequency or power are examples of a signal-generator action. An event is a change such as a trigger from the VXI backplane. The triggering model used in the signal generator consists of four identical sequences, denoted **SEQ1** through **SEQ4**, with three states per sequence. The trigger model of these sequences is shown in Figure 4-6.

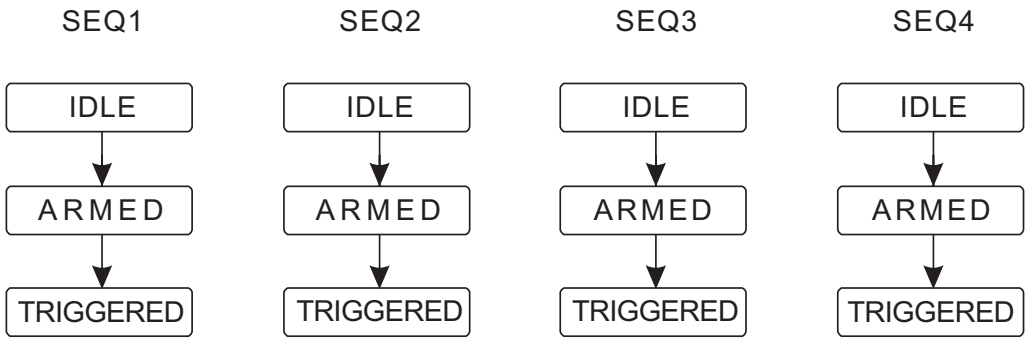


Figure 4-6. Trigger Model

Each trigger sequence consists of three states, or layers. The idle state is the least active of the three. In the idle state, the signal generator is simply waiting to be commanded to the armed state. Upon command, the signal generator moves from the idle state to the armed state, where it remains while waiting for the trigger. After receiving the trigger, the signal generator moves from the armed state to the triggered state and initiates the programmed action, such as changing frequency. After completing the action, the signal generator 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 an :**ABOR**t command, all four triggering sequences are placed in the idle state. The **:INIT**iate**[:IMM**ediate**]** command or the **:INIT**iate**:CONT**inuous **ON** command moves the signal generator out of the idle state and into the armed state. The **:INIT**iate**[:IMM**ediate**]** 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 sequences. Upon receiving this command, the signal generator returns to the armed state instead of the idle state after triggering an action.

The signal generator remains in the armed state until the trigger is detected. The **:TRIG:SEQ**<n>**:SOUR** command selects 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 provides the trigger, but when the **:TRIG:SEQ**<n>**:ECO** command is used, the signal generator can be programmed to wait for a specific number of transitions (**ECOunt**) before proceeding to the triggered state.

Table 4-2. Trigger Sources

BUS

ECLTrg0,ECLTrg1,

TTLTrg0,TTLTrg1,TTLTrg2,TTLTrg3,TTLTrg4,TTLTrg5,TTLTrg6,TTLTrg7

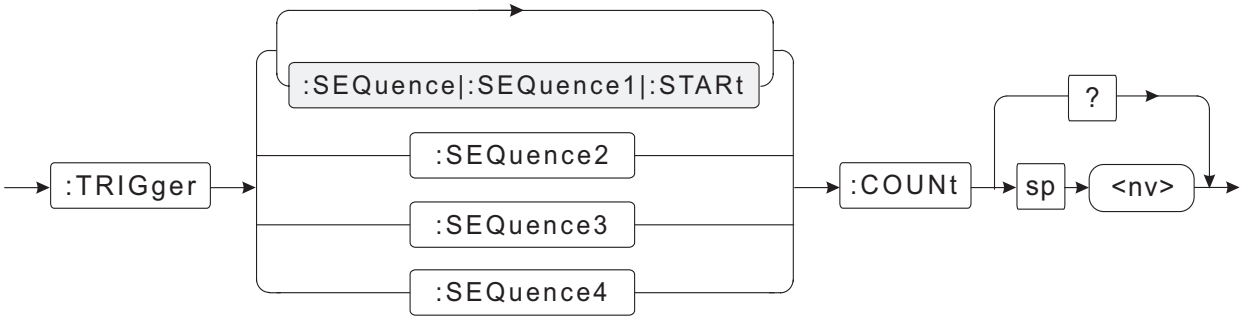
The BUS source actually represents two sources, either the SCPI command ***TRG** or the VXI word serial trigger command. Each trigger sequence (**SEQ1**...**SEQ4**) can have a different trigger source or a single source can be linked to all four trigger sequences. By default, all four trigger sequences 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 idle state in the trigger model. In the trigger model, the signal generator begins in the idle state. Upon receiving a command such as **:INIT**, the signal generator 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 signal generator. After initiating the action, the signal generator either moves back to the idle state or moves back to the armed state, depending on the trigger-sequence-count variable. This variable controls the number of times the signal generator loops between the armed state and the triggered state before returning to the idle state.

Examples:

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

   :TRIGger:SEQuence1:COUNt 6

   or

   :**TRIGger:SEQuence:COUNt 6**

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

   :TRIGger:SEQuence3:COUNt 34

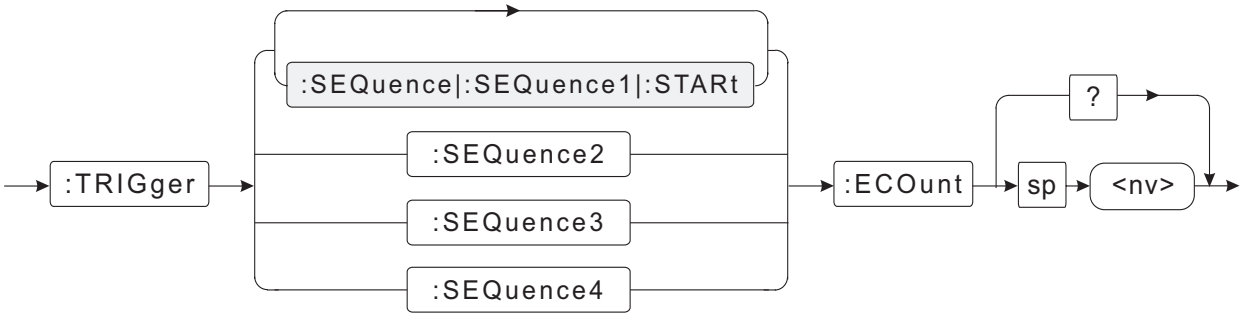
3. To query the signal generator for the current trigger sequence 3 count value, send this command:

   :TRIGger:SEQuence3:COUNt?

After receiving the query command, a read command returns the current sequence count value in <NR1> format. For example, if the count value was previously set to the default value and the signal generator is queried for the current count value, it should return **1**, indicating that trigger sequence 3 goes from the armed state to the triggered state only once, and, after being triggered, returns to the idle state.

TRIGger**:SEQ**uence<n>**:ECO**unt

Default: SEQuence**1**
Range: 1 to 32767



The event-count variable controls the number of occurrences of a particular event that are required to move the signal generator from the armed state to the triggered state. When the signal generator 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 causes the signal generator to move from the armed state to the triggered state. However, the signal generator 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 **ECO**unt variable.

> **NOTE**
>
> The parameters **:SEQ**uence**1**, **:SEQ**uence, and **:STAR**t are equivalent and can be used interchangeably. Their use is also optional, since, if no sequence is specified, **:SEQ**uence**1** is assumed.

Examples:

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

   :TRIGger**:SEQ**uence**2:ECO**unt **10**

2. To program the signal generator 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 this command:

   :TRIGger**:SEQ**uence**4:ECO**unt **12**

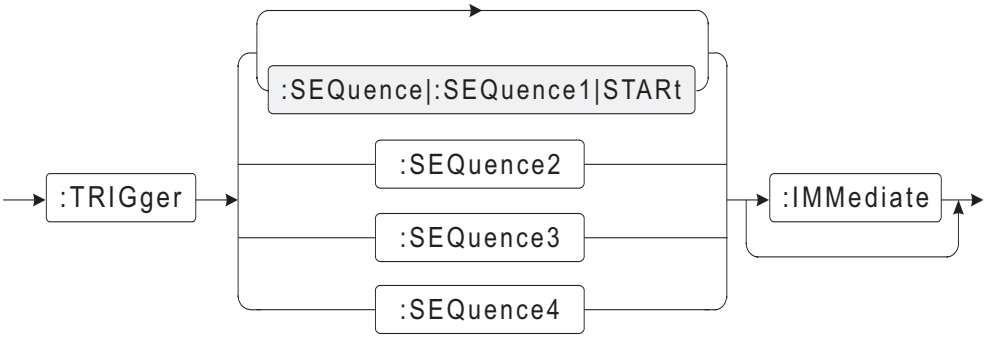
3. To query the signal generator for the current event count for trigger sequence 4, send this command:

   :TRIGger**:SEQ**uence**4?**

   After the signal generator receives the query command, a read command returns the current event count value in <NR1> format. For example, if the count was previously set to ten, and the signal generator is queried for the current count, it returns **10**, indicating that trigger sequence 4 will wait for ten events before sourcing a trigger.

TRIGger**:SEQ**uence<n>**:IMM**ediate

Default: SEQuence**1**



This is the master trigger-control command. It immediately triggers a particular sequence (**SEQ1**...**SEQ4**). When the signal generator receives this command, it ignores the event count variable (**ECO**unt) and immediately triggers the selected sequence, providing the sequence is already armed. The variable selects the sequence to trigger. If the variable is omitted, it defaults to 1.

> **NOTE**
>
> The parameters **:SEQ**uence**1**, **:SEQ**uence, and **:STAR**t are equivalent and can be used interchangeably. Their use is also optional, since, if no sequence is specified, **:SEQ**uence**1** is assumed. Use of the parameter **:IMM**ediate is also optional.

Examples:

1. To trigger sequence 1, send this command:

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

   or

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

   or just

   :TRIG

2. To selectively trigger sequence 2, send this command:

   :TRIGger**:SEQ**uence**2:IMM**ediate

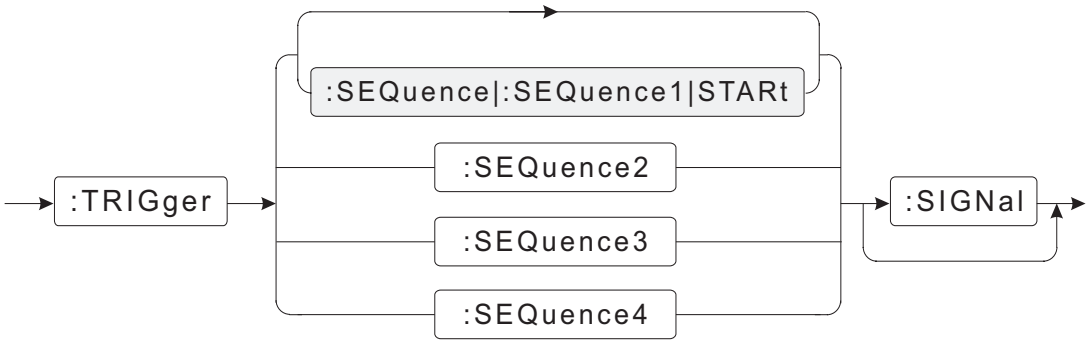
   or

   :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**:SEQ**uence<n>**:SIGN**al

Default: SEQuence**1**



This command initiates an immediate trigger, acting as if the selected **TRIG:SOUR**ce signal/event occurred. The **:TRIG**ger**:SEQ**uence<n>**:SOUR**ce command selects between a BUS trigger, one of the two **ECLT**rg lines, or one of the eight **TTLT**rg 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.

> **NOTE**
>
> The parameters **:SEQ**uence**1**, **:SEQ**uence, and **:STAR**t are equivalent and can be used interchangeably. Their use is also optional, since, if no sequence is specified, **:SEQ**uence**1** is assumed.

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**, a simulated BUS trigger is issued by sending the command:

   :TRIG:SEQ2:SIGNal

2. If the **ECLT**rg**1** was previously selected to serve as the trigger source for trigger sequence 3 using the command **:TRIG:SEQ3:SOUR ECLT**rg**1**, a simulated **ECLT**rg**1** trigger is issued by sending the command:

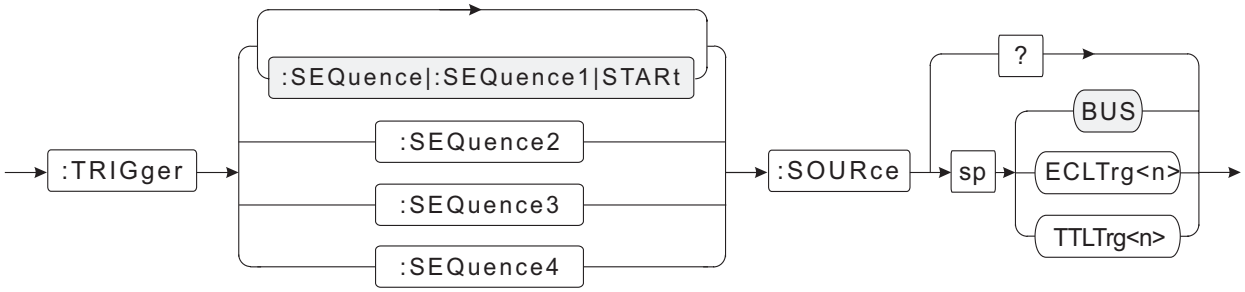
   :TRIG:SEQ3:SIGNal

Related Command:

:TRIGger**:SEQ**uence<n>**:SOUR**ce

TRIGger**:SEQ**uence<n>**:SOUR**ce

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 generated by either the IEEE 488.2 common command ***TRG** or the VXI word serial trigger command. There are 10 triggers in addition to the **BUS** trigger available, including **TTLTrg0** through **TTLTrg7** and **ECLTrg0** and **ECLTrg1**.

NOTE

The parameters **:SEQ**uence**1**, **:SEQ**uence, and **:STAR**t are equivalent and can be used interchangeably. Their use is also optional, since, if no sequence is specified, **:SEQ**uence**1** is assumed.

Examples:

1. To set the signal generator to trigger-sequence one of the VXIbus **ECLTrg1** trigger source, send this command:

   :TRIGger:SEQuence**1:SOUR**ce **ECLTrg1**

2. To set the signal generator to trigger-sequence one of the VXIbus **TTLTrg4** trigger source, send this command:

   :TRIGger**:SEQ**uence**1:SOUR**ce **TTLTrg4**

3. To query the signal generator for the trigger source of sequence 3, send this command:

   :TRIGger**:SEQ**uence**3:SOUR**ce**?**

   After the signal generator receives the query command, a read command returns the sequence 3 trigger source in <char> format. For example, if the trigger source was previously set to **ECLTrg0**, and the signal generator is queried for the current trigger source, it should returns **ECLT0** indicating the source of triggers for trigger sequence 3 (**TRIG:SEQ3**) is **ECLTrg0** from the VXI backplane.

UNIT SUBSYSTEM

The Unit subsystem controls default units associated with data entries.

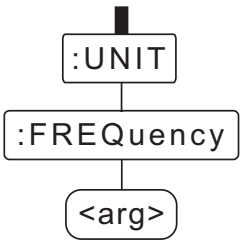
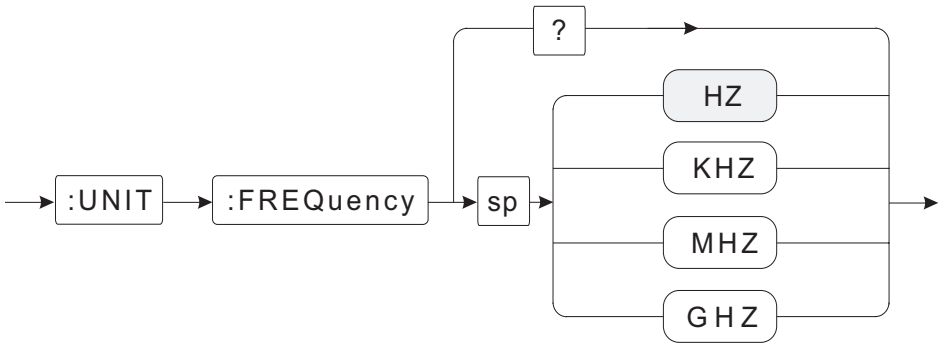


Figure 4-7. Unit Subsystem Command Tree

UNIT:FREQuency



Default: Hz

This command specifies a default metric suffix associated with frequency units. The unit selected is used with all frequency responses from the signal generator and applies to all frequency entries without an explicit suffix.

Examples:

1. To change the default frequency unit to GHz, send this command:

   :UNIT:FREQuency **GHZ**

2. To query the signal generator for the current frequency unit, send this command:

   :UNIT:FREQuency**?**

   After the signal generator receives the query command, a read command returns the current frequency unit in the <char> format. For example, if the frequency unit was previously set to MHz and the signal generator is queried for the current frequency unit, it should return **MHZ**, indicating the current frequency unit is set to MHz.

5

OPERATION AND PROGRAMMING EXAMPLES

INTRODUCTION

This section provides general information about using the Phase Matrix 1140B and 1141B VXIbus Synthesized Signal Generators in a VXIbus system. Specifically, information is provided on the VXIbus support hardware required to operate the signal generator in a system, and a variety of programming examples.

VXIBUS SYSTEM CONFIGURATIONS

All VXIbus systems have 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, dc power, and forced-air cooling. The system controller is a computer that provides the user interface for the system and a platform for developing and running the system software. It can be a separate, external computer, or it can be integrated with the Slot 0 device. The Slot 0 device provides backplane clocks, configuration and synchronization signals, and an interface to an external system controller. Typical VXIbus systems are shown in Figure 5-1 and Figure 5-2 (on pages 5-2 and 5-3).

EXTERNAL COMPUTER

In the system shown in Figure 5-1, a standard, external computer is used. System software can be developed in any operating system compatible with the computer (e.g., DOS, Windows, UNIX), and communication can be via any interface supported by both the computer and the Slot 0 device (e.g., GPIB, USB, FireWire, RS-232). The variety of hardware, software, and interfaces available leads to the virtually unlimited versatility of VXIbus test-equipment systems. The Slot 0 Resource Manager routes commands from the sytem controller over the VXI backplane to the addressed instrument and returns data from individual instruments in the VXIbus chassis to the system controller.

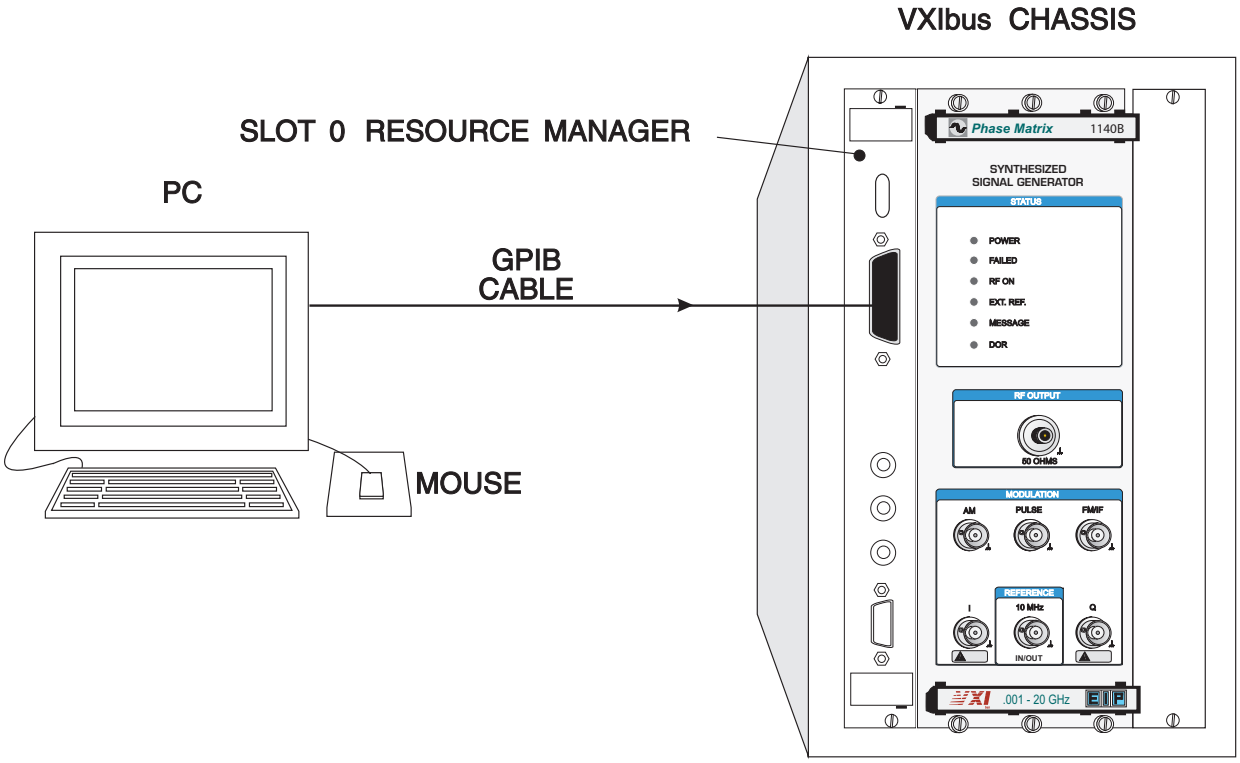


Figure 5-1. Typical VXIbus System Using an External Computer

EMBEDDED COMPUTER

In the system shown in Figure 5-2, an embedded computer installed in Slot 0 of the VXIbus chassis combines the functions of the system controller and the Slot 0 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 control lines. Typically, the embedded computer includes a GPIB or other communications port to control standard GPIB instruments, or in larger systems, to communicate with and control Slot 0 devices in additional VXIbus chassis.

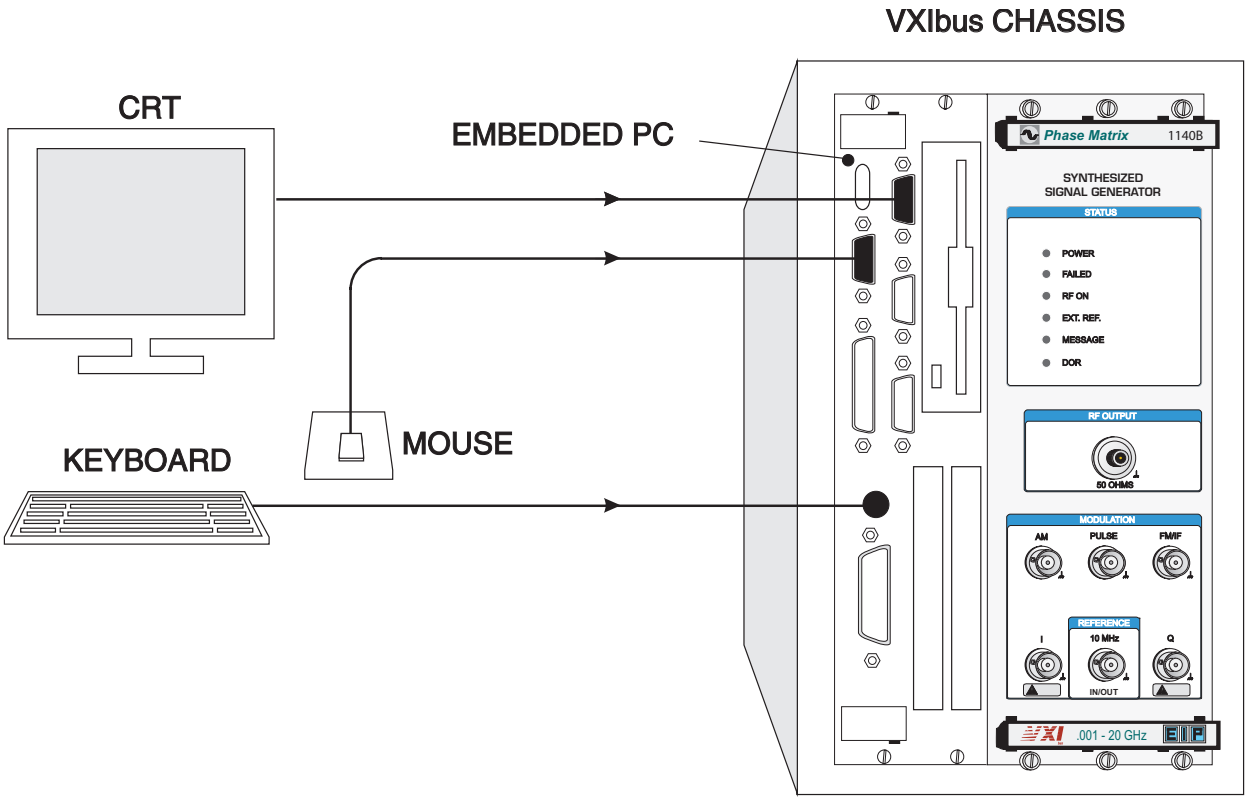


Figure 5-2. Typical VXIbus System Using an Embedded Computer

SYSTEM CONFIGURATION

VXIbus chassis are available in varying physical sizes, cooling capacities, and dc-output power. However, a chassis physically large enough to house the modules may lack the dc-power or cooling capacity to run the system modules. To ensure the chassis' adequacy, refer to the peak-current requirements of each of the system modules, and verify that the total load on each of the VXIbus power supplies is within the specified capabilities of the chassis. Normally, if the chassis has adequate dc-power capacity, it can provide sufficient cooling, since the required cooling is directly related to the instruments' power consumption. Nonetheless, it is a good idea to calculate the cooling requirements vs. cooling capacity, especially if the power requirements of the instruments

is close to the chassis capacity and if the system is to be operated near the upper limits of the instruments' rated temperature range.

In many chassis, proper air flow through the modules may not occur unless the chassis is full, since any open space provides a low-resistance path for the cooling air. To alleviate this problem, baffles should be installed to restrict airflow through any unused slots in the chassis. This is not a concern, however, in those chassis which utilize a plenum system to ensure adequate airflow through each slot regardless of chassis loading.

Another consideration in choosing a VXIbus chassis for RF and microwave applications is the RF characteristics of the chassis. Some chassis are designed specifically for RF and Microwave applications. These chassis offer increased intermodule shielding, low EMI, and low-noise (linear) power supplies. This helps to ensure that sensitive equipment can operate without interference from the chassis or the external environment.

These special RF chassis are not required for all microwave systems. The 1140B and 1141B Microwave Signal Generators will fully meet all performance specifications in any commercially available VXIbus chassis having adequate power capability. 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. Hence, a special RF chassis should be considered to obtain optimum system performance.

Once an appropriate VXIbus chassis is selected, system configuration consists only of setting the hardware address and any other configuration switches on the modules and loading any required system configuration data into the firmware (consult the manual for each of the modules).

SAMPLE PROGRAMS

TIMED FREQUENCY SWEEP

This example programs the signal generator to perform a single, timed-frequency, stepped sweep from 4 to 5 GHz in 10 MHz steps. The signal generator remains at each step in the stepped sweep for one second (dwell). The program listing for the Timed Frequency Sweep is shown below. All programs in Section 5 are 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:imm on
:trig:seq2:immediate
```

To provide more insight into programming the signal generator, each of the steps in the preceding program are discussed separately below and on the following pages.

The program begins with this command:

:init:cont off; :abort; *rst

This line comprises three commands separated by semicolons. This group of commands initializes the signal generator. The first command, **:init:cont off**, causes the signal generator 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 an IEEE 488.2 Common Command that sets all user-programmable variables to default values. Taken together, these three commands set the signal generator to a known state.

Line two of the program is:

:unit:freq ghz; *cls; :status:preset

This line also comprises 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 an IEEE 488.2 Common Command that 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 zeros,

2) it clears the Operational-enable register to all zeros, 3) it clears the Operational-Negative-transition filter to all zeros, 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 signal generator 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 cause the signal generator 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 starts the frequency sweep. For this example, a trigger from trigger sequence 2 starts 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** and **;stop 5**, set the frequency-sweep range. The frequency units were set to GHz in the second line, so the start frequency is 4 GHz, and the stop frequency is 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 signal generator. The first mode is **CW** (or **FIX**ed); it is the mode to use when setting a fixed frequency. The second mode is **LIST**. The list mode couples frequency control to a triggered list. The third mode is **SWE**ep**1**; 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 signal generator 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 signal generator normally takes a single frequency step and then returns to the idle state. With initiate-continuously on, the signal generator bypasses the idle state, falls through to the armed state, and waits for the trigger from the dwell timer. When the trigger occurs, the signal generator steps to the next frequency. This process continues until the signal generator has stepped through the entire range. If the sweep count is 1 when the signal generator reaches the stop frequency, it stops.

Line ten of the program is:

:trig:seq2:immediate

This command immediately triggers sequence 2, causing the signal generator to start the frequency sweep.

TIMED-POWER SWEEP

This example program commands the signal generator to perform a single, timed, stepped power sweep from 0 to 8 dBm in 1 dB steps at a frequency of 5 GHz. The signal generator remaina (dwells) 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 this program initialize the signal generator to a known state. Frequency sweeps are identified by **sweep1**; power sweeps are identified by **sweep2**. Line 3 of the program sets a

sweep-count variable of 1 for power sweeps. This means that the power sweep is performed only one time. The fourth line sets 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 and the power-step size. The seventh line couples output-power control to **sweep2**. The remaining three lines are identical to those in the first program on page 5-5 and serve the same purposes.

FREQUENCY-AND-POWER SWEEP

This program combines elements from the first two programs to command the signal generator to perform a combination frequencyand-power sweep. Specifically, the program commands the signal generator to sweep from 5 to 10 GHz in 1 GHz steps 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 is achieved via **VXI TTLTrg1**. Each change of the signal generator'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 SIGNAL GENERATOR

```
: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 the signal generator 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 signal generator to make a frequency step.

INITIALIZE SIGNAL GENERATOR

```
: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
:sweep: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 signal generator in the desired configuration prior to sending the list data because the signal generator stores setup information along with the other variables at each point vector in the list. For example, if the **:output on** command is changed to **:output off**, each point vector is stored with the output RF turned off.

INITIALIZE SIGNAL GENERATOR

```
: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
```

SET UP 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
```

A

VXIbus

INTRODUCTION

This VXIbus overview is from the VXIbus Specification, revision 1.3.

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 æ12 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.

B

COMMMAND SUMMARY

COMMON COMMANDS, IEEE 488.2

Command	Description	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

Keyword	Arguments
ABORt	
CALibration	
[:ALL]?	
:ATTenuator[:STATe]	ON|OFF|1|0
:POWer[:STATe]	ON|OFF|1|0

Keyword	Arguments
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
[SOURce]	
:AM	
[:DEPTh]	<integer>|DEFault| MIN|MAX|UP|DOWN
:STEP[:INCRement]	< integer>
:MODE	CW|FIXed|LIST
:STATe	ON|OFF|1|0
:CM	
:FREQuency	<NRf>
:MODE	CW|FIXed|LIST
:STATe	ON|OFF|1|0
:DM	
:MODE	CW|FIXed|LIST
:STATE	ON|OFF|1|0
:FM	
:MODE	CW|FIXed|LIST
:STATE	
:FREQuency	
:CENTer	<nv>
[:CW|FIXed]	<nv>|DEFault|MIN|MAX|UP|DOWN
:STEP[:INCRement]	<nv>
:FTUNe	<nv>|DEFault
:MODE	CW|FIXed|LIST|SWEep|SWEep1
:SPAN	<nv>
:FULL	
:STARt	<nv>

Keyword	Arguments
[SOURce] (continued)	
:FREQuency (continued)	
:STOP	<nv>
:LIST	
:COUNt	<nv>
:DIRection	UP|DOWN
:PVECtor	<char>
:DWELl	<NR3>
:LINK	<char>
:MODECW|FIXed|LIST	
:POINts?	
:STARt	<char>
:POWer	
:ALC[:STATe]	ON|OFF|1|0
:ATTenuation	<NRf>
:AUTO	ON|OFF|1|0
:CENTer	<nv>
:DAC	<NRf>
[:LEVel]	
[:IMMediate][:AMPlitude]	<nv>
:STEP[:INCRement]	<NRf>
:MODE	CW|FIXed|LIST|SWEep|SWEep1
:SPAN	<nv>
:FULL	
:STARt	<nv>
:STOP	<nv>
:PULM	
:MODE	CW|FIXed|LIST
[:STATe]	ON|OFF|1|0
:ROSCillator	
:SOURce	INTernal|EXTernal
:SWEep<n> (n = 1|2)	
:COUNt	<nv>
:DIRection	UP|DOWN
:DWELl	<nv>
:LINK	<char>
:GENeration	STEP
:SPACing	LINear
:STARt	SWE1 = <char> SWE2 = TRIG:SEQ3
:STEP	
:FREQuency (SWE1 Only)	<nv>
:POWer (SWE2 Only)	<nv>

Keyword	Arguments
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

C

FUNCTIONAL BLOCK DIAGRAMS

FUNCTIONAL BLOCK DIAGRAMS

This section provides the functional block diagrams for the Phase Matrix 1140B and 1141B VXIbus Synthesized Signal Generators.

This Page Intentionally Left Blank

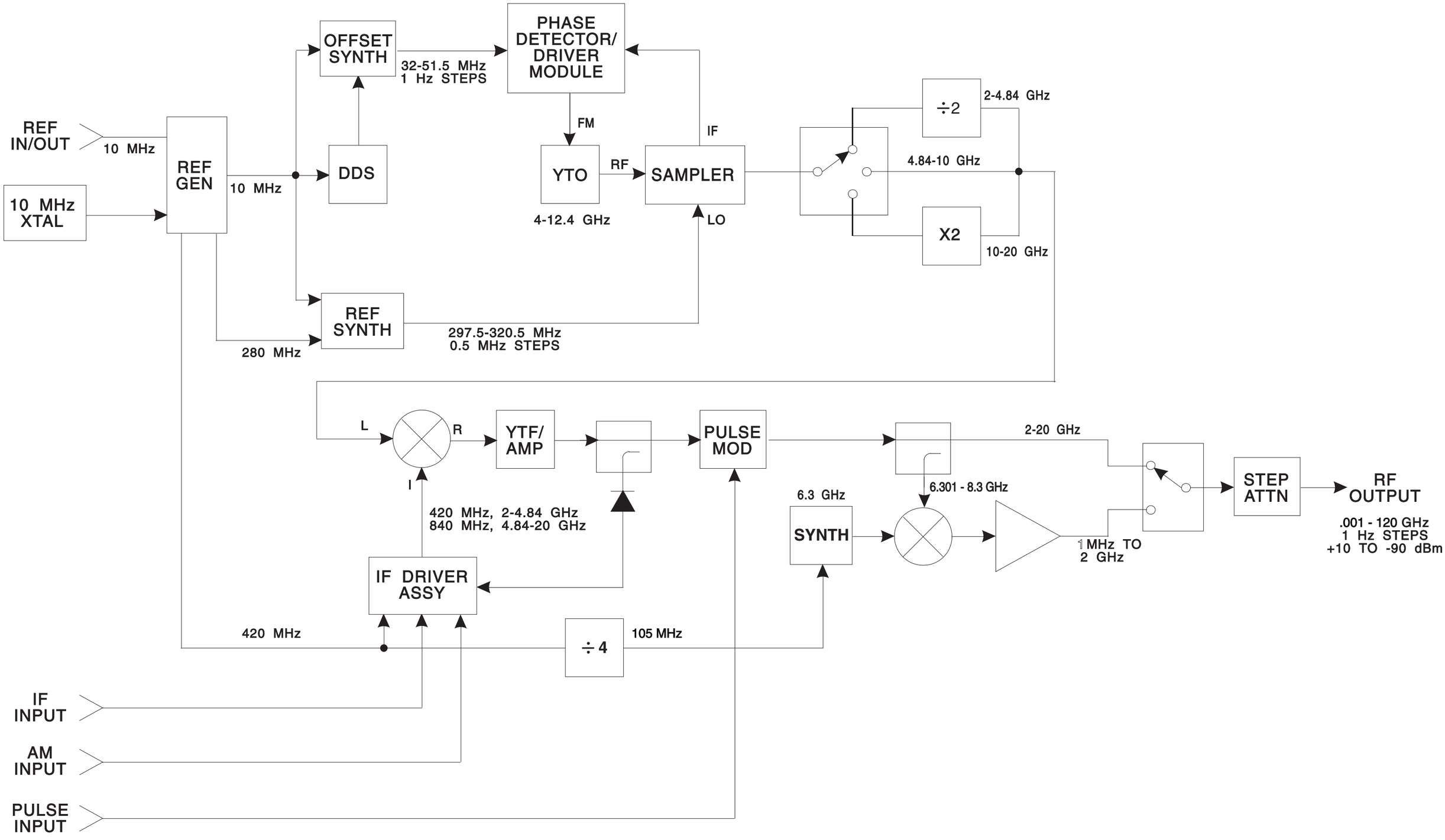


Figure C-1. Model 1140B Functional Block Diagram

This Page Intentionally Left Blank

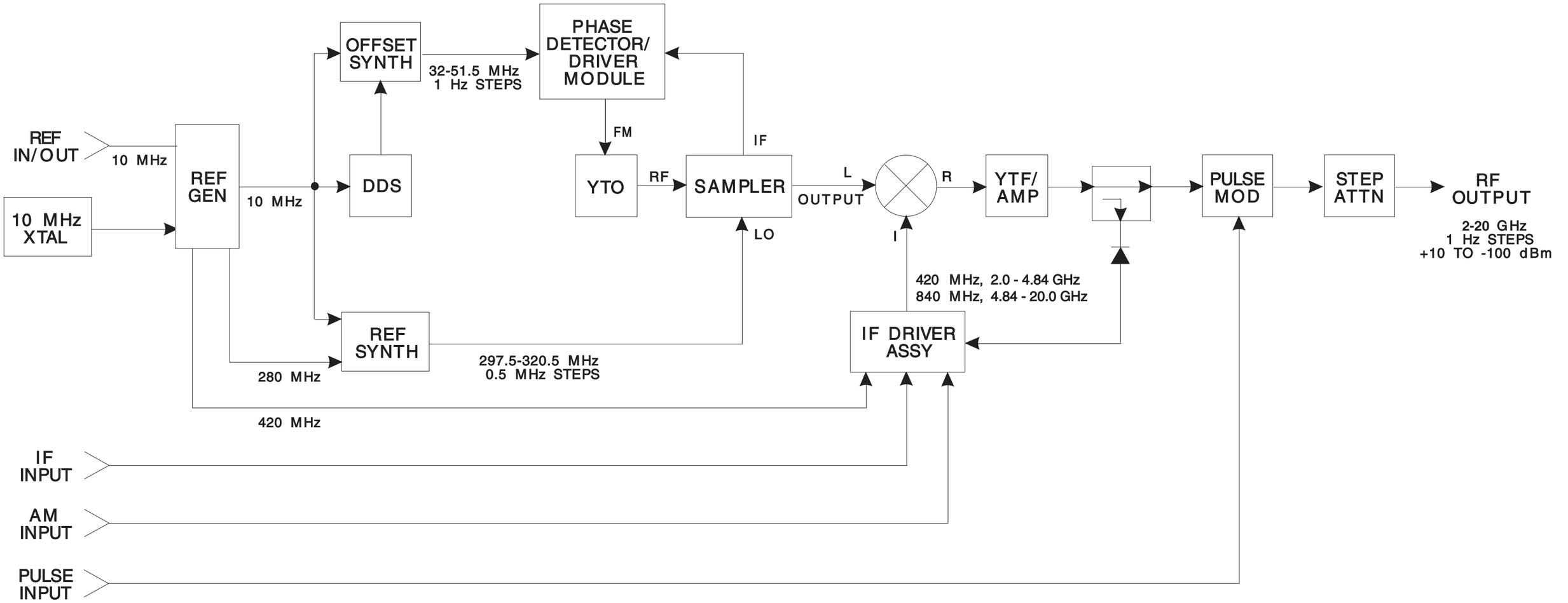


Figure C-2. Model 1141B Functional Block Diagram

This Page Intentionally Left Blank

D

COMMMAND TREE

INTRODUCTION

This section provides a complete command tree showing all of the programming commands for the Phase Matrix 1140B and 1141B signal generators.

This Page Intentionally Left Blank

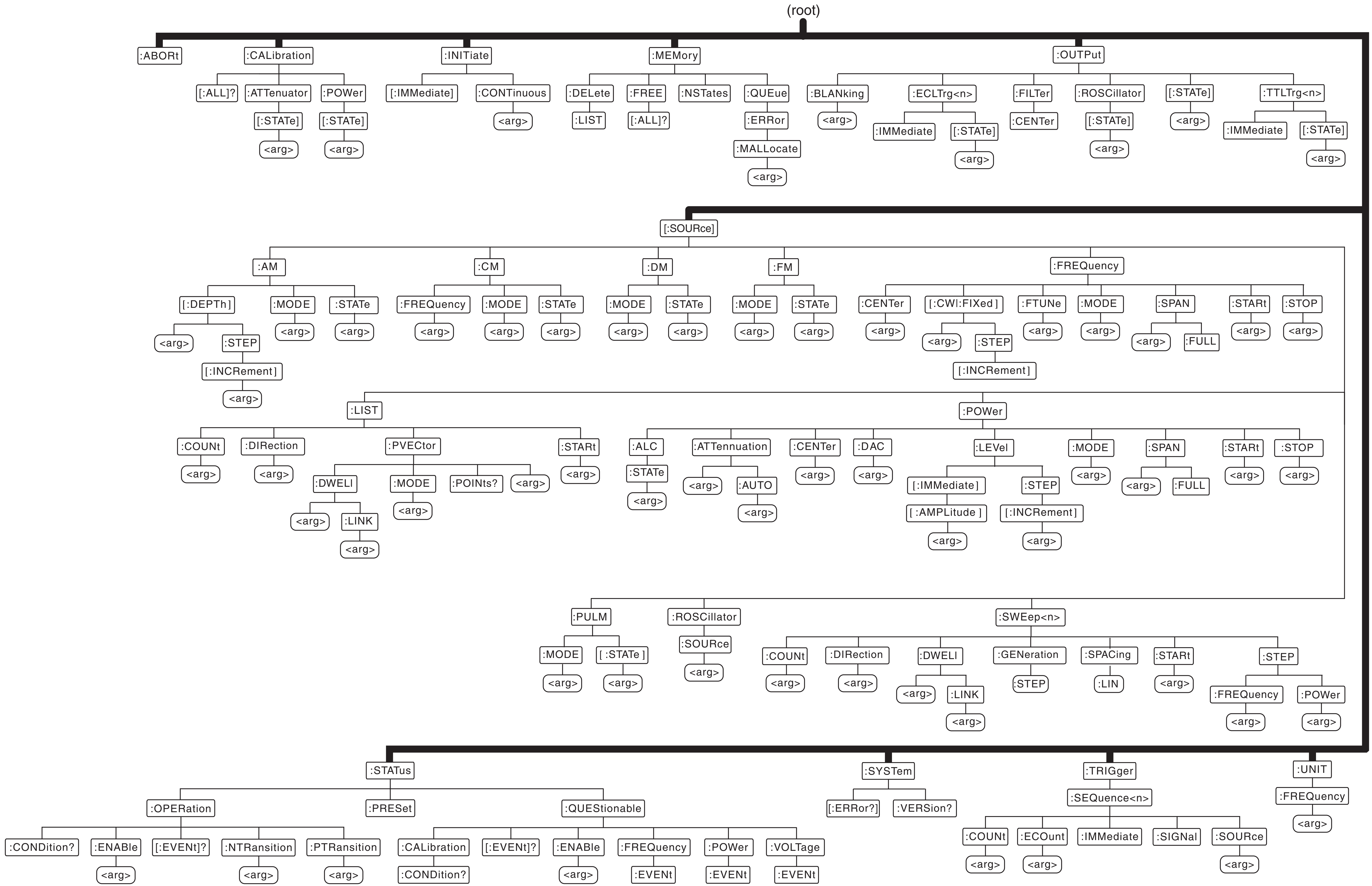


Figure D-1. 1140B & 1141B Command Tree

This Page Intentionally Left Blank

E

POWER SEARCH

INTRODUCTION

Under most conditions, the synthesizer is operated with Automatic Level Control (ALC) enabled. This ensures a constant level of power across the frequency band and under conditions of changing temperature. There are, however, conditions under which it becomes necessary to disable the ALC and control the output power manually. Chief among these are those applications where low-frequency amplitude modulation is accomplished through the synthesizer's IF port (see Block Diagram, Figure C-1 or Figure C-2, Appendix C).

In basic 1140B and 1141B units, those in which Option 02 (Frequency Modulation) or Option 05 (Digital Modulation, also known as I/Q Modulation) is not installed, a front-panel Complex Modulation input is connected to the IF port. If, however, Option 02 or Option 05 is installed, the IF port of the synthesizer is connected to an internal FM or I/Q Modulator, and the front-panel jack serves as the FM signal input. In any case, the synthesizer translates the modulating signal to the nominal output frequency and power level to which the unit has been commanded. When modulation via the IF port (CM, FM, or DM) is turned off, an internally generated IF of 420 MHz or 840 MHz, depending on the band, is automatically applied to the IF port.

It should be noted that Pulse Modulation and Amplitude Modulation are accomplished by other circuitry, are unaffected by the ALC, and are totally independent of and may be enabled concurrently with modulation via the IF port.

The architecture of the synthesizer is such that the ALC maintains a constant output amplitude despite any variances in the amplitude of the signal, whether it is an externally supplied Complex Modulation signal or an internally generated I/Q signal. Hence, if these signals are Amplitude Modulated at a frequency less than the ALC bandwidth, the ALC tends to cancel the modulation. Under these conditions, it is necessary to disable the ALC and control the output signal level manually.

The output power of the synthesizer is controlled by a voltage-controlled attenuator. The control voltage is called the ALC MONITOR voltage and consists of the sum of the ALC voltage and the POWER DAC (Digital to Analog Converter) output. Since the contribution of the ALC voltage

to this total varies considerably across the power range, turning off the ALC can produce a radical shift in output power level. The synthesizer is calibrated only with the ALC enabled, and a lookup table pairing power settings to Power DAC values is created. Hence, the correlation between power settings and output power is totally invalid with the ALC turned off.

It is therefore necessary, when the ALC is disabled, to control the output power through the entry of Power DAC settings. Obviously, this can be done by connecting a power sensor to the output of the synthesizer and adjusting the Power DAC value until the desired output is obtained.

The following Power Search routine, utilizing commands issued by the host computer and some simple processing, can also be used. No additional equipment or sensors are required.

POWER SEARCH THEORY

Since, at any given frequency (and such other conditions as ambient operating temperature), there is a one-to-one correlation between the ALC MONITOR voltage and power output, calibrating the output power is simply a matter of reading the ALC MONITOR voltage at the desired power setting with the ALC enabled and calculating a Power DAC value that returns the ALC MONITOR to that value when the ALC is disabled.

To that end, one of two approaches can be used. When the synthesizer is not modulated via the IF port, the internally generated 420 MHz or 840 MHz IF is applied to the IF port and can be used in the calibration of the synthesizer. This approach avoids the need for any special externally applied signal. However, the accuracy of the calibration is dependent upon the manufacturing tolerances of the internal signal levels.

A more precise calibration can be obtained by calibrating the unit with the modulating signal to be used. In the case of a Complex Modulation signal, this requires generation of a CW signal of the nominal frequency and amplitude of the complex signal to be applied during the calibration routine. In the case of Digital (I/Q) Modulation, a DC level of 0.4 volts must be applied to either the I or Q input (the other input may be set to 0 volts or simply left open).

The Power Search routine works by reading the ALC MONITOR and POWER DAC values with the ALC on, then turning the ALC off and calculating a new POWER DAC value to return the ALC MONITOR to its original value.

In the Power Search routine outlined on the following pages, suggested values for the constant K = 10 and L = 0.02 are given. These values can be adjusted by the user to better suit the application. The Power Search routine is a reiterative loop. The value of K influences the rate at which the routine converges on the final value. If the value of K is too small, it takes more cycles of the software loop to arrive at the final value. If the value of K is too large, the result overshoots and never settles within the acceptable range of the final value. The value of L sets the acceptable tolerance of the final value. In general, setting a smaller value of L requires the use of a smaller value of K.

POWER SEARCH ROUTINE - COMPLEX MODULATION (Units with neither option 02 or option 05)

A. Initial setup of synthesizer:

1. Set the *FREQUENCY* to the desired output frequency.

   :FREQ:CW XX.XX GHZ

   (*Where XX.XX is the nominal output frequency in GHz.*)

2. Set the *POWER* to the desired amplitude.

   :POW XX.X

   (*Where XX.X is the nominal amplitude in dBm.*)

NOTE

The nominal frequency and amplitude of a Complex Modulation signal is defined as the instantaneous value of frequency and amplitude which will result in an instantaneous output frequency and amplitude of the synthesizer equal to the frequency and amplitude to which the synthesizer has been set by the :SOURce:FREQuency and :SOURce:POWer commands.

B. Preliminary:

1. Turn the *RF OUTPUT* off.

   :OUTP:STAT 0

2. Turn *COMPLEX MODULATION* on.

   :CM:STAT 1

3. Apply an unmodulated (CW) signal, the frequency and amplitude of which is equal to the nominal frequency and amplitude of the Complex Modulation signal, to the **IF/FM** jack.

   The nominal frequency must also be communicated to the synthesizer using the **:SOURce:CM:FREQuency** command.

4. Turn the *ALC* on.

   :POW:ALC 1

C. Read the *POWER DAC* value. Store as $D = D_0$.

:POW:DAC?

D. Read the *ALC MONITOR* value. Store as $A = A_0$.

:DIAG:MEAS? 9

E. Turn the *ALC* off.

:POW:ALC 0

F. Perform the following routine:

1. Read the *ALC MONITOR* value. Store as $P = P_0$.

:DIAG:MEAS? 9

2. Calculate $\Delta_A = A - P$.
3. Test: Is $|\Delta_A| < L$, where L is the minimum acceptable limit of $|\Delta_A|$ (suggested value: 0.02).
   a. If "yes," go to G.
   b. If "no," continue.
4. Calculate $\Delta_D = K\Delta_A$, where K is a constant (suggest K = 10).
5. Calculate $D_1 = D + \Delta_D$.
6. Set $D = D_1$.
7. Set the *POWER DAC* value to D.
8. Go to 1.

G. Connect the normal Complex Modulation signal to the synthesizer.

H. Turn the *RF OUTPUT* on.

:OUTP:STAT 1

NOTE

If a nominal CW signal is not available, an alternative, less accurate method of setting the power may be used. The foregoing procedure is modified as follows:

Step B.2. Turn *COMPLEX MODULATION* off.

:CM:STAT 0

Step B.3. Disregard. (Any signal applied to the **IF/FM** jack is ignored.)

Step G. Add the following:

1. Turn *COMPLEX MODULATION* on.

   :CM:STAT 1

2. Enter the nominal Complex Modulation frequency:

   :CM:FREQ XXX MHZ

   (Where *XXX* is the nominal complex-modulation frequency in MHz.)

Use of this alternative procedure results in an output-power error, the magnitude of which is determined by the sum of the manufacturing tolerances of several signals internal to the synthesizer. Typically, the deviation from the nominal power value is less than 2 dBm.

POWER SEARCH ROUTINE - DIGITAL (I/Q MODULATION) (Option 05 Units)

A. Preliminary:

1. Turn the *RF OUTPUT* off.

   :OUTP:STAT 0

2. Turn *DIGITAL (I/Q) MODULATION* on.

   :DM:STAT 1

3. Apply 0.4 vdc (either positive or negative) to either the **I** or **Q** input.
4. Turn *ALC* on.

   :POW:ALC 1

B. Set *POWER* to the desired amplitude.

   :POW XX.X

   (Where *XX.X* is the amplitude in dBm.)

C. Read the *POWER DAC* value. Store as $D = D_0$.

   :POW:DAC?

D. Read the *ALC MONITOR* value. Store as $A = A_0$.

:DIAG:MEAS? 9

E. Set the *ALC* to **OFF**.

:POW:ALC 0

F. Perform the following routine:

1. Read the *ALC MONITOR* value. Store as $P = P_0$.

:DIAG:MEAS? 9

2. Calculate $\Delta_A = A - P$.

3. Test: Is $|\Delta_A| < L$, where L is the minimum acceptable limit of $|\Delta_A|$ (suggested value: 0.02).
   a. If "yes," go to G.
   b. If "no," continue.
4. Calculate $\Delta_D = K\Delta_A$, where K is a constant (suggest K = 10).
5. Calculate $D_1 = D + \Delta_D$.
6. Set $D = D_1$.
7. Set the *POWER DAC* value to D.
8. Go to 1.

G. Connect the normal I/Q signal to the generator.

H. Turn the *RF OUTPUT* on.

:OUTP:STAT 1

NOTE

If a nominal 0.4 volt signal is not available, an alternative, less accurate method of setting the power may be used. The foregoing procedure is modified as follows:

Step A.2. Turn *DIGITAL MODULATION* off.

:DM:STAT 0

Step A.3. Disregard. (Any signal applied to the **I** or **Q** jacks is ignored.)

Step G. Add the following:

1. Turn DIGITAL MODULATION on.

:DM:STAT 1

Use of this alternative procedure results in an output-power error, the magnitude of which is determined by the sum of the manufacturing tolerances of several signals internal to the synthesizer. Typically, the deviation from the nominal power value is less than 2 dBm.

This Page Intentionally Left Blank

Artisan Technology Group is an independent supplier of quality pre-owned equipment

Gold-standard solutions

Extend the life of your critical industrial, commercial, and military systems with our superior service and support.

We buy equipment

Planning to upgrade your current equipment? Have surplus equipment taking up shelf space? We'll give it a new home.

Learn more!

Visit us at **artisantg.com** for more info on price quotes, drivers, technical specifications, manuals, and documentation.

Artisan Scientific Corporation dba Artisan Technology Group is not an affiliate, representative, or authorized distributor for any manufacturer listed herein.

We're here to make your life easier. How can we help you today?

(217) 352-9330 | sales@artisantg.com | artisantg.com

ARTISAN® TECHNOLOGY GROUP