WAVETEK

OPERATOR'S & MAINTENANCE MANUAL

Model 1385
50 MHz VXIbus Arbitrary Waveform Synthesizer

Wavetek Ltd.
Hurricane Way,
Norwich Airport Industrial Estate
Norwich
Norfolk NR6 6JB
UK

Tel: 44 1603 404 824
Fax: 44 1603 483 670

Manual Issue 1, August 1996
Manual Part Number: 850309-1.0

Wavetek Corporation
Standard Warranty Policy

Wavetek warrants that all Products manufactured or procured by Wavetek conform to Wavetek's published specifications and are free from defects in materials and workmanship for a period of one (1) year from the date of delivery to the original Buyer, when used under normal operating conditions and within the service conditions for which they were designed. This warranty is not transferrable and does not apply to used or demonstration products.

The obligation of Wavetek arising from a Warranty claim shall be limited to repairing, or at its option, replacing without charge, any assembly or component (except batteries) which in Wavetek's sole opinion proves to be defective within the scope of the Warranty. In the event Wavetek is not able to modify, repair or replace nonconforming defective parts or components to a condition as warranted within a reasonable time after receipt thereof, Buyers shall receive credit in the amount of the original invoiced price of the product.

Wavetek must be notified in writing of the defect or nonconformity within the Warranty period and the affected Product returned to Wavetek's factory, designated Service Provider, or Authorized Service Center within thirty (30) days after discovery of such defect or nonconformity. Buyer shall prepay shipping charges and insurance for Products returned to Wavetek or its designated Service Provider for warranty service. Wavetek or its designated Service Provider shall pay costs for return of Products to Buyer.

Wavetek shall have no responsibility for any defect or damage caused by improper storage, improper installation, unauthorized modification, misuse, neglect, inadequate maintenance, accident or for any Product which has been repaired or altered by anyone other than Wavetek or its authorized representative or not in accordance with instructions furnished by Wavetek.

The Warranty described above is Buyer's sole and exclusive remedy and no other warranty, whether written or oral, expressed or implied by statute or course of dealing shall apply. Wavetek specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. No statement, representation, agreement, or understanding, oral or written, made by an agent, distributor, or employee of Wavetek, which is not contained in the foregoing Warranty will be binding upon Wavetek, unless made in writing and executed by an authorized representative of Wavetek. Under no circumstances shall Wavetek be liable for any direct, indirect, special, incidental, or consequential damages, expenses, or losses, including loss of profits, based on contract, tort, or any other legal theory.

April 1, 1994

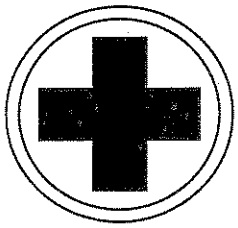
SAFETY WARNING!
If this equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

CE

This product complies with the requirements of the following European Community Directives:
89/336/EEC (Electromagnetic Compatibility) and **73/23/EEC (Low Voltage)**
as amended by **93/68/EEC (CE Marking).**

However, noisy or intense electromagnetic fields in the vicinity of the equipment can disturb the measurement circuit. Users should exercise caution and use appropriate connection and cabling configurations to avoid misleading results when making precision measurements in the presence of electromagnetic interference.

SAFETY FIRST



PROTECT YOURSELF. Follow these precautions:

- Don't touch the outputs of the instrument or any exposed test wiring carrying the output signals. This instrument can generate hazardous voltages and currents.
- Don't bypass the VXI chassis' power cord's ground lead with two-wire extension cords or plug adaptors.
- Don't disconnect the green and yellow safety-earth-ground wire that connects the ground lug of the VXI chassis power receptacle to the chassis ground terminal (marked with ⏚ or ⚠).
- Don't hold your eyes extremely close to an rf output for a long time. The normally nonhazardous low-power rf energy generated by the instrument could possibly cause eye injury.
- Don't energize the VXI chassis until directed to by the installation instructions.
- Don't repair the instrument unless you are a qualified electronics technician and know how to work with hazardous voltages.
- Pay attention to the **WARNING** statements. They point out situations that can cause injury or death.
- Pay attention to the CAUTION statements. They point out situations that can cause equipment damage.

CONTENTS

CONTENTS (Continued)

SECTION 3 OPERATION

CONTENTS (Continued)

TABLES

ILLUSTRATIONS

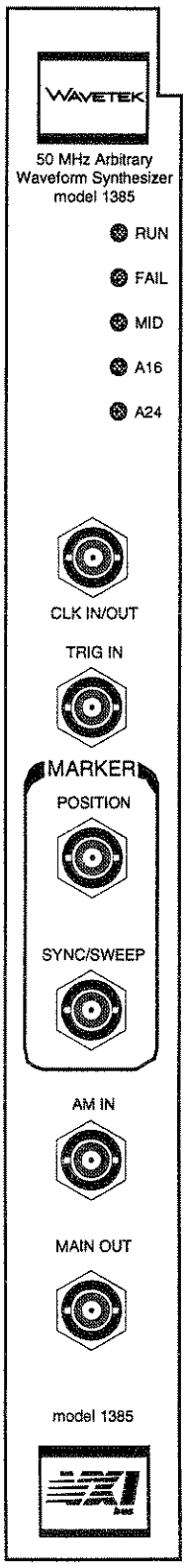


Figure 1-1. Model 1385 50 MHz VXIbus Arbitrary Waveform Synthesizer

Specifications

Section 1

1.1 THE MODEL 1385

The MODEL 1385 is a high performance Synthesized Arbitrary Waveform Generator (ARB) with the following main features:

- Isolated Analog Output
- Up to 50 MHz Sampling Frequency
- 12 Bit Vertical Resolution
- 128K points (512K optional) Horizontal Resolution
- Intermodule Triggering, Summing and Phase Control
- Waveform Linking and Looping
- 64K bytes Shared Memory for fast data transfer
- Auxilliary Analog Output
- SCPI Compatible Command Language
- Single Slot, C-Size VXIbus Module

The waveform synthesizer can be programmed to produce standard waveforms in the frequency range of 1 mHz to 20 MHz; or arbitrary waveforms from 5 points minimum to 128K (512) maximum sampled at frequencies from 125 mHz to 50 MHz. Additionally, a Clock Output is provided from 125 mHz to 100 MHz.

Waveforms can be created by selection of the standard waveforms, drawing waveforms by defining straight line segments, or downloading of binary images. The A24 Shared Memory may be used for significantly faster downloads than by using the word-serial protocol.

Waveforms can also be linked together to form complex waveforms. Up to 4096 segments can be linked together.

The main waveform output provides up to 11Vp-p into 50Ω (22Vp-p into open circuit). Waveform dc offset or dc output is also provided up to ±5.5 V into 50Ω, 11V into open circuit). Output Lo (outer of BNC) may be up to 7V away from chassis ground.

The control language adheres to the SCPI (Standard Commands for Programmable Instruments) format Version 1992.0, February 1992 (refer to the SCPI manual for further information). SCPI is an industry standard language for remote instrument programming. The Wavetek Model 1385 waveform synthesizer is a single-slot "C" size VXIbus module. Using any manufacturer's VXIbus chassis, the Model 1385 can be controlled using the SCPI language and the appropriate controller.

Multiple ARBs may be linked and operated together inside one VXIbus chassis. In parallel operation, model 1395's may be slaved to a master clock/trigger bus on the VXIbus backplane to create a multichannel waveform synthesizer with phase control between channels.

The model 1385 has extensive self-adjustment utilities built in. Calibration constants are maintained in non-volatile memory (contains no battery).

1.2 SPECIFICATIONS

1.2.1 Waveforms (Functions)

Programmable standard functions include sine, triangle, square, positive ramp, negative ramp, positive haversine, negative haversine, random (noise), sinc (sin x/x) and dc. (The function "WTST" is a reserved function name used for factory maintenance, and it should not be selected as a name for an arbitrary waveform.) One to 450 arbitrary waveforms (traces) may be stored by name in volatile 131,072 point (optionally 524,288 point) RAM memory. Each trace has 12 bits vertical resolution, and from 5 points to the maximum number of points in the waveform memory.

1.2.2 ARB Waveform Creation and Editing

The Arb has a variety of ways to create a waveform. Binary data may be down-loaded from a computer. Internal "standard waveform" algorithms will create exactly one cycle of the waveform requiring nothing more than a name and a space set aside for it (random and sinc, obviously, are not cyclical). Previously created waveshapes residing in memory may be copied to a new trace. Waveforms can be built using line segments.

The Model 1385 Arb has several editing features. After filling memory with data defining the waveform, the user may select only a portion to be "played back" using the `TRACe:LIMit` command. The selected portion may be used for creating a new waveshape using the `TRACe:DATA` command. A trace may also be overwritten with new data with the `TRACe:DATA` command. Any waveform may be stretched or shrunk by copying it into a different size memory space; waveform points are automatically added or removed to retain the integrity of the shape using the `TRACe:DATA` command. By copying waveshape segments end to end, new waveshapes can be created with the `TRACe:DATA` command. A waveform may be resized using the `TRACe:POINts` command. A line segment of any size between 5 points and the maximum memory size can be created using the `TRACe:LINE` command. Any waveform in the directory can be selected for "play back" with the `FUNCtion:USER` <trace_name> and `FUNCtion:SHAPe USER` commands. Individual waveshapes may be deleted by name or the entire memory can be erased using the `TRACe:DELete` command.

1.2.3 Operational Modes

CONTinuous:
The selected trace is output continuously at the selected frequency, amplitude and offset. The sync marker is output once per waveform (selectable as a pulse at the start of the waveform or as a zero-crossing output of the waveform) and the position marker is output at any selected points of the waveform. Frequency is determined by the `TRACe:MODE` (`CW` or `RASTer`), programmed `FREQuency` value (`CW` waveform frequency or `RASTer` sample clock frequency), and `ROSCillator:SOURce` (`INTernal` 125 mHz to 50 MHz, VXIbus `CLOCk`, or `EXTernal` clock source).
For details, see paragraph 1.2.6, **Frequency**.

TRIGgered:
Waveform output is quiescent at first data point of selected trace until a triggering event (selectable by `TRIGger: SOURce` as `INTernal`, `EXTernal`, VXIbus `TTLTrg` or VXIbus Local Bus `CHAin`), after which waveform cycle(s) at the programmed frequency, amplitude and offset is initiated. The waveform completes the number of cycles set by the **Trigger Count** and returns to its quiescent baseline value for another triggering cycle. The triggering baseline is the level of the first waveform address.
For details, see paragraph 1.2.12, **Triggering**.

SYNChronous GATE:
Same as Triggered except output is continuous for duration of gate signal. Last waveform cycle is always completed when gate signal is removed.

ASYNchronous GATE:
Same as Triggered except output is continuous for duration of gate signal. When gate signal goes false, the output is returned to the present state.

AM/SCM:
Operates as in Continuous Mode above, except that the output can be Amplitude Modulated or Suppressed Carrier Modulated by external signals.
For details, see paragraph 1.2.13, **Modulation**.

SWEep:
Operates as in Continuous Mode above, except that the output frequency can be swept by an internal sweep generator between programmed start and stop frequencies.
Sweep capability is provided for standard waveforms and Arbitrary waveforms with a length that is a multiple of 4096 points. A horizontal sweep output voltage is also provided.
For details, see paragraph 1.2.11, **Sweep**.

SEQuence:
Sequence mode provides sophisticated linking, looping and advancing of multiple waveform segments. This allows the creation of long and very complex waveform sequences. For details, see paragraph 1.2.10, **Linked Sequence Operation**.

1.2.4 Input and Output Specifications

1.2.4.1 Outputs
The Model 1385 Arb has four BNC output signals on the front panel: the Main Out, the Position Marker, the Sync Marker, and the Sample Clock.

The Arb also provides a clock to a selected VXIbus backplane ECLTRG line, and a trigger output to the VXIbus Local Bus or to a selected VXIbus TTLTRG line. The ECL Trigger lines can be used to share waveform sample clocks. The TTL Trigger lines can be used for intermodule triggering.

MAIN OUT:
Front panel mounted female BNC, source of programmed function at selected frequency, amplitude and offset. Source impedance is 50 Ω. Protected against short circuit to ground. The BNC marker may be up to 7V away from chassis ground. Any such voltage is rejected in the output by 80dB in the bandwidth DC-440Hz, through up to 1Ω in the Lo lead.

SYNC/SWEEP OUTPUT:
Front panel mounted female BNC. The SYNC MARKER is a TTL compatible pulse into 50Ω at the waveform frequency. Sync generation technique is selectable as "ZCROss" or as "BBITs".

If ZCROss is selected, the sync is generated from zero-cross detecting the waveform. The sync marker is a TTL high whenever the waveform is positive. This is the preferred selection when TRACe:MODE is set to CW (phase accumulation). This is because in CW a particular point may not be used in every scan through the trace.

When BBITs is selected, the SYNC MARKER is a TTL high for a variable number of samples (see POSITION MARKER description for explanation) starting at the first waveform memory location used. When TRACe:MODE is RASTer, either sync technique is applicable. Protected against short circuit to ground.

Levels:	Low level < 0.4V into > 50Ω
	High level > 2.0V into > 50Ω
Rise and Fall time:	< 5 ns into 50Ω

Configuration as a Sweep (Horizontal Sweep) is made when the Frequency Mode is set to Sweep or to List. A linear output ramp from 0 to +10 volts (with an accuracy of ±500 mV, open circuit) proportional to sweep position between selected start and stop limits is provided to drive the horizontal axis of a plotter display device. The output impedance is 600 Ω ± 5%.

POSITION MARKER OUTPUT:
Front panel mounted female BNC. TTL compatible pulse into 50 Ω. User can clear the markers low at all points or set the marker high at any point in a trace. Protected against short circuit to ground.

A marker set at address zero will be true during the trigger quiescent baseline. If address 1 is set (and zero is not), the POSITION MARKER output follows the trigger event plus the pipeline delay.

The Position Marker is one trace point (not necessarily 1 clock) wide for each location selected. In Raster mode, the trace point corresponds to a clock cycle. In CW mode, for high frequency waveforms, a trace point may not be accessed in each pass through the waveform. For very low frequencies, and in CW mode, each trace point may be sampled for a number of clock cycles.

Levels:	Low level < 0.4V into > 50Ω
	High level > 2.0V into > 50Ω
Rise and Fall time:	<8 ns into 50Ω

CLOCK IN/OUT:
Front panel mounted BNC, selectable as either TTL level clock input or TTL level clock output. TTL Clock output is 0.1251 Hz to 50 MHz waveform sample clock in normal operation and 0.1251 Hz to 100 MHz in Clock mode. The output is protected against short circuits to ground.

Configured as an output:

Range:	0.1251 Hz to 100 MHz
Resolution/Accuracy:	Same as the frequency synthesizer.
Levels:	Low level < 0.5V into 50Ω
	High level > 2.1V into 50Ω
Rise and Fall time:	<3 ns into 50Ω

TRIGGER OUTPUT (to VXI Backplane):
One of the eight VXIbus TTLTrigger lines can be programmed as trigger output. The source of the output trigger signal can be selected as "BIT", "Loop COMplete", or "Burst COMplete". The BIT signal is set to be output during a specified Trace, either at the end (Trigger Marker) of the Trace or at selected point(s) within the Trace (Position Marker). LCOMplete indicates that a SEQuence segment has completed its loop count. BCOMplete indicates a Trace or a SEQuence has completed its burst count.

When these sources are selected, the minimum pulse width is 30 ns and maximum frequency that can be applied to a VXIbus TTLTrigger line is 12.5 MHz (per VXIbus specification). Exceeding these limits should be avoided by setting waveform sample frequency below 33 MHz or by programming 2 consecutive BITs when using the TTL Trigger lines for a trigger output.

CLOCK OUTPUT (to VXI Backplane):
Either of the ECL Trigger lines can be programmed as a clock output for intermodule timing. The "master" module supplies its internal clock to this output to be used by "slave" modules as a clock reference for Phase Lock or for tightly controlled trigger timing. When in TRACe: MODE CW and internal clock is selected, the internal clock is a fixed 50 MHz. In TRACe:MODE RASTer the internal clock's mantissa can range from 25 MHz to 50 MHz with 5 digits of resolution under user control.

To set Phase Lock ON, the module selected as the "master" drives the selected ECL Trigger line (ECLTrg<n> ON) with its frequency synthesizer clock signal. All modules, including the "master", get their Reference Oscillator (clock) from the ECLTrg line (ROSC:SOUR ECLT<n>) for optimum timing accuracy. When ECLTrg<n> is selected as an output by the "master":

Clock Frequency Range:	25 MHz to 50 MHz (Raster); 50 MHz (CW).
Resolution/Accuracy:	Same as frequency synthesizer.

1.2.4.2 Inputs

The Model 1385 has two TTL signal inputs on the front panel, clock and trigger. The external clock frequency may range from dc to 50 MHz, the external trigger may range from dc to 5 MHz. Additionally, clock inputs can be accessed from the selected VXIbus ECL Trigger line, and trigger inputs can be accessed through VXIbus Local Bus or the selected TTL Trigger line. The clock and trigger input lines from the backplane are limited by the VXIbus specifications to a maximum of 62.5 MHz for clock and 12.5 MHz for trigger. See VXIbus System Specification for usage.

TRIG IN:
Front panel mounted female BNC, accepts external TTL triggering signal. Input impedance is >1 kΩ. Protected to ±15 Vdc.

Trigger Slope:	Positive or Negative selectable
Amplitude Range:	TTL levels, VinHmin = 2.1 V, VinLmax = 0.8V
Min pulse width:	20 ns
Frequency:	dc to 5 MHz

AM IN:

Front panel mounted female BNC. Signal present at this input amplitude modulates the Main Output signal. AM (amplitude modulation) and SCM (suppressed carrier modulation) are supported. Protected to ± 20 Vdc. For details, see paragraph 1.2.13, **Modulation.**

Frequency Range:	dc to 500 kHz
Amplitude Range:	±1 V maximum
Input Impedance:	10 kΩ

CLOCK IN/OUT:
Front panel mounted female BNC, selectable as either TTL level clock input or TTL level clock output. Clock input used as waveform sample clock. Input impedance is 1 kΩ. Protected to ±20 Vdc.

Configured as an input:

Frequency:	dc to 50 MHz
Amplitude Range:	TTL levels, VinHmin = 2.0 V, VinLmax = 0.4V
Min Pulse Width:	10 ns

TRIGGER INPUT (from VXIbus Backplane):
One of the eight VXIbus TTL Trigger lines (TTLTrg0-7) can be programmed as trigger input from the VXIbus to the model 1385. The TTL Trigger line has a VXI specification limit of 12.5 MHz maximum and 30 ns minimum pulse width. Additionally, the 1385 module has a practical limit of 5 MHz maximum for a trigger input signal.

If another 1385 module is driving the TTL Trigger line, the above limits must not be exceeded. See "**Trigger Output (to VXI Backplane)**" in paragraph 1.2.4.1.

See paragraph 1.2.12, **Triggering**, for examples of VXIbus Backplane triggering.

CLOCK INPUT (from VXI Backplane):
The ECL Trigger lines can be programmed as a clock input from the VXIbus to the model 1385. The "master" module supplies its internal clock to this output to be used by "slave" channels as a clock source for waveform generation. This allows tightly coupled intermodule operation in Phase Lock or triggered modes.

The "slave" module(s) will receive the clock signal

on the selected ECLTrigger line when the Reference Oscillator Source (ROSC:SOUR) is ECLTrg0 or ECLTrg1:

Clock Frequency Range:	25 MHz to 50 MHz (Raster); 50 MHz (CW).

Note

For Standard functions, Trace Mode is CW, and the waveform sample frequency (and thus the Clock output from the Master) is 50 MHz fixed. For the USER function, Trace Mode is Raster, sample frequency is selectable, and the Master's clock output will vary between 25 MHz and 50 MHz with the mantissa of the [SOURce:] FREQuency: RASTer parameter.

Local Bus Inputs/Outputs (VXIbus Backplane)

The VXIbus Local Bus is used for triggering and phase locking.

LBUSA00, LBUSB00

These pins are internally connected to as the Phase Reset Bus. The Phase Reset signal is monitored by all phase locked modules. When this signal is asserted all modules are reset and held at the start address of the active trace. This signal can be driven by any phase locked module. It is driven whenever phase lock is enabled and a programming change is made.

LBUSA02

This pin is used to receive the Chain Trigger signal from the module to the left. The Chain Trigger signal is one of the trigger sources.

LBUSB02

This pin is used to drive the Chain Trigger signal to the module to the right. The Chain Trigger signal is always enabled and its source is the same as that for the TTL Trigger Lines.

LBUSA03, LBUSB03

These pins are internally connected to form the End Trigger Bus. The End Trigger Bus is used to carry the End Trigger signal from the right-most module back to the left-most module. Any module may be programmed to drive the End Trigger signal. The End Trigger signal is one of the trigger sources.

1.2.5 Waveform Characteristics

Square Transition Time:

For ≤10Vp-p:	<9.0 ns
For > 10 Vp-p:	<9.5 ns
Square Aberrations:	<(5% + 20 mV)

Square Symmetry: (0 °C to +50 °C)

< 10 MHz:	50 % ±1 %
≥ 10 MHz:	50 % ±2 %

Sine Distortion: (Maximum Harmonic level, Elliptic filter selected)

<100 kHz, ≤ 10 Vp-p:	-60 dBc
<100 kHz, > 10 Vp-p:	-55 dBc
<5 MHz, ≤10Vp-p:	-45 dBc
<5 MHz, >10Vp-p:	-40 dBc
≤20 MHz, ≤10Vp-p:	-35 dBc
≤20 MHz, >10Vp-p:	-28 dBc

Intermodulation Products: (Maximum Spur level, Elliptic filter selected)

<5 MHz:	- 60 dBc
<10 MHz:	- 50 dBc
≤20 MHz:	- 35 dBc

1.2.6 Frequency

Range:	Sine - 1mHz to 20 MHz.
	Square - 1mHz to 25 MHz.
	Haversines - 1mHz to 20 MHz.
	Other Standard Waveforms - 1mHz to 2 MHz.

Resolution – 8 digits limited by 1 mHz, 5 digits when >20 MHz; 5 digits when the selected function is USER vs. a Standard function.
5 digits when triggered and freq > 10 MHz.

Frequency Accuracy – Determined by the selected clock source. When internal source, frequency reference is provided by the VXIbus [CLK10]. Frequency accuracy is equal to the selected source accuracy specification +200 nHz.

1.2.6.1 Arb Clock and Waveform Timing:

CW (Phase Accumulate) Mode:
The waveform is generated by a phase accumulator. "Standard" waveforms occupy a fixed 4k block of points and are output in CW playback mode. When standard waveforms are selected in a triggered or gated mode of operation, the clock frequency resolution is reduced from eight to five digits at frequencies above 10MHz.

Raster Mode:
User defined (arbitrary) waveforms are generated by scanning through each point in the trace, one clock cycle per point. User waveforms can have horizontal resolution ranging from 5 points to 128K (512K optional) points. The internal raster clock frequency is programmable from 125 mHz to 50 MHz with 5 digits resolution, limited by 0.1 mHz. Waveform frequency is calculated by dividing the clock frequency by the number of points in the trace.

1.2.7 Amplitude

Range:	0.015 to 11Vp-p into 50Ω 0.03 to 22Vp-p into > 10 kΩ
Resolution:	3.5 digits
Monotonicity:	0.2 %

Sinewave Flatness: (relative to 1 kHz amplitude, Elliptic filter selected, non-sweep modes)

< 5 MHz, T_{CAL} ± 10°C :	± 2 %
< 5 MHz, 0 to 50°C:	± 5 %
< 20 MHz, T_{CAL} ± 10°C:	± 5 %
< 20 MHz, 0 to 50°C:	± 10 %

Accuracy: The greater of +1% of setting or the following Limit:

Ampl(Vp) + ABS(Offset)	Limit
> 2.500V & ≤ 5.500 V	± 15 mVp
> 1.250 & ≤ 2.500 V	± 7.5 mVp
> 0.625 & ≤ 1.250 V	± 3.75 mVp
> 312.5 & ≤ 625 mV	± 2 mVp
> 156.3 & ≤ 312.5 mV	± 1 mVp
> 78.13 & ≤ 156.3 mV	± 500 µVp
> 39.06 & ≤ 78.13 mV	± 250 µVp
≤ 39.06 mV	± 125 µVp

1.2.8 Offset

Range:	± 5.5Vdc into 50Ω ± 11Vdc into >10 kΩ
Resolution:	3.5 digits
Accuracy:	The same as for Amplitude Accuracy.

1.2.9 Filtering

(user selectable):	20 MHz 4 pole Bessel 20 MHz 7 pole, 6 zero Elliptic

1.2.10 Linked SEQuence Operation

# of Waveform Segments:	2 to 4096
Segment Loop Count	1 to 1,048,576 or continuous.
Start Conditions:	Continuous or wait for trigger to Start a SEQuence. Uses the Word Serial command or any selected start trigger event.
Advance Conditions:	Segment Loop Count complete; Automatic Loop continuously until selected advance trigger event true; Triggered
Advance Trigger Types:	Event - Trigger must transition to the true state to qualify as an event. Trigger event is latched.
Advance Types:	Synchronous - Current segment is completed before next segment starts. Asynchronous - When advance conditions are met, next segment is started immediately. Current segment is not completed.
Sequence Modes:	Continuous or Triggered; Trigger Count selectable (1 to 65,536).

1.2.11 Sweep

Sweep Time:	30 ms to 1000 s (15 frequency points at 30 ms) with (1/512) s resolution and an accuracy of 0.1% ± (1/512) s.

Sweep Modes:

Continuous up or down - Output frequency sweeps from start frequency to stop frequency, or stop to start if direction is down, with selected characteristic (linear or log).

Continuous up/down - Output frequency sweeps from start frequency to stop frequency, then back to start frequency with selected characteristic.

Triggered up or down - Same as Continuous except output holds at start frequency (or stop if down selected) until receipt of trigger. Programmed number of sweeps, set by Sweep Count, are completed for each trigger signal.

Triggered up/down - Same as Continuous up/down except output holds at start frequency until receipt of trigger. Programmed number of sweeps are completed for each trigger signal.

Triggered Sweep & Hold - Same as Triggered up or down except frequency is held at end of each sweep. An additional trigger is required to return to beginning of sweep.

Triggered Sweep & Hold with Reverse - Same as Triggered up/down except frequency is held at stop frequency. An additional trigger is required to initiate a sweep back to start frequency.

Sweep Spacing:	Linear or Log
Sweep Count:	1 to 1,000,000
Minimum sweep trigger pulse width:	> 500 µs

1.2.12 Triggering

Trigger Sources:

BUS Trigger (*TRG or GET; TRIGger:IMMEDiate)

VXIbus Word Serial Trigger Command

Trigger Input Connector(s)

Internal Trigger Generator(s)

VXI TTL Trigger line driven by another module.

Chained Trigger, receive trigger signal on the VXIbus Local Bus driven from adjacent module.

Trigger Destinations:

Start Trigger:	Initiates gated or trigger modes and starts sequences.
Advance Trigger:	Conditions advances between segments of a sequence.

Internal Trigger Generator(s):

Period:	200 ns to 1000 s
Resolution:	200 ns
Accuracy:	Same as VXIbus CLK10

Trigger Delays and Jitter: (Specified for Trigger Input connectors with TTL input signal)

Delay:	With Standard Functions:	<250 ns
	With User Waveforms:	<400 ns
Jitter:	With Standard Functions:	<20 ns
	With User Waveforms:	<40 ns

Note

Trigger delays and jitter specified with internal sample clock only. If external clock is used:

Delay:	7 clock periods + <100 ns
Jitter:	±1 clock period

Trigger Count:

For waveforms:	1 to 1,048,575
For sequences:	1 to 65,536

Note

Triggered modes of operation are limited to 10 MHz waveform frequency.

1.2.13 Modulation

Types:

AM (Double sideband with carrier)

SCM (Double sideband suppressed carrier)

Bandwidth:	> 500 kHz
Carrier Suppression (SCM):	> -40 dB

Modulation Distortion:

Modulation Freq ≤ 100 kHz:	No harmonic	> -50 dBc
Modulation Freq ≤ 1 MHz:	No harmonic	> -30 dBc

AM Scale Factor: Proportional to programmed amplitude, as follows:

Ampl(Vp) + ABS(Offset)	Ratio of Vout to Vin required for 100 % AM
> 2.500V & ≤ 5.500 V	10:1
> 1.250 & ≤ 2.500 V	5:1
> 0.625 & ≤ 1.250 V	2.5:1
> 312.5 & ≤ 625 mV	1.25:1
> 156.3 & ≤ 312.5 mV	0.625:1
> 78.13 & ≤ 156.3 mV	0.3125:1
> 39.06 & ≤ 78.13 mV	0.1563:1
≤ 39.06 mV	0.07813:1

SCM Scale Factor:	5 V/V
Scale Factor Accuracy:	Carrier ≤ 5 MHz: +5 %; Carrier > 5 MHz: +20 %.

Note

All scale factors assume Main Output terminated into 50Ω load.

1.2.14 Intermodule Operation

Intermodule Phase Control

Two adjacent modules can be assigned a fixed phase relationship. The "Slave" module must be driven by the "Master's" clock generator and the waveforms must be of the same length and frequency. Any change in phase angle between channels will require one waveform cycle to re-acquire phase lock. Phase control signals use the VXIbus Local Bus.

Note

*Phase lock operates with **adjacent** model 1385's using the VXIbus Local Bus.*

Frequency Range:	1 µHz to 20 MHz.
Phase Resolution:	360°/4096 points, standard functions; 360°/points, User defined waveforms.
Phase Accuracy:	± (t/T x 360°), where t = 1 clock period + 10 ns and T = waveform period.

Intermodule Triggering

Adjacent modules can also use the VXI Local Bus to "daisy chain" a trigger signal from the "Start" module, through a number of adjacent modules in the "Chain" to the "End" module. Each module receives the triggering signal on the Local Bus CHAin line from the module to its left, and drives the CHAin line with its selected Trigger Source to the module on its right. The "End" module can be set up to drive a selected TTL Trigger line with its selected Trigger Source back to the "Start" module, closing the loop.

In this fashion, complex and versatile intermodule triggering schemes may be set up. Each module can have its Trigger Source (the signal that it uses to drive the CHAin line) and its output waveform set up independently. Trigger Sources include BIT (pulse occurring at the end of or in a selected position within a trace), Burst COMplete, or Loop COMplete.

1.2.15 Frequency List

Fast frequency changes are possible using [Source:] Frequency:Mode List. In this mode of operation the output frequency is determined by the contents of the Frequency List. The Frequency List is a user programmable list of up to 1024 frequency values.

A trigger event causes a transition to the next frequency in the list. When the last frequency in the list is reached the next trigger returns to the first frequency in the list. The effective size of the list is programmable from 1 to 1024 using the [Source:]List:Points command.

The maximum effective trigger rate in this mode is approximately 2 kHz.

1.2.16 Option 001

Expanded Waveform RAM

Quadruples waveform data storage volatile RAM from 128K to 512K points.

1.2.17 AutoCal/Diagnostics

Each 1385 Module contains DC voltage measurement capability. This feature provides the ability to conduct a limited AutoCal and self diagnostic. Some parts of the calibration (e.g., amplifier flatness) require the use of external measurement equipment. The calibration data is stored in EEPROM. The Processor accesses the data and uses it to correct the output as required to maintain the specified performance.

Note

Performance specifications apply within the specified environmental conditions after a 20 minute warm up period. Specifications are subject to change without notice.

The "T_{CAL}" nomenclature used in this specification refers to the ambient temperature at which the last full Calibration was performed. This temperature must be within the range of 10 to 40 °C.

1.3 GENERAL

1.3.1 SCPI Programming

The Model 1385 adheres to the Standard Commands for Programmable Instruments (SCPI) remote programming format Version 1992.0, February 1992 (refer to the SCPI manual for further information). SCPI is an industry standard language for remote instrument programming. It addresses a variety of test and measurement instrument requirements.

The Wavetek Model 1385 Arbitrary Waveform Generator is a single slot, C size VXIbus module. Using any manufacturer's VXIbus chassis, the Model 1385 can be controlled using the SCPI language and the appropriate controller. Root level commands include:

MODE	**OUTPut**	**SOURce**
STATus	**SYSTem**	**TRIGger**
CALibration	**INITiate**	**RESet**
TEST	**TRACe**	

The Model 1385 supports all Word Serial Commands specified in the VXIbus System Specification (Rev. 1.3) Tables E.1 and E.2 for the above subset/protocol classification. It also supports all IEEE-488.2 Common Commands mandated for use with SCPI.

1.3.2 VXI Interface

CLK10

The internal frequency synthesizer and internal trigger timer utilize the CLK10 signal.

TTLTrigger Lines

Trigger signals can be sourced and received on any one of the eight TTL Trigger Lines.

ECL Trigger Lines

The ECL Trigger Lines can be used to share the output of one module's internal frequency synthesizer among multiple modules. This allows modules to share a clock with the same phase. This is important in order to phase link multiple modules.

Local Bus

The Local Bus is used to transfer high speed trigger and synchronization signals between adjacent modules in a VXIbus chassis. ECL level signals appear on LBUSA00, LBUSC00, LBUS A01 and LBUSC01. TTL level signals appear on LBUSA02, LBUSC02, LBUS A03 and LBUSC03. These signals are always enabled.

The CHAIN trigger signal is driven onto LBUSC02 and received from LBUSA02. This signal is used to trigger adjacent modules. Multiple adjacent modules can propagate the CHAIN trigger down the chain.

The END CHAIN trigger is bussed between LBUSA03 and LBUSC03. Any module can be programmed to drive or receive this signal. Typically the last module in the chain is programmed to drive the END CHAIN trigger signal while the first module in the chain is programmed to receive it. This allows the loop to be closed in the chain.

Shared Memory

64k bytes of A24/D16 Shared Memory are available to be used for the high speed transfer of trace data. Data transfer rates using Shared Memory are much higher than what is possible using Word Serial Data Transfer Protocol.

VXIbus Interface Card

The VXIbus Interface Card contains a Message Based Device interface (MBD) which supports the following subsets/protocols:

A16/A24 D16 Slave

A16/A24 D16 Master

VXIbus Instrument Protocol (I)

VXIbus IEEE-488.2 Instrument Protocol (I4)

Event Generator

Response Generator

All Word Serial Commands specified in the VXIbus System Specification (Rev. 1.3) Tables E.1 and E.2 for the above subset/protocol classification are supported.

Processor & Memory

- 68HC000 CPU (16 MHz)
- 64 kB of local Static RAM
- 128 kB of EPROM
- Real Time Clock generates system tick and adds time and event capability to application code.

VXIbus Interface

- VXIbus P1 and P2 connector
- A16/A24 D16 Bus Master capability
- 64 kB A24 D16 Shared Memory
- Implements the complete Message Based Device interface.
- Full A16/A24 register access qualification.
- Drivers and Transceivers meet the high VMEbus output drive requirements.
- All optional A16 Registers provided.

Application Interface

- Access to all CPU address, data and control lines
- VXIbus TTL Trigger and Local Bus headers.
- VXIbus ECL Trigger and 10 MHz clock buffers.
- SYSCLOCK, RESET* and ACFAIL lines
- Power supplies +5, -5.2, -2, ±12, ±24

1.3.3 Environmental

Temperature Range:	Temperature of last Self Calibration ±10°C for specified operation.
Operating:	0°C to 50°C.
Storage:	-40°C to +71°C (RH not controlled).
Warm-up Time:	30 minutes for specified operation, except stability specifications require 60 minutes.
Altitude:	
Operating:	Sea level to 10,000 ft.
Storage:	Sea level to 15,000 ft.
Relative Humidity (non-condensing):	
0°C to +10°C:	not controlled.
+11°C to +30°C:	95 ±5% RH max.
+31°C to +40°C:	75 ±5% RH max.
+41°C to +50°C:	45 ±5% RH max.
Vibration:	Operates at a vibration level of 0.013 in. from 5 to 55 Hz (2g at 55 Hz).
Shock:	Non-operating, 40g, 9 ms half-sine.
Bench Handling:	Non-operating, 4 in. or point of balance drop, any face, solid wooden surface.

1.3.4 Size

Dimensions:	Single slot, "C" size VXI module. (31 x 262 x 350 mm).
Weight:	<1.6 kg (3.4 lb).

1.3.5 Power

Total:	< 35 Watts

Voltage	Peak Current	Dynamic Current
+5V	2.5A	0.1A
+12V	0.9A	0.05A
-12V	1.0A	0.05A
+24V	0.11A	0.2A
–24V	0.12A	0.2A
–5.2V	2.2A	0.1A
–2V	0.25A	0.02A

1.3.6 Reliability

22,000 hours MTBF at 25°C, ground benign. MIL-HDBK-217 calculation at 50% component stress.

1.3.7 Cooling Requirement

Within a VXIbus mainframe with cooling air. Minimum airflow requirement for 10°C rise is 0.20 mm (0.0075 in) H_2O at 8.57 l/sec (18.15 CFM).

1.3.8 Safety

Designed to MIL-T-28800D, UL-1244, and the VXIbus System Specification, Revision 1.3.

1.3.9 EMC

MIL-STD-461C, Part 7, RE-02, and VXIbus System Specification, Revision 1.3; RE, RS, CE, CS.

Preparation

Section 2

2.1 RECEIVING INSPECTION

Check the shipment at the time of delivery and inspect each box for damage. Describe any box damage and list any shortages on the delivery invoice.

2.1.1 Unpacking Instructions

1. **Unpack the boxes.** Unpack the boxes in a clean and dry environment. Save all the packing material in case the instrument must be returned for repair.
2. **Inspect the shipment for damage.** Inspect the equipment carefully for any signs of mechanical damage regardless of the condition of the shipping boxes.
3. **If necessary, file a claim.** In the case of mechanical damage, call the shipper immediately and start the claim process.
4. **Contact Wavetek.** Inform your Wavetek Sales Office that your equipment has arrived damaged. If you are in the USA or Canada, you may contact Wavetek Customer Service direct (1-800-874-4865 or 619 279-2200, FAX 619_565-9558).

2.1.2 Returning Equipment

Please follow these steps when you return equipment to Wavetek:

1. **Save the packing material.** Always return equipment in its original packing material and boxes. If you use inadequate material, you may be liable for any shipping damage as carriers won't pay claims on incorrectly packed equipment.
2. **Contact Wavetek.** Inform your Wavetek Sales Office that you wish to return your equipment. If you are in the USA or Canada, you may contact Wavetek Customer Service direct (1-800-874-4865 or 619 279-2200, FAX 619_565-9558) and ask for a Returned Material Authorization code.
3. **Pack and ship the equipment.**

2.2 PREPARATION FOR STORAGE OR SHIPMENT

2.2.1 Packaging

If at all possible, always use the original shipping container. However, when using packing materials other than the original, use the following guidelines:

1. Wrap the Model 1385 in ESD sensitive packing material.
2. Use a double-walled cardboard shipping container.

Protect all sides, including the top and bottom, with shock absorbing material (minimum of 2 inch thick material) to prevent movement of the Model 1385 within the container. Seal the shipping container with approved sealing tape. Mark "FRAGILE" on all sides, top, and bottom of the shipping container.

2.2.2 Storage

The Model 1385 should be stored in a clean, dry environment. In high humidity environments, protect the Model 1385 from temperature variations that could cause internal condensation. The following environmental conditions apply to both shipping and storage;

Temperature:	–40°C to +71°C
Relative Humidity:	not controlled, non-condensing
Altitude:	<40000 ft (12192 m)
Vibration:	< 2g
Shock:	< 40g

2.3 PREPARATION FOR USE

Paragraph 2.3 covers the following topics:

Logical Address Selection
Data Transfer Bus Arbitration
Installation

2.3.1 Logical Address Selection

The VXIbus chassis Resource Manager identifies units in the system by the unit's logical address. The VXIbus logical address can range from 0 to 255. However, addresses 0 and 255 are reserved for special functions. Address 0 identifies the Resource Manager. Address 255 permits the Resource Manager to dynamically address the unit based on the units VXIbus chassis slot.

To change the Model 1385's logical address, use the eight position DIP switch (figure 2-1) accessible from the side panel. The Model 1385 uses binary values (2^0 to 2^7) to set the address using the active low address switch. This means the OFF position represents a logical 1. Conversely, an ON position represents a logical 0. Switch position number one is the least significant bit of the address. View A in figure 2-1 illustrates a switch set to a logical address of 3.

Wavetek ships the Model 1385 with a logical address of 255 for Dynamic Configuration. Refer to view B in figure 2-1.

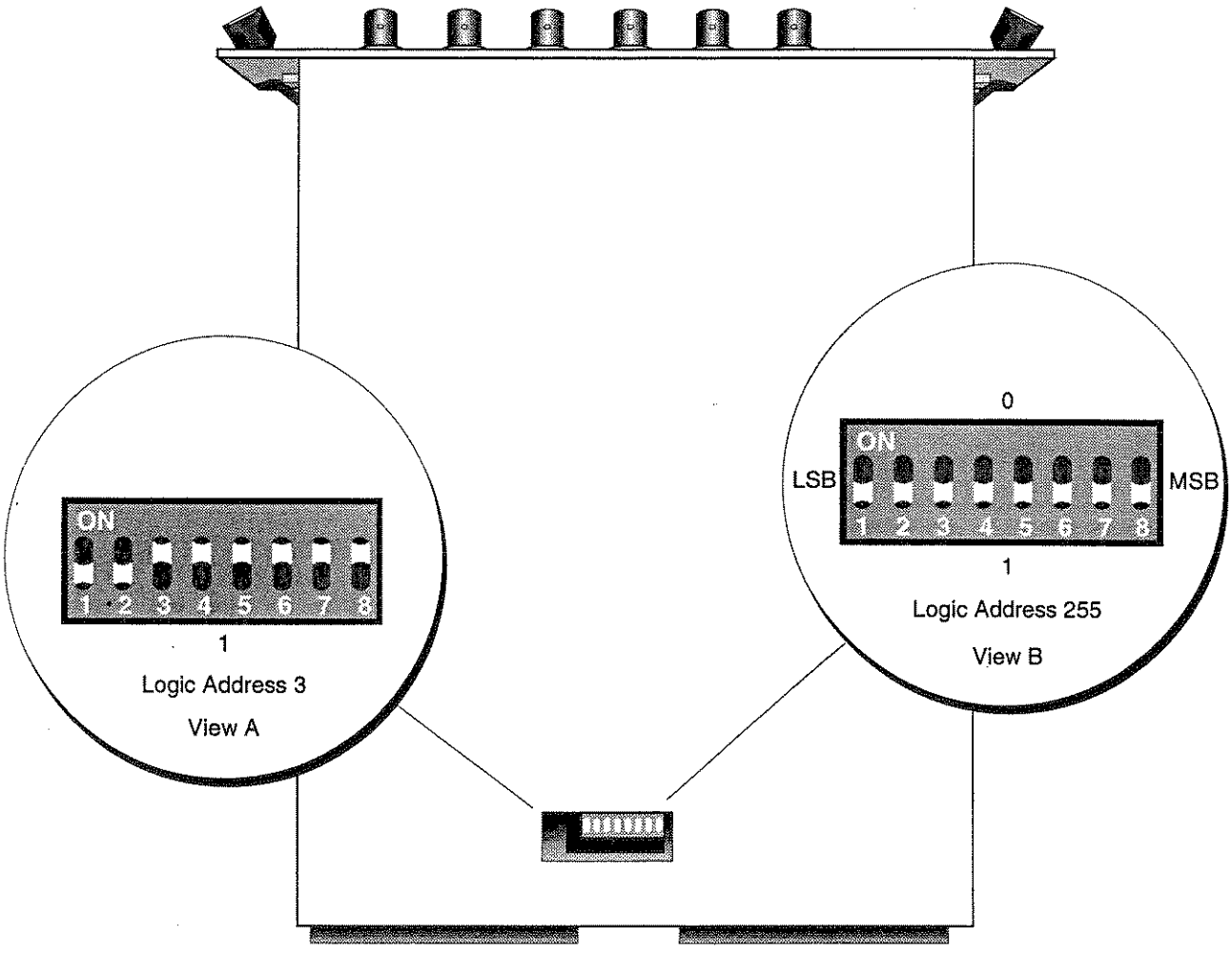


Figure 2-1. Setting the Module's Logical Addresses

2.3.2 Data Transfer Bus Arbitration (Figure 2-2)

The Model 1385 has VMEbus Mastership capability. This means the Sweep/Function Generator, when enabled, sends Responses and Events as signals to its Commander. The Model 1385 cannot drive the interrupt lines.

The Model 1385 is configured as a level 3 requestor by the factory. The level 3 Bus Request and Bus Grant lines are used (BR3*, BG3IN*, and BG3OUT*).

(Continued Overleaf)

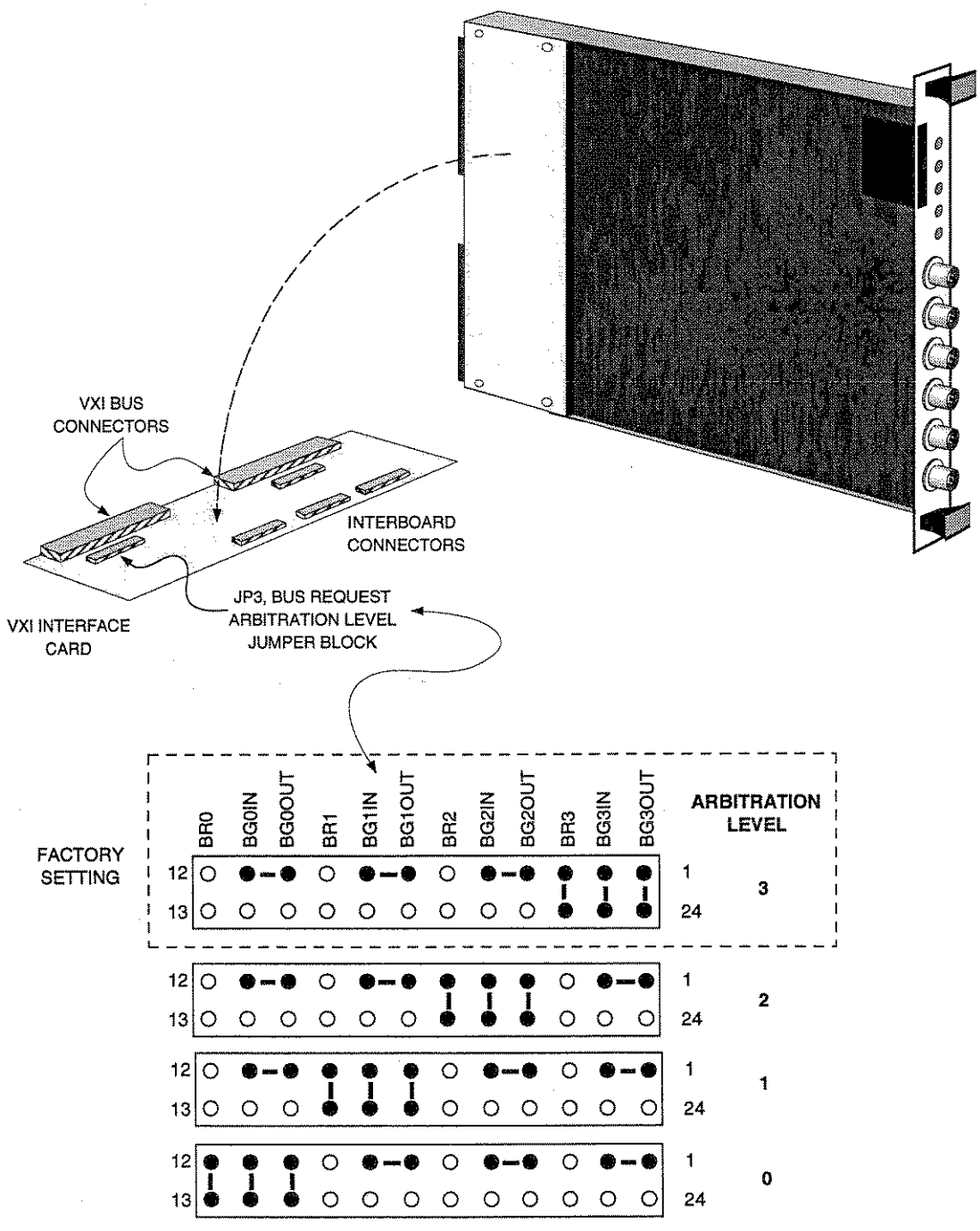


Figure 2-2. Model 1385 Bus Arbitration Level Jumpers

2.3.2 Data Transfer Bus Arbitration *(Contd.)*

The other Bus Grant lines are daisy-chained by jumpers. The VMEbus specifications describe three priority schemes: Prioritized, Round-robin, and Single level. The Prioritized arbitration assigns the bus according to a fixed priority scheme where each of four bus lines has a priority from highest (BR3*) to lowest (BR0*). Round-robin arbitration assigns the bus on a rotating basis. Single level arbitration only accepts requests on BR3*.

If a different requestor level is required, the jumpers must be changed. The following instructions and figure 2-2 will aid in reconfiguring the Model 1385 to a new level. Refer to the VMEbus specification for more information on '*data transfer bus arbitration*'.

CAUTION

The module contains CMOS devices which are sensitive to static electricity. When performing the bus arbitration level change, static electricity discharge straps should be worn.

1. Remove the four flat head screws on the Model 1385 left side panel, remove the panel.
2. Remove the four pan head screws holding the VXIbus Interface card to the main Sweep/ Function Generator board.
3. Slowly and gently lift the VXIbus Interface card up from the Function Generator board. Considerable force may be required as there are four connectors between the two boards with a total of 136 pins. Do not use a metallic prying tool.
4. Change the data transfer bus arbitration jumpers to the desired level. Refer to figure 2-2.
5. Carefully install the VXIbus Interface card onto the Sweep/Function Generator board. Install the four pan head screws, the side panel and the four flat head screws.

2.4 INSTALLATION

The instrument will be installed in a VXIbus mainframe in any slot except slot 0 (zero). When inserting the instrument into the mainframe, it should be gently rocked back and forth to seat the connectors into the backplane receptacles. The ejectors will be at right angles to the front panel when the instrument is properly seated into the backplane. The two captive screws above and below the ejectors are used to secure the instrument into the chassis.

2.5 INITIAL CHECKOUT AND OPERATION VERIFICATION

This procedure provides the operator, service technician, or receiving inspector, etc. with a quick method of verifying the functional operation of the module. This procedure does not test the unit's specifications. This procedure also assumes that the module is properly installed in a "C" size VXIbus chassis with a VXIbus controller in slot 0. Required tools and test equipment are given in table 2-1.

Table 2-1. Test Equipment and Tools

Equipment	Comments
Oscilloscope	Bandwidth: 100 MHz
Signal Source	Frequency: 1kHz to 5MHz Output: TTL
BNC 50Ω Feed-through (2ea)	Accuracy: 0.5% Power: 2W
BNC Coax Cable (2ea)	RG58U, 3 ft. length

Because each step in the procedure is dependent on the preceding step, start with step 1 and continue through to the end. Do not send any command unless specifically instructed to do so within the procedure.

1) Verify proper LED operation during instrument power-up

Table 2-2. LED Status Indications

LED	Normal Result
Run	On
Fail	On, then Off after a second
MID	Flashes very briefly
A16	Flashes
A24	Off

2) Send: `*tst?`

If response = 0, continue

If response ≠ 0, decode error value (see Appendix B).

NOTE

If the test fails on a new or newly factory repaired unit, contact Wavetek. If you are in the USA or Canada, you may contact Wavetek Customer Service direct (1-800-874-4865 or 619 279-2200, FAX 619_565-9558).

3) Connect a coax cable between the MAIN OUT connector and oscilloscope through a 50Ω terminator.

Send: `outp on`

Verify 1Vp, 1kHz sine wave at the MAIN OUT connector.

4) Move cable from the MAIN OUT connector to the MARKER SYNC connector.

Send: `mark:sync on`

Verify TTL level, 1kHz square wave from the MARKER SYNC connector.

5) Send: `mark:sync:sour bbit`

Verify sets of TTL level pulses at 1ms intervals.

6) Move cable from the MARKER SYNC connector to the MAIN OUT connector. DOES N'T WORK

Send: `trac:def temp,50;data temp,tri;:func:shap user` RBH

Verify 1Vp, 1 MHz triangle wave.

7) Move cable from the the ARB OUT connector to the MARKER POSITION connector.

Send: `mark:pos:poin temp,1;poin temp,3;poin temp,5`

Verify three TTL level pulses, each with a width of 20 ns separated by an interval of 20 ns.

8) Send: `mark:pos:aoff temp`

Verify pulses disappear.

9) Move cable from the MARKER POSITION connector to the ARB OUT connector.

Connect an external 100 kHz, TTL level signal to the TRIG IN connector through a 50Ω terminator.

Send: `init:cont off;:trig:sour ext;:trac:data temp,sin`

Verify a 1Vp, 1 µs wide sine wave at 10 µs intervals.

10) Change the 100 kHz external TTL signal to 5 MHz. Move the external source cable from the TRIG IN connector to the CLOCK IN connector.

Send: `init:cont on;:rosc:sour ext`

Verify 1Vp, 100 kHz sine wave.

3.1 INTRODUCTION

This section provides the Operator/Programmer with the information needed to operate the Model 1385 Arbitrary Waveform Synthesizer in a VXI system. The unit resides in a VXI chassis and is subject to all of the restrictions and benefits of that environment.

Paragraph 3.2 describes the Model 1385 connectors and LED indicators. Paragraph 3.3 defines the Model 1385 programming messages. Paragraph 3.4 demonstrates how to operate the Model 1385 using the defined messages.

3.2 CONNECTORS AND LED INDICATORS

This paragraph describes the Model 1385 front panel connectors and LED indicators. Figure 3-1 illustrates the front panel; bold numbers identify the indicators and connectors. Table 3-1 describes the function of each item shown in figure 3-1.

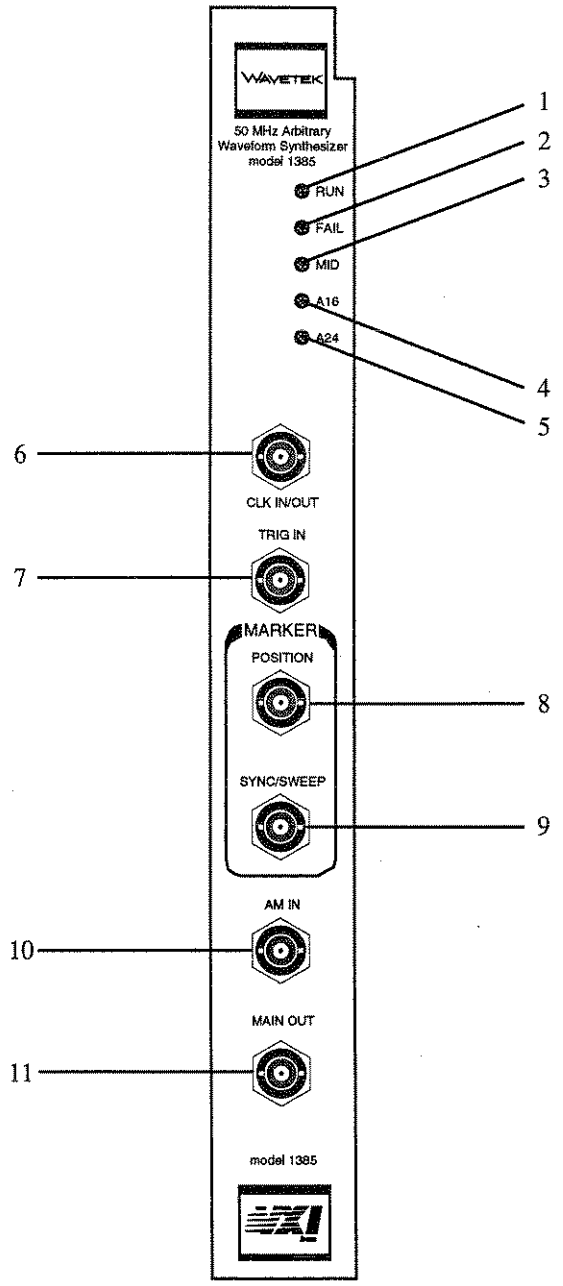


Figure 3-1. Front Panel Connectors and Indicators

Table 3-1. Model 1385 Front Panel Connectors and Indicators

Item	Item Name	Function
1	RUN LED Indicator	When lit, indicates the VXIbus Interface Module microprocessor is running.
2	FAIL LED Indicator	When lit, indicates the VXIbus Interface Module registers are not initialized.
3	MID LED Indicator	This indicator is illuminated when the VXI Bus MODID signal to the slot occupied by this module is asserted.
4	A16 LED Indicator	When lit, indicates that devices on the VXIbus are accessing the generator's A16 registers.
5	A24 LED Indicator	This indicator illuminates during a VMEbus access to the A24 shared memory.
6	CLK IN/OUT Connector	This connector receives an external TTL level signal as a waveform sample time clock, or outputs a TTL signal with a programmable frequency in the range of 125 mHz to 100 MHz.
7	TRIG IN Connector	This connector receives a TTL level external trigger signal.
8	POSITION Connector	This connector outputs TTL pulses which can be set as Position Markers at specific points within arbitrary waveforms.
9	SYNC/SWEEP Connector	This connector outputs TTL pulses as waveform Synchronization Markers, or when the generator is sweeping frequency, outputs a voltage from 0V to 10V to indicate position within the sweep. This may be used to drive an X-Y plotter.
10	AM IN Connector	This connector is the input for external signals to Amplitude Modulate the MAIN OUT. The input voltage range is from -1V to +1V.
11	MAIN OUT Connector	This connector supplies the generator's waveform output. Output level range is from 15 mVpp to 11 Vpp into 50Ω.

KEYWORD	PARAMETER FORM	NOTES
[SOURce]		
:FREQuency		
[:CW|FIXed]	<numeric_value>	(1e-6, **1e3**, 2.5e7)
:MANual	<numeric_value>	(1e-1, **1e3**, 2e7)
:MODE	<**CW**|SWEep|LIST|SMPI>	
RASTer	<numeric_value>	(1e-1, **5e7**, 5e7)
STARt	<numeric_value>	(1e-1, **1e3**, 2e7)
STOP	<numeric_value>	(1e-1, **1e5**, 2e7)

Figure 3-2. Example of Keyword, Parameter Form and Note Hierarchy

3.3 Model 1385 Programming

The Model 1385 communicates within the SCPI (Standard Commands for Programmable Instruments) and IEEE 488.2 standards. Therefore, the Model 1385 must respond to two types of commands: SCPI commands and IEEE 488.2 Common Commands. The IEEE 488.2 Common Commands support functions that are common to all instruments, such as reset, self test and status reporting. Common Commands are non-hierarchical (can be included within SCPI commands without disturbing their hierarchical relationships) and are easily identified by their leading asterisk (*). SCPI commands support functions that are specific to the instrument.

This paragraph provides the following information:

3.3.1 SCPI Command Table

Table 3-2 lists the SCPI commands used in the Model 1385 and indicates their hierarchical relationships. The IEEE 488.2 Common Commands are listed in a separate table (Table 3-5).

3.3.1.1 Long/Short Keyword Forms

The Model 1385 recognizes specific keywords that must be in the accepted long or short format. No other form of the keyword is accepted. For example, to send 'frequency' as part of a message, the short form keyword, shown in the table as upper case letters 'FREQ', or the long form of the same keyword containing both upper and lower case characters 'FREQuency' may be sent. Equal weight is given to upper and lower case characters when sending messages to the Model 1385.

3.3.1.2 Keyword Organization

The indentations of *keywords* indicates their hierarchical relationships according to a tree system. The left-most edge is called the *root node*. Keywords closer to the root node are higher in hierarchy; lower nodes are to the right of their parent node. To program or query a settable parameter, the full path must be defined to reach the keyword appended with the required parameter form.

A SCPI programming string typically starts at the root node and proceeds to the right through branch nodes to the *leaf node*. This string of keywords, separated by colons and completely defining a single path, is defined as a Program Header.

The two forms of Program Headers are commonly referred to as "commands" (see paragraph 3.3.2) or as "queries" (see paragraph 3.3.2.5). A Program Header followed by Program Data is defined as a Program Message Unit (paragraph 3.3.2.1).

In Figure 3-2, the left-most keyword, [SOURce], is directly off the root node. Nodes in this position are called *Subsystems*, and all keywords indented under [SOURce] are part of the Source Subsystem. FREQuency is one of the main parameters under the Source Subsystem. The third level keywords under FREQuency set or query the various frequency related parameters. The brackets around the SOURce and CW keywords indicate that they are *implied nodes*, and they may be included in or omitted from the Program Header at the programmer's option. When included, do not use the brackets in the command.

Referring to Table 3-2, [SOURce] is the only Model 1385 Subsystem which is implied (in brackets). This is the default Subsystem, and is assumed unless another Subsystem is specified at the start of a command.

The root node itself is an implied node and is not directly programmed. A colon at the start of a command resets the SCPI Command Interpreter (included in instrument firmware) to the root node. A leading colon at the root node location is unnecessary (see paragraph 3.3.2.3).

3.3.1.3 Parameter Forms

For the Model 1385, parameters may be in the form of a decimal numeric value (numeric_data), alpha characters (character_data), Boolean data, or Definite Length Arbitrary Block data (paragraph 3.3.3.11). Examples of the first three are:

```
FREQ 1000 (numeric_data)

FUNC SIN (character_data)

OUTP ON (Boolean_data)
```

Notice that in all cases, a space separates the header from data.

Numeric data values for most parameters may be in the form of an integer, a fixed or floating point value, or a special keyword as shown in the following:

integer;	FREQ 1000
fixed point;	FREQ 10.1
floating point;	FREQ 10E3
special form character;	FREQ MIN

When any of the three special form decimal <numer-ic_value> keywords, 'MINimum', 'MAXimum', or 'DEFault', are sent, the parameter being addressed is set to a predetermined <numeric_value>.

The 'MAXimum' and 'MINimum' <numeric_values> are the upper and lower limit values of the parameter. The 'DEFault' <numeric_value> is within the limits of the parameter selected. Defaults values are listed in paragraph 3.4.3.

The Model 1385 uses several character data keywords. These are shown in Table 3-2.

Boolean data expresses an enabled ("on" or "1") or disabled ("off" or "0") state.

3.3.2 Command Message Format

The following paragraphs provide the programmer/operator with an introduction to the general rules that must be followed when sending messages to the Model 1385. For an understanding beyond what is covered in this paragraph, refer to the appropriate SCPI and IEEE 488.2 documents.

Operating the Model 1385 is easy, provided the programmer/operator pays strict attention to the message format, as shown in this manual. Each character, including spaces, must be properly placed or the Model 1385 will record any unrecognized parts of the command string as an error.

Table 3-2 shows the Model 1385 message structure and message relationships. Refer to this while working within this paragraph.

Note

The Model 1385 keeps track of programming errors in memory. The programmer/operator must use the 'SYSTem:ERRor?' query to review these errors.

3.3.2.1 Program Message Unit

The Program Header (command or query) has been previously defined as a complete single path from the root node to a leaf node. It consists of one or more keywords separated by colons. It may also have a leading colon used to explicitly select the root node as the starting point. A Program Message Unit <pmu> consists of a Program Header followed (optionally) by Program Data.

3.3.2.2 Program Message

A Program Message (message) consists of one or more <pmu>'s deliminated by semicolons and followed by a Program Message Terminator, <pmt>.

3.3.2.3 Program Message Delimiters

To piece together the Program Message, the Model 1385 expects commands and parameters in the correct order (per Table 3-2), separated by defined delimiters: colons (:), semicolons (;), and spaces ( ).

Use the colon to separate keywords within a Program Message Unit, for example,

```
VOLT:LEV:IMM:AMPL 5
```

Do not insert spaces between keywords and colons. Placing the optional colon at the beginning of a Program Message Unit ensures the parser starts from the "root" or top level. For example, a complete message with the leading colon is as follows:

```
:OUTP ON
```

The leading colon at the beginning of any new message is optional because the Program Message Terminator (<pmt>) at the end of the previous message sets the parser to the "root" level. The leading colon is not shown for most messages in this section.

The semicolon is used as a Program Message Unit Separator (<pms>). It permits multiple Program Message Units to be linked together into a single message. The colon may follow the semicolon to start the next message unit at the "root". For example,

```
SOUR:FREQ:CW 1E4;:OUTP ON
```

Without the colon following the semicolon, the message must start within the same subsystem as the previous message.

For example:

```
SOUR:FUNC SIN;FREQ:CW 1E4
```

A space separates the Program Header from its data, as shown in the previous example.

Table 3-2. Model 1385 Command Summary

KEYWORD	PARAMETER FORM	NOTES
CALibration		
[:ALL]?		
:DATA	<block>	
:AFCorrection	<point>, <frequency>, <gain>	
:AMPLitude		
[:GAIN]	<numeric_value>	(0.0, **0.0**, 1000)
:OFFSet	<numeric_value>	(0.0, **2048**, 4095)
:AMZero	<numeric_value>	(0.0, **2048**, 4095)
:OFFSet		
[:GAIN]	<**POS**itive|NEGative>, <numeric_value>	(0.0, **0.0**, 1000)
:OFFSet	<**POS**itive|NEGative>, <numeric_value>	(0.0, **2048**, 4095)
:PAZero	<numeric_value>	(0.0, **2048**, 4095)
:SCMZero	<numeric_value>	(0.0, **2048**, 4095)
:STORe		
:STATe	<Boolean_data> (**ON**)	
INITiate		
[:IMMediate]		
:CONTinuous	<Boolean_data> (**ON**)	
OUTPut		
:CLOCk		
:FREQuency	<numeric_value>	(1e-1, **1e3**, 1e8)
:SOURce	<**RAST**er|SYNThesizer>	
:ECLTrg<n>		<n> = 0 to 1 VXIbus
[:STATe]	<Boolean_data> (**OFF**)	ECLTrigger lines
:FILTer		
[:LPASs]		
:TYPe	<**BESS**el|ELLiptic>	
[:STATe]	<Boolean_data>	
[:STATe]	<Boolean_data> (**OFF**)	
:TRIGger		
:MARKer	<**TRIG**ger|POSition>	
:SOURce	<**BIT**|BCOMplete|LCOMplete|INTernal>	
:END		
[:STATe]	<Boolean_data> (**OFF**)	
:TTLTrg<n>		<n> = 0 to 7 VXIbus
[:STATe]	<Boolean_data> (**OFF**)	TTLTrigger lines
RESet		
[SOURce]		
:AM		
[:STATe]	<Boolean_data> (**OFF**)	
:MODE	<**AM**>|<SCM>	

Table 3-2. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:CLOCk		
:CONFigure	<**INP**ut|OUTPut>	
:FREQuency		
[:CW|FIXed]	<numeric_value>	(1e-6, **1e3**, 2.5e7)
:MANual	<numeric_value>	(1e-1, **1e3**, 2e7)
:MODE	<**CW**|SWEep|LIST|SMPI>	
:RASTer	<numeric_value>	(1e-1, **5e7**, 5e7)
:STARt	<numeric_value>	(1e-1, **1e3**, 2e7)
:STOP	<numeric_value>	(1e-1, **1e5**, 2e7)
:FUNCtion		
[:SHAPe]	<shape_name> (<**SIN**usoid|TRIangle|SQUare|HFSQuare|DC|NHSine|PHSsine|PRAMp|NRAMp|PRNoise|USER|SMEMory|WTST>)	
:SQUare		
:DUTY	<numeric_value>	50.0
:TRIangle		
:SYMMetry	<numeric_value>	50.0
:USER	<trace_name>	
:MODE	<**FIX**ed|SEQuence>	
:LIST		
:FREQuency	<numeric_value>,<list_index>	(1e-1, **1e3**, 2e7), (0,1023)
:POINts	<numeric_value>	(1, **1**, 1024)
:MARKer		
:POSition		
:AOFF	<trace_name>	
:POINt	<trace_name>, <point_index>	
:SYNC		
:SOURce	<**ZCR**oss|BBITs>	
[:STATe]	<Boolean_data> (**ON**)	
:TRIGger		
[:STATe]	<trace_name>,<Boolean_data> (**OFF**)	
:PHASe		
[:ADJust]	<numeric_value>	(-180, **0**, 180)
:LOCK	<Boolean_value> **(OFF)**	
:ROSCillator		
:SOURce	<INTernal|EXTernal|ECLTrg<n>>	
:SEQuence		
:ADVance	<**SYNC**hronous|ASYNchronous>, <index>	
:CRATio	<numeric_value>, <list_index>	
:DWELl	<numeric_value>, <list_index>	
:FUNCtion	<trace_name>, <list_index>	
:LENGth	<numeric_value>	
:SEGment	<trace_name>, <numeric_value> <numeric_value>, <**AUTO**matic|TRIGgered>, <**SYNC**hronous|ASYNchronous>, <list_index>	
:STARt	<**SYNC**hronous|ASYNchronous>, <list_index>	

Table 3-2. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:SWEep		
:COUNt	<numeric_value>	(1, **1**, 1e6)
:DIRection	<**UP**|DOWN>	
:MODE	<**CRES**et|TRESet|HRESet|CREVerse|TREVerse|HREVerse|MANual>	
:SPACing	<**LIN**ear|LOGarithmic>	
:TIME	<numeric_value>	(30e-3, **1**, 1e3)
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric_value>	(0, **1**, 5.5)
:OFFSet	<numeric_value>	(-5.5, **0**, 5.5)
STATus		
:OPERation		
:CONDition		
:ENABle		
:ENABle?	<NRf>	
[:EVENt]		
:PRESet		
:QUEStionable		
:CONDition		
:ENABle		
:ENABle?	<NRf>	
[:EVENt]		
SYSTem		
:ERRor?		
:DATE	<year>,<month>,<day>	
:TIME	<hour>,<minute>,<second>	
:VERSion?		
TEST		
[:ALL]?		
:RAM?		
TRACe		
:CATalog?	<trace_name>,<trace_name>,. . .	Up to 50 traces.
:DEFine	<trace_name>,<numeric_value>|<trace_name>	value is new trace size
:DATA	<trace_name>,<block>|<trace_name>	(arbitrary block data)
:LINE	<trace_name>,<point_index1>,<point_value1>,<point_index2>,<point_value2>	Up to 256K for standard functions
:POINt	<trace_name>,<point_index>,<point_value>	
:DELete		
[:NAME]	<trace_name>	
:ALL		

Table 3-2. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[TRACe] (continued)		
:DIRectory?	<name>,<size>,<limit>,<limit>. . .	Up to First 50 traces
:FREE?	<numeric_value>,<numeric_value>	Points available, in-use
:LIMits	<trace_name>,<start_index>,<stop_index>	
:MODE	<**CW**|RASTer>	
:POINts	<trace_name>,<numeric_value>	Value is new trace size
TRIGger		
:COUNt	<numeric_value>	(1, **1**, 1048575)
:GATE		
[:STATe]	<Boolean_data> (**OFF**)	
[:IMMediate]		
:POLarity	<**POS**itive|NEGative>	
:SOURce		
[:STARt]	<**INT**ernal|EXTernal|CHAin|ECHain|TTLTrg<n>>	<n> = 0 to 7 VXIbus TTLTrigger lines
:ADVance	<**INT**ernal|EXTernal|CHAin|ECHain>	
:TIMer	<numeric_value>	(2e-7, **1e-3**, 1e4)

3.3.2.4 Program Message Terminators

The Model 1385 accepts New Line (NL, <LF>), END, or NL with END as the Program Message Terminator (<pmt>). However, the END (<EOI>) is the preferred <pmt> because it initiates an immediate transfer from the input buffer to the Language Processor for parsing. The other terminators may not be recognized until the buffer fills.

3.3.2.5 Queries

Unless otherwise indicated, each header with a parameter form also has a query form so that the current setting may be reported back. A query is programmed by following the leaf node keyword with a question mark (?), no space.

For example, send:

```
SOUR:FREQ:CW?
```

or the reduced form:

```
FREQ?
```

to query the frequency setting. The response for this query is a floating point numerical value representing the frequency in Hertz. For example, if the response is 1 kHz, the returned value is:

```
1.000000E+03
```

For queries that include parameters, the question mark and a space are inserted prior to the parameter; for example:

```
FREQ? MAX
```

Some commands may exist in query form only, for example:

```
SYST:ERR?
```

Some queries are mandated such as *ESE?, *SRE?, and *TST?; see paragraph 3.3.4.

3.3.3 Model 1385 SCPI Commands

This paragraph introduces the operator to the Model 1385 Program Message Units. Paragraph 3.4 covers the relationship between these units. For a description of message format, refer to paragraph 3.3.2.

This paragraph uses only the short form keyword recognized by the Model 1385 (refer to Table 3-2). Program Message Terminators are assumed, and therefore not shown in the examples. However, most optional keywords are shown to document the program flow.

3.3.3.1 CALibration Subsystem

The Calibration query causes an internal self calibration to be performed and a response to be placed in the Output Queue. The response to the `*CAL?` query is an ASCII string representing an integer value.

The value of "0" is returned if the auto calibration passed and a non-zero value in the range of 32767 to -32768 is returned if the auto calibration failed. The interpretation of the value returned in the event of a failed self calibration is defined in Appendix A.

The `*CAL?` query invokes the same internal self calibration functions and returns the same response as the `CALibration[:ALL]?` query documented below. Table 3-3a shows the CALibration Subsystem excerpted from the Command Table:

`CALibration[:ALL]?`

Performs a DC calibration of the output amplitude and offset voltage levels and stores the calibration data in non-volatile memory. If the calibration is successful, use of the data is enabled. If the calibration is unsuccessful for any reason, use of the data is disabled and default correction factors are used.

This query returns a value of "0" if the auto-calibration is successful and a non-zero positive integer value if not. The response value will indicate the nature of the failure.

The value of a 16-bit Self Calibration Status Word is returned in response to the calibration query. The format of the Self Calibration Status Word is shown below:

Self Calibration Status Word

Table 3-3a. CALibration Keyword Excerpt From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
CALibration		
[:ALL]?		
:DATA	<block>	
:AFCorrection	<point>, <frequency>, <gain>	
:AMPLitude		
[:GAIN]	<numeric_value>	(0.0, **0.0**, 1000)
:OFFSet	<numeric_value>	(0.0, **2048**, 4095)
:AMZero	<numeric_value>	(0.0, **2048**, 4095)
:OFFSet		
[:GAIN]	<**POS**itive|NEGative>, <numeric_value>	(0.0, **0.0**, 1000)
:OFFSet	<**POS**itive|NEGative>, <numeric_value>	(0.0, **2048**, 4095)
:PAZero	<numeric_value>	(0.0, **2048**, 4095)
:SCMZero	<numeric_value>	(0.0, **2048**, 4095)
:STORe		
:STATe	<Boolean_data> (**ON**)	

The Self Calibration Status Word is split into two fields, one occupying the lower eight bits and the other occupying the upper eight bits.

The Calibration Number field contains the number of the first sub-calibration in which a failure was detected. Sub-calibration numbers range from 1 to 255. Sub-calibrations are performed in the same sequence as they are numbered.

The Error Code field contains a bit weighted code that is unique to the sub-calibration. Refer to Appendix A for more information.

`CALibration:DATA <block>`

Allows calibration data to be transferred directly to and from the Trace Memory in the form of Arbitrary Block Program Data.

The `CALibrate:DATA:STORe` command must be used if data is to be transferred to the EEPROM. The format of this data will be documented in Section 5 of this manual.

`CALibration:DATA:AFCorrection`
`<point>,<frequency>,<gain>`

Sending this program message sets the amplitude gain correction for a specified frequency point. The value of "point" is an index into a table of gain corrections, and it should be an integer ranging from 0 to 5. Associated with each gain correction is the frequency at which the gain correction was calculated. The default table is shown below:

Frequency	Point	Gain
0 MHz	0	1.0
7 MHz	1	1.0
13 MHz	2	1.0
17 MHz	3	1.0
20 MHz	4	1.0
25 MHz	5	1.0

Internally the value of the gain parameter defaults to 1.0. Programmed gains should not be too far from this value. The gain and frequency parameters for points 0 and 5 should remain unchanged.

Between frequency points, linear interpolation is used to calculate the gain correction of the amplitude. The frequency points chosen correspond to the average position of the break points in the frequency response of the elliptic filter.

`CALibration:DATA:AFCorrection?`
`<point>`

Allows the contents of the amplitude gain correction table to be queried. The response is in the format:

`<frequency>,<gain>`

`CALibration:DATA:AMPLitude[:GAIN]`
`<numeric_value>(0.0)`

Sending this program message directly sets the amplitude gain correction. This value is usually calculated by the self calibration, and has a DEFault and MINimum value of 0.0 and a MAXimum value of 1000. This value may also be queried.

`CALibration:DATA:AMPLitude:OFFSet`
`<numeric_value>(2048)`

Sending this program message directly sets the offset of the amplitude control. Self calibration usually calculates this value, which has a DEFault value of 2048, a MINimum value of 0, and a MAXimum value of 4095. This value is an integer and may also be queried.

`CALibration:DATA:AMZero`
`<numeric_value>(2048)`

Sending this program message directly sets the Amplitude Modulation Zero DAC. The numeric value is an integer value between 0 and 4095 corresponding to the range of the 12-bit DAC. This value defaults to 2048. Self calibration usually determines this value. This value may also be queried.

`CALibration:DATA:OFFSet[:GAIN]`
`<POSitive|NEGative>,`
`<numeric_value>(0.0)`

Sending this program message directly sets the Offset Gain Correction. Self calibration usually calculates this value, which has a DEFault and MINimum value of 0.0 and a MAXimum value of 1000.

This value may also be queried, as follows:

```
CALibration:DATA:OFFSet[:GAIN]?
                    <POSitive|NEGative>
```

```
CALibration:DATA:OFFSet:OFFSet
                    <POSitive|NEGative>,
                    <numeric_value>(2048)
```

Sending this program message directly sets the output offset voltage control offset. Self calibration usually calculates this value, which has a DEFault value of 2048, a MINimum value of 0, and a MAXimum value of 4095.

This value is an integer and may also be queried using the query shown below.

```
CALibration:DATA:OFFSet:OFFSet?
                    <POSitive|NEGative>
```

```
CALibration:DATA:PAZero
                    <numeric_value>(2048)
```

Sending this program message directly sets the DAC controlling the preamplifier zero correction. This parameter is an integer value between 0 and 4095 which corresponds to the range of the 12-bit DAC. This value defaults to 2048. Self calibration usually determines this value, which may also be queried.

```
CALibration:DATA:SCMZero
                    <numeric_value>(2048)
```

Sending this program message directly sets the DAC controlling the Suppressed Carrier Modulation Zero. The parameter is an integer value between 0 and 4095 corresponding to the range of the 12-bit DAC. This value defaults to 2048, and may be queried.

```
CALibration:DATA:STORe
```

Sending this program message causes correction data that has been downloaded using the program messages in the CALibrate:DATA subsystem to be stored into non-volatile memory. This should be done only after all correction data has been finalized so as to minimize writes to the EEPROM.

```
CALibration:STATe <ON|OFF>
```

Enables correction of the output amplitude and offset voltage levels using the calibration data stored in non-volatile memory. If the calibration corrections are disabled then default corrections are used.

3.3.3.2 INITiate Subsystem

Refer to Table 3-3b

```
INITiate[:IMMediate]
```

This command is included to support the SCPI specification, but it does not alter the setup of the Model 1385.

```
INITiate:CONTinuous <ON|OFF>
```

Sending this program message selects between continuous mode of operation and a non-continuous mode of operation. In continuous, the selected trace or function is continuously output at the module's Main Out, using the (default) command:

```
INIT:CONT ON
```

Non-continuous modes include Triggered and Gated modes. Triggered mode outputs the selected trace or function for a number of cycles determined by the trigger COUNt once per triggering event at the module's Main Out, using the command:

```
INIT:CONT OFF;:TRIG:GATE OFF;
                :TRIG:COUN <value>
```

Gated mode causes the selected function or trace to be output while the trigger source is true, and quiescent while the source is false.

Table 3-3b. INITiate Keyword Excerpt From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
INITiate [:IMMediate] :CONTinuous	<Boolean_data> (**ON**)	

3.3.3.3 OUTPut Subsystem

Refer to Table 3-3c.

```
OUTPut:CLOCk:FREQuency
                    <numeric_value>(1e3)
```

Sets up the CLK IN/OUT BNC as an output sourcing a clock signal with the specified frequency. The output clock frequency ranges from 1e-1 to 1e8 Hz, with 1e3 as the default value. This command causes all other outputs to be turned off and completely reconfigures the internal state of the instrument to support this mode of operation.

```
OUTPut:CLOCk:SOURce
                    <RASTer|SYNThesizer>
```

Selects the source of the clock output to the CLK IN/OUT BNC.

- **RASTer** Raster clock. When the trace mode is set to raster using the program message:

  ```
  TRACe:MODE RASTer
  ```

  then the raster frequency can be programmed using the program message:

  ```
  [SOURce:]FREQuency:RASTer
                    <numeric_value>
  ```

- SYNThesizer Direct output of frequency synthesizer. The frequency of this signal is twice the programmed raster clock frequency when the raster clock frequency ranges between 25 and 50 MHz. When the raster frequency is less than 25 MHz then the actual synthesizer frequency is not readily determined.

```
OUTPut:ECLTrg<n>[:STATe] <ON|OFF>
```

Enables the VXIbus ECL Trigger Lines to drive the backplane. Each ECL Trigger Line output can be separately enabled. Valid numeric suffixes are in the range 0 through 1. The signal sourced by the ECL Trigger Lines is the divided output of the frequency synthesizer which is in the range of 25 MHz to 50 MHz.

```
OUTPut:FILTer[:LPASs][:STATe]
                    <ON|OFF>
```

Enables the 20 MHz low pass filter between the output of the waveform DAC and the pre-amplifier.

```
OUTPut:FILTer[:LPASs][:STATe]?
```

Returns a numeric value indicating the enable state of the filter.

```
OUTPut:FILTer[:LPASs]:SELect
                    <BESSel|ELLiptic>
```

Selects the filter type as Bessel (flattest response with constant group delay) or Elliptic (steepest cutoff).

```
OUTPut[:STATe] <ON|OFF>
```

Controls the state of the MAIN OUT output relay.

Table 3-3c. OUTPut Keyword Excerpt From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
OUTPut		
:CLOCk		
:FREQuency	<numeric_value>	(1e-1, **1e3**, 1e8)
:SOURce	<**RAST**er|SYNThesizer>	
:ECLTrg<n>		<n> = 0 to 1 VXIbus
[:STATe]	<Boolean_data> (**OFF**)	ECLTrigger lines
:FILTer		
[:LPASs]		
:TYPe	<**BESS**el|ELLiptic>	
[:STATe]	<Boolean_data>	
[:STATe]	<Boolean_data> (**OFF**)	
:TRIGger		
:MARKer	<**TRIG**ger|POSition>	
:SOURce	<**BIT**|BCOMplete|LCOMplete|INTernal>	
:END		
[:STATe]	<Boolean_data> (**OFF**)	
:TTLTrg<n>		<n> = 0 to 7 VXIbus
[:STATe]	<Boolean_data> (**OFF**)	TTLTrigger lines

```
OUTPut:TRIGger:MARKer
                    <TRIGger|POSition>
```

Selects the source of the trigger qualifier marker. The trigger qualifier marker is used by the internal hardware to condition the selected trigger to be output by the module (see below).

- **TRIG**ger Selects the trigger marker bit. The trigger marker is set 7 points from the last point of a trace. If the trace is smaller than 9 points it is set on the second point. The trigger marker must be enabled for a specified trace.
- POSition Selects the position marker bit. The position marker can be set on any point in a specified trace.

```
OUTPut:TRIGger:SOURce
                    <BIT|LCOMplete
                    |BCOMplete|INTernal>
```

Selects the trigger output source. The trigger source can be sent to VXIbus Local Bus (trigger chaining) or to one of the VXIbus TTL Trigger lines.

- **BIT** Selects a signal driven by the trigger marker bit in Trace RAM as selected by the `OUTPut: TRIGger:MARKer` command.
- BCOMplete Selects the Marker qualified with the Burst Complete signal.

  If `FUNCtion:MODE: SEQuence` has been programmed, then this signal is additionally qualified with the Loop Complete signal.
- LCOMplete Selects the Marker qualified with the Loop Complete signal.
- INTernal Selects the output of the internal trigger timer, set by the `TRIGger:TIMer <numeric_value>` command.

```
OUTPut:TRIGger:END[:STATe]
                    <ON|OFF>
```

Connects or disconnects the selected trigger output to the VXIbus Local Bus ECHain (end chain) line.

```
OUTPut:TTLTrg<n>[:STATe] <ON|OFF>
```

Enables the VXIbus TTL Trigger Lines to drive the backplane. Only one TTL Trigger Line output can be on at any given time. Enabling the output of one TTL Trigger Line automatically disables the output of all other TTL Trigger Lines.

The signal sourced by the VXIbus TTL Trigger Lines is selected by the `OUTPut:TRIGger:MARKer` and `OUTPut:TRIGger:SOURce` commands.

Valid numeric suffixes are in the range 0 thru 7.

3.3.3.4 RESet Subsystem

```
RESet
```

Resets all parameters to their default state (see paragraph 3.4.3). This command has no effect on the Trace subsystem.

3.3.3.5 SOURce Subsystem

The Source SubSystem is shown in Table 3-3d.

```
[SOURce:]AM:STATe <ON|OFF>
```

Enables the amplitude modulation input (AM IN BNC).

```
[SOURce:]AM:MODE <AM|SCM>
```

Selects the amplitude modulation mode.

- **AM** Standard 0 to 100 % amplitude modulation.
- SCM Suppressed carrier amplitude modulation (also referred to as Double-Sideband Suppressed Carrier).

```
[SOURce:]CLOCk
   :CONFigure <INPut|OUTPut>
```

Configures the use of the CK IN/OUT BNC as an input or as an output.

```
[SOURce:]FREQuency [:CW]
                <numeric_value> (1e3)
```

Controls the frequency of the function output when the Trace Mode is set to `CW`. Default value is 1.0e3, and allowable values range from 1e-6 to 2.5e7.

```
[SOURce:]FREQuency
   :MANual <numeric_value>  (1e3)
```

Controls the frequency of the function output when `[SOURce:]FREQuency:MODE SWEep` and `[SOURce:]SWEep:MODE MANual` are selected. Default value is the Start Frequency of the sweep (1.0e3). Allowable range is 1e-1 to 2e7.

[SOURce:]FREQuency:MODE
<**CW**|SWEep|LIST|SMPI>

Controls the frequency sweep logic.

- **CW** The output frequency of a standard function is determined by the programming of [SOURce:] FREQuency[:CW]. The output frequency of a USER function is determined by the number of points in the trace and the programming of [SOURce:]FREQuency RASTer.
- SWEep The function output frequency is swept from one frequency to another in a fixed time period. The sweep mode, direction, and duration are set by the Sweep Subsystem commands. The sweep frequencies range between the limits set by the [SOURce:]FREQuency:STARt and the [SOURce:]FREQuency:STOP commands. When in the MANual Sweep Mode the output frequency is set by the [SOURce:]FREQuency: MANual command.

Table 3-3d. SOURce Keyword Excerpt From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
[SOURce]		
:AM		
[:STATe]	<Boolean_data> (**OFF**)	
:MODE	<**AM**>|<SCM>	
:CLOCk		
:CONFigure	<**INP**ut|OUTPut>	
:FREQuency		
[:CW|FIXed]	<numeric_value>	(1e-6, **1e3**, 2.5e7)
:MANual	<numeric_value>	(1e-1, **1e3**, 2e7)
:MODE	<**CW**|SWEep|LIST|SMPI>	
:RASTer	<numeric_value>	(1e-1, **5e7**, 5e7)
:STARt	<numeric_value>	(1e-1, **1e3**, 2e7)
:STOP	<numeric_value>	(1e-1, **1e5**, 2e7)
:FUNCtion		
[:SHAPe]	<shape_name> (<**SIN**usoid|TRIangle|SQUare|HFSQuare|DC|NHSine|PHSsine|PRAMp|NRAMp|PRNoise|USER|SMEMory|WTST>)	
:SQUare		
:DUTY	<numeric_value>	50.0
:TRIangle		
:SYMMetry	<numeric_value>	50.0
:USER	<trace_name>	
:MODE	<**FIX**ed|SEQuence>	
:LIST		
:FREQuency	<numeric_value>,<list_index>	(1e-1, **1e3**, 2e7), (0,1023)
:POINt	<numeric_value>	(1, **1**, 1024)
:MARKer		
:POSition		
:AOFF	<trace_name>	
:POINts	<trace_name>, <point_index>	
:SYNC		
:SOURce	<**ZCR**oss|BBITs>	
[:STATe]	<Boolean_data> (**ON**)	
:TRIGger		
[:STATe]	<trace_name>,<Boolean_data> (**OFF**)	

- LIST The function output frequency is determined by the contents of a table of 1024 frequency settings referred to as the Frequency List. The effective length of the List is determined by the [SOURce:]LIST:POINts command. The frequencies in the list are sequenced with trigger events. Each trigger event selects the next frequency in the list. When the last element in the list is reached, the next trigger sequences back to the first element in the list.

- SMPI The function output frequency is determined by a 48-bit Phase Increment value placed in the lower 6 bytes of A24 shared memory. The value in shared memory is transferred to the Address Generator upon receipt of a Word Serial Command with the value of 0X0003. The Response Register WR bit is not set until the value has been transferred.

Table 3-3d. SOURce Keyword Excerpt From Table 3-2 (Continued)

KEYWORD	PARAMETER FORM	NOTES
:PHASe		
[:ADJust]	<numeric_value>	(-180, **0**, 180)
:LOCK	<Boolean_value> **(OFF)**	
:ROSCillator		
:SOURce	<INTernal|EXTernal|ECLTrg<n>>	
:SEQuence		
:ADVance	<**SYNC**hronous|ASYNchronous>, <index>	
:CRATio	<numeric_value>, <list_index>	
:DWELl	<numeric_value>, <list_index>	
:FUNCtion	<trace_name>, <list_index>	
:LENGth	<numeric_value>	
:SEGment	<trace_name>, <numeric_value> <numeric_value>, <**AUTO**matic|TRIGgered> <**SYNC**hronous|ASYNchronous>, <list_index>	
:STARt	<**SYNC**hronous|ASYNchronous>, <list_index>	
:SWEep		
:COUNt	<numeric_value>	(1, **1**, 1e6)
:DIRection	<**UP**|DOWN>	
:MODE	<**CRES**et|TRESet|HRESet|CREVerse|TREVerse|HREVerse|MANual>	
:SPACing	<**LIN**ear|LOGarithmic>	
:TIME	<numeric_value>	(30e-3, **1**, 1e3)
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric_value>	(0, **1**, 5.5)
:OFFSet	<numeric_value>	(-5.5, **0**, 5.5)

```
[SOURce:]FREQuency :RASTer
                <numeric_value> (5e7)
```

Controls the trace scan rate when the Trace Mode is set to RASTer. Default value is 5.0e7. Allowable range is 1e-1 to 5e7.

```
[SOURce:]FREQuency:STARt
                <numeric_value> (1e3)
```

Controls the start frequency of the function output when the frequency mode is set to SWEep. Default value is 1.0e3. Allowable range is 1e-1 to 2e7.

```
[SOURce:]FREQuency:STOP
                <numeric_value> (1e5)
```

Controls the stop frequency of the function output when the frequency mode is set to SWEep. Default value is 1.0e5. Allowable range is 1e-1 to 2e7.

```
[SOURce:]FUNCtion[:SHAPe]
                    <shape_name>
```

Selects the shape of the output signal.

The following function shapes are available:

- DC An non-varying signal with respect to time with an amplitude of 0.
- HFSQuare High frequency square wave. This square wave has faster rise and fall times than the standard square wave. Its maximum frequency is 25 MHz.
- NHSine Negative haversine.
- NRAMp Negative ramp.
- PHSine Positive haversine.
- PRAMp Positive ramp.
- PRNoise Periodic random noise.
- SINusoid A sinusoidal signal.
- SMEMory Uses the data located in the first 8k bytes of the VXIbus A24 Shared Memory. Data in Shared Memory is organized as 4096, 16-bit words. Each word is an integer value between 1 and 4095.
- SQUare A square wave signal.
- TRIangle A triangle wave signal.
- USER Selects the user defined function specified by the `SOURce:FUNCtion:USER` command. Selecting a user function automatically switches the method of waveform generation to raster scan.
- WTST An alternating pattern of high and low values. The values 4095 and 1 are alternately written to every point in the trace. This pattern is used for factory maintenance procedures.

For all functions except USER the following parameters are automatically set:

```
TRACe:MODE CW[SOURce:]MARKer
:SYNC:SOURce ZCRoss OUTPut
       :FILTer ON
```

The Model 1385 automatically selects filters at the time that the FUNCtion is selected. For TRIangle, NRAMp and PRAMp, the Bessel filter is inserted. For SINusiod, NHSine, PHSine, and SQUare, the Elliptic filter is inserted. For all other functions, no filter is automatically inserted.

Selection of function USER will cause the following parameters to be automatically set as described below:

```
TRACe:MODE RASTer[SOURce:]MARKer
:SYNC:SOURce BBITs OUTPut
:FILTer OFF
```

These settings may be changed after the function shape has been selected.

```
[SOURce:]FUNCtion:SQUare:DUTY
                <numeric_value> (50.0)
```

This command adjusts the duty cycle of the squarewave function. The duty cycle is the time the output is high over the period of the function expressed as a percentage. A duty cycle of 0 would cause the output to be a DC level at negative amplitude. A duty cycle of 100 would cause the output to be a DC level at a positive amplitude. A duty cycle of 50 causes the output to spend half the period at a negative amplitude and half the period at a positive amplitude.

```
[SOURce:]FUNCtion:TRIangle:SYMMetry
                <numeric_value> (50.0)
```

This command adjusts the symmetry of the triangle function. The symmetry is the time the function spends rising to the time the function spends falling expressed as a percentage. A symmetry of 0 would cause the output to be a negative ramp. A symmetry of 100 would cause the output to be a positive ramp. A symmetry of 50 causes the output to spend half the period rising and half the period falling.

```
[SOURce:]FUNCtion:USER
                    <trace_name>
```

Selects one of the user functions defined under the Trace subsystem. The user function will be output only if USER is selected by the `[SOURce:]FUNCtion[:SHAPe]` command.

```
[SOURce:]FUNCtion:MODE
                    <FIXed|SEQuence>
```

Controls the function sequence logic. If the function mode is set to FIXed then the output function is determined by `[SOURce:]FUNCtion[:SHAPe]` parameter. If the function mode is set to SEQuence then the output function is determined by the contents of the sequence table.

```
[SOURce:]LIST:FREQuency <numeric_
                value>(1e3),<list_index>(1)
```

This command allows the elements in the Frequency List to be modified by the user. The Frequency List is a table of frequency values which is utilized when `[SOURce:]FREQuency:MODE LIST` is programmed. The frequency values in the list may range from 1e-3 to 2e7. Each element in the list has a default value of 1e3. The Frequency List contains 1024 elements with indices from 0 to 1023. The number of "active" elements in the list is set with the `[SOURce:] LIST:POINts` command. The query form of this command returns the frequency value at a specified index location, as follows:

```
[SOUR:]LIST:FREQ? <list_index>
```

```
[SOURce:]LIST:POINts
                <numeric_value> (1)
```

This command allows the active length of the Frequency list to be modified by the user. This value defaults to 1 and ranges from 1 to 1024.

```
[SOURce:]MARKer:POSition:AOFF
                    <trace_name>
```

Sets all POSITION marker bits to the inactive state. There is no query form for this command.

```
[SOURce:]MARKer:POSition:POINt
                <trace_name>,<point_index>
```

Sets the POSITION marker at the specified point within the specified trace to the active state. There is no query form for this command. "Point index" is an integer value.

```
[SOURce:]MARKer:SYNC:SOURce
                    <ZCRoss|BBITs>
```

Selects the method used to generate the SYNC marker output.

- ZCROss Selects the output of a comparator. The comparator output is high if the instantaneous signal is above mid-scale level and low if the signal is below mid-scale level.
- BBITs The Marker signal is derived from a bit in Trace RAM. The bit is set so that the marker is active for the first several points of the trace.

```
[SOURce:]MARKer:SYNC:[STATe]
                    <ON|OFF>
```

Enables the SYNC marker output. This command will be accepted but will not perform any function since there is no way to disable this output.

```
[SOURce:]MARKer:TRIGger[:STATe]
                <trace_name>,<Boolean>
```

The Trigger Marker is an internal signal generated to condition selected trigger outputs. Selected by OUTPut:TRIGger:MARKer as TRIGger (end of trace). This command selects which trace(s) will generate Trigger Markers, and thus trigger outputs to the VXI Local Bus or TTL Trigger lines. Queried with the following:

```
[SOURce:]MARKer:TRIGger[:STATe]?
                    <trace_name>
```

```
[SOURce:]PHASe[:ADJust]
                <numeric_value> (0)
```

Controls the phase of the output waveform. The parameter has units of degrees. The value defaults at 0 degrees and ranges from -180 to +180 degrees.

```
[SOURce:]PHASe:LOCK <ON|OFF>
```

Enables phase locking between modules. Phase locked modules must reside in adjacent slots in the VXIbus chassis, because phase lock signals use the VXIbus Local Bus. In order for two or more modules to be phase locked they must have traces of the same size running at the same frequency.

For tight coupling, all modules should be using the same clock. This is done by having the "master" module, usually the left-most module in the group, output its clock to the backplane.

[SOURce:]PHASe:LOCK <ON|**OFF**> *(Contd.)*

The Clock output is enabled with the command OUTPut: ECLTrg<n>[:STATe] ON. All modules, including the Master, must source the master module's clock output from the backplane. This is done by using the command [SOURce:]ROSCillator:SOURce ECLTrg<n>.

[SOURce:]ROSCillator:SOURce<**INT**ernal|EXTernal|ECLTrg0|ECLTrg1>

Selects the source of the reference oscillator.

- **INT**ernal Selects the output of an internal frequency synthesizer.
- ECLTrg<n> Selects the signal from one of the VXIbus ECL Trigger Lines on the backplane.
- EXTernal Selects the signal from the CLOCK IN/OUT BNC. This BNC must be configured as an input using the command:

[SOURce:]CLOCk:CONFigure INPut

[SOURce:]SEQuence:ADVance <**SYNC**hronous|ASYNChronous>,<index>

Selects the conditions under which an advance is made to the next segment in the sequence where a trigger is required to initiate the advance.

- **SYNC**hronous Advances to the next segment after the completion of a complete cycle.
- ASYNChronous Advances to the next segment without necessarily completing the cycle.

The index ranges from 0 to 4095 as the Sequence Length ranges from 1 to 4096. The query form SEQuence:ADVance? <index> returns AUTO or TRIG for the segment selected by the index suffix.

[SOURce:]SEQuence:CRATio <numeric_value>,<index>

Defines the ratio between the programmed raster clock period and the point period of the segment. The raster clock period is multiplied by the valueof the clock ratio. This allows a segment to be played back at a lower raster frequency.

[SOURce:]SEQuence:DWELl <numeric_value>,<index>

Defines the repeat count, the number of times to cycle through the specified function in the function list. There is a one-to-one correspondence between elements in the function list and elements in the dwell list.

Programming the dwell to 0 will cause the function to be repeated indefinitely (continuous dwell), so a TRIGgered ADVance condition should be set up so that advance to the next segment will occur on a trigger. The index ranges from 0 to 4095 as the Sequence Length ranges from 1 to 4096.

The query form SEQuence:DWELl? <index> returns the numeric dwell value for the segment selected by the index suffix.

[SOURce:]SEQuence:FUNCtion <trace_name>,<index>

Defines a list of user defined functions which are to be sequenced through when the function mode is set to Sequence with the command [SOURce:]FUNCtion: MODE SEQuence.

In the Model 1385 the number of elements in this list is limited to four. The index value is set from 0 to 4095 as the [SOURce:]SEQuence:LENGth command ranges from 1 to 4096.

The query form SEQuence:FUNCtion? <index> returns the trace name for the segment selected by the index suffix.

[SOURce:]SEQuence:LENGth <numeric_value> **(1)**

Defines the number of traces in the sequence. The maximum sequence length is 4096.

[SOURce:]SEQuence:STARt <**AUTO**matic|TRIGgered>,<index>

Selects the conditions under which the segment starts playback. There is a one to one correspondence between elements in the function list and elements in the advance list.

- **AUTO**matic Plays back automatically.
- TRIGgered Waits for a trigger or first point before commencing playback.

```
[SOURce:]SEQuence:SEGment
      <trace_name>,<DWELl>,<CRATio>,
      <STARt>,<ADVance>,<index>
```

Sets up a sequence segment in one step. Combines all of the SOURce:SEQuence commands in one command.

- trace name Name of the trace to use in the segment.
- DWELl The repeat count of this segment.
- CRATio The clock multiplier ratio.
- STARt Automatic or triggered.
- ADVance Synchronous or Asynchronous.
- index The index of the segment.

```
[SOURce:]SWEep:COUNt
              <numeric_value> (1)
```

Determines the number of sweeps which are enabled by a single trigger event when Sweep is in a non-continuous mode. Default value is 1. Allowable values range from 1 to 1 million.

```
[SOURce:]SWEep:DIRection <UP|DOWN>
```

Controls the sweep direction. If UP is selected the sweep is performed in ascending order from STARt to STOP. If DOWN is selected the output frequency is swept from STOP to STARt.

```
[SOURce:]SWEep:MODE <CRESet|TRESet
    |HRESet|CREVerse|TREVerse
    |HREVerse|MANual>
```

Sets the mode of the sweep. The sweep modes have the following characteristics:

- **CRES**et Sweeps from the start frequency to the stop frequency and then returns to the start frequency in a continuous loop.
- TRESet Waits for trigger and then sweeps from the start frequency to the stop frequency and then resets to the start frequency.
- HRESet Waits for a trigger and then sweeps from the start frequency to the stop frequency and then waits for another trigger before returning to the start frequency.
- CREVerse Sweeps from the start frequency to the stop frequency and then sweeps back to the start frequency in a continuous loop.
- TREVerse Waits for a trigger and then sweeps from the start frequency to the stop frequency and then sweeps back to the start frequency.
- HREVerse Waits for a trigger and then sweeps from the start frequency to the stop frequency and then waits for another trigger before sweeping back to the start frequency.
- MANual Uses the frequency set by the `[SOURce:]FREQuency:MANual` command if it is within the range of frequencies set by the `[SOURce:] FREQuency:STARt` and `[SOURce:]FREQuency:STOP` commands.

```
[SOURce:]SWEep:SPACing
              <LINear|LOGarithmic>
```

Determines the frequency verses time characteristics of the sweep.

- **LIN**ear Output frequency is swept linearly between the STARt and STOP frequencies.
- LOGarithmic Output frequency is swept on a logarithmic curve fitted between the STARt and STOP frequencies. This objective of the logarithmic sweep is to spend equal amounts of time within each octave or decade of frequency.

```
[SOURce:]SWEep:TIME
                  <numeric_value>(1)
```

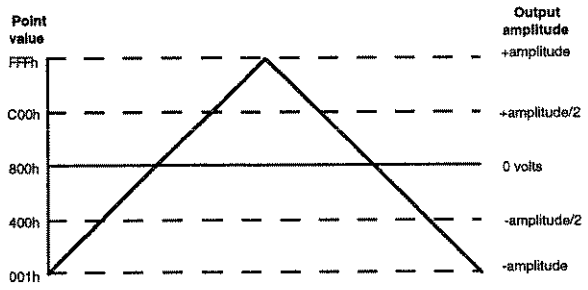
Sets the duration of the sweep in seconds. The sweep time may range from 30 ms to 1000 s.

[SOURce:]VOLTage[:LEVel][:IMMediate]
[:AMPLitude] <numeric_value> **(1)**

Sets the absolute value of the maximum amplitude voltage. Default is 1 Vp and allowable values range from 0 to 5.5 Vp.

The amplitude voltage is at maximum when the selected trace point is at its minimum or maximum value. The value of a point in trace memory affects the output amplitude in the manner shown in the preceding diagram.

Each point in Trace Memory contains a value in the range 000h (0) to FFFh (4095). As Trace Memory is scanned these values are converted to analog voltages for output. The ARB is calibrated so that the value 800h corresponds to 0 volts amplitude and the values 001h and FFFh correspond to the negative and positive full scale amplitude voltages.



Trace Point Value vs. Amplitude Nomengraph

All internally generated traces of a cyclical nature (SINusoid, SQUare, TRIangle, etc.) are generated such that their most negative point has a value of 001h and their most positive point has a value of FFFh. This makes their amplitude voltages symmetrical about 0 volts and the absolute value of their peak voltages equal to the programmed amplitude in Vp.

[SOURce:]VOLTage[:LEVel][:IMMediate]
:OFFSet <numeric_value> **(0)**

Controls the level of the output offset voltage. Allowable values are from -5.5 Vdc to +5.5 Vdc.

3.3.3.6 STATus Subsystem

Refer to Table 3-3e.

STATus:OPERation:CONDition?

Returns the contents of the Operation Condition Register. The Model 1385 supports this query, but will only return the value "0", indicating operational condition.

STATus:OPERation:ENABle <NRf>

Sets the enable mask of the Operation Event Register, which allows true conditions to be reported in the summary bit. The Model 1385 supports the command by saving the mask value and by not generating an error, although the Status registers do not exist.

The <NRf> notation indicates that SCPI's <numeric_ value> format is not used in this case. Refer to the IEEE-488.2 <DECIMAL NUMERIC PROGRAM DATA>, flexible Numeric Representation for more information.

The STATus:OPERation:ENABle? query returns the enable mask of the Operation Event Register. The 1385 returns the value sent previously with the command above using the <NR1> format.

STATus:OPERation[:EVENt?]

Returns the contents of the Operation Event Register. The 1385 supports this query, but will only return the value "0", indicating operational condition.

STATus:PRESet

Sets the enable registers to all 1's. The Model 1385 accepts the command without performing any action.

STATus:QUEStionable:CONDition?

Returns the contents of the Operation Condition Register. The 1385 supports this query, but will only return the value "0", indicating operational condition.

STATus:QUEStionable:ENABle <NRf>

Sets the enable mask of the Operation Event Register, which allows true conditions to be reported in the summary bit. The Model 1385 supports the command by saving the mask value and by not generating an error, although the Status registers do not exist.

The <NRf> notation indicates that SCPI's <numeric_ value> format is not used in this case. Refer to the IEEE-488.2 <DECIMAL NUMERIC PROGRAM DATA>, flexible Numeric Representation for more information.

Table 3-3e. STATus Keyword Excerpt From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
STATus		
:OPERation		
:CONDition		
:ENABle		
:ENABle?	<NRf>	
[:EVENt]		
:PRESet		
:QUEStionable		
:CONDition		
:ENABle		
:ENABle?	<NRf>	
[:EVENt]		

The STATus:OPERation:ENABle? query returns the enable mask of the Operation Event Register. The Model 1385 returns the value sent previously with the command above using the <NR1> format.

STATus:QUEStionable[:EVENt?]

Returns the contents of the Operation Event Register. The Model 1385 supports this query, but will only return the value "0", indicating operational condition.

3.3.3.7 SYSTem Subsystem

Refer to Table 3-3f.

SYSTem:ERRor?

Returns the next message from the system error queue. With each query, the unit returns a number followed by a brief description. The error queue holds up to eight errors, with one returned for each query sent, until the queue is empty. Table 3-4 describes the system error messages.

SYSTem:DATE <year>,<month>,<day>

Sets the system date.

SYSTem:DATE?

Returns the system's idea of the date in the following format:

<year>,<month>,<day>

SYSTem:TIME <hour>,<minute>,<second>

Sets the system time.

SYSTem:TIME?

Returns the system's idea of the time in the following format:

<hour>,<minute>,<second>

SYSTem:VERSion?

Returns the system's firmware version number using the following format:

<manufacturer>,<model>,
<serial_number|0>,<firmware_level|0>

Table 3-3f. SYSTem Keyword Excerpt From Table 3-2

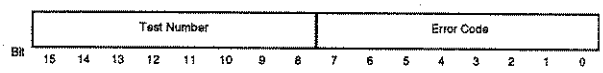
KEYWORD	PARAMETER FORM	NOTES
SYSTem		
:ERRor?		
:DATE	<year>,<month>,<day>	
:TIME	<hour>,<minute>,<second>	
:VERSion?	<manufacturer>,<model>,<serial number |0>,<firmware_level|0>	

3.3.3.8 TEST Subsystem

The TEST subsystem is an area where device specific commands can be added to facilitate testing. The `TEST[:ALL]?` query in this subsystem performs the same SELF TEST function as the IEEE-488.2 Common Command `*TST?`

`TEST[:ALL]?`

Performs a non-destructive test on the application card hardware. A result of zero is returned if the test passed. A non-zero result is returned if the test failed. Interpretation of the result code will indicate the nature of the error. The format of the 16-bit result code is shown below:

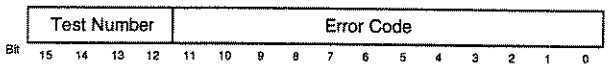


TEST[:ALL?] Response Format

The upper 8-bits of the result code contain the sub-test number in which a failure was detected. The lower 8-bits contain a bit-weighted error code that indicates the exact cause of the failure. The sub-test descriptions and the meanings of their result codes are documented in Appendix B.

`TEST:RAM?`

Performs a destructive test of the Trace Memory. A result of zero is returned if the test passed. A non-zero result is returned if the test failed. Interpretation of the 16-bit result code will indicate the nature of the error. Bits 15 through 12 encode the sub-test number on which the first failure was detected. The interpretation of the rest of the bits in the response depends on the sub-test. Refer to the Trace Memory test documentation in Appendix B for a full interpretation of the Error Code.



Trace Memory Test Response Format

Table 3-3g. TEST and TRACe Excerpts From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
TEST [:ALL]? :RAM?		
TRACe :CATalog? :DEFine :DATA :LINE :POINt :DELete [:NAME] :ALL :DIRectory? :FREE? :LIMits :MODE :POINts	 <trace_name>,<trace_name>,. . . <trace_name>,<numeric_value>|<trace_name> <trace_name>,<block>|<trace_name> <trace_name>,<point_index1>,<point_value1>, <point_index2>,<point_value2> <trace_name>,<point_index>,<point_value> <trace_name> <name>,<size>,<limit>,<limit>. . . <numeric_value>,<numeric_value> <trace_name>,<start_index>,<stop_index> <CW|RASTer> <trace_name>,<numeric_value>	 Up to 50 traces. value is new trace size (arbitrary block data) Up to 256K for standard functions Up to First 50 traces Points available, in-use Value is new trace size

3.3.3.9 Trace Subsystem

Refer to Table 3-3g.

`TRACe:CATalog?`

Returns a string containing the names of all defined traces. Trace names are separated by commas. If there are more than 50 traces, only the first 50 traces will be reported.

`TRACe[:DATA]`
`<trace_name>,(<block>|<trace_name>)`

Initializes the contents of the trace whose name is specified by the first parameter. The second parameter may be binary data in Definite Length Arbitrary Block Data format**, the name of another defined trace, or the name of one of the standard functions. This command operates on the portion of the destination trace within its "trace limits" as set by the `TRACe:LIMits` command.

When another trace is specified as the data source the source data is scaled horizontally to fit the destination trace. Data is copied from the area of the source trace specified by its trace limits to the area of the destination trace specified by its trace limits.

When a standard function is specified as the data source it is scaled horizontally to fit the area of the destination trace specified by its trace limits.

When the pseudo-standard function SMEMory is used as a source, data is copied from the base address of the A24 Shared Memory point by point up to the number of points needed to fill the area of the destination trace specified by its trace limits.

Note

***See paragraph 3.3.3.11 for a description of the IEEE-488.2 Definite Length Arbitrary Block Data format.*

`TRACe[:DATA]? <trace_name>`

Returns the contents of the trace whose name is specified by the first parameter in Definite Length Arbitrary Block Data format**. Only the data contained in the portion of the trace set by the trace limits is returned.

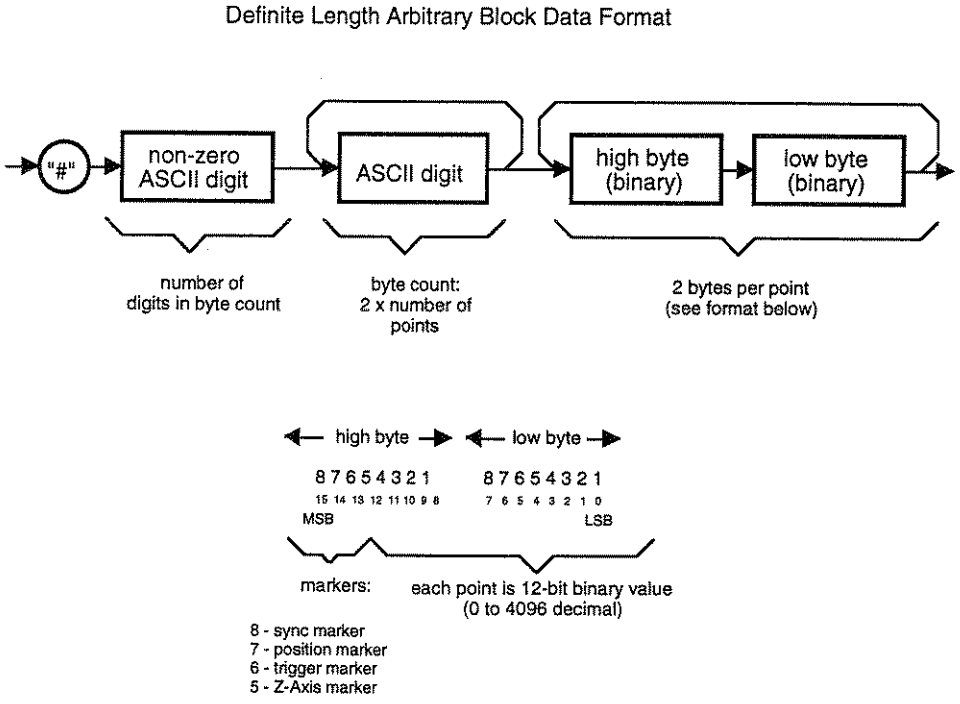


Figure 3-3. Definite Length Arbitrary Block Data Format

```
TRACe[:DATA]:LINE <trace_name>,
  <point_index1>,<point_value1>,
  <point_index2>,<point_value2>
```

Draws a line segment within the boundaries of the trace whose name is specified by <trace_name>. <point_index1> is the index of the start point and <point_value1> is its decimal value. <point_ index2> is the index of the end point and <point_ value2> is its value. A line is drawn connecting the start point and end point. The index of the first point in a trace is 0. The point value is an integer ranging from 0 to 4095.

```
TRACe:DATA:POINt <trace_name>,
        <point_index>,<point_value>
```

Sets the point specified by <point_index> in the trace specified by <trace_name> to the value specified by <point_value>. The index of the first point in a trace is 0. The point value is an integer ranging from 0 to 4095.

```
TRACe:DATA:POINt?
        <trace_name>,<point_index>
```

Returns the value of the point specified by <point_ index> in the trace specified by <trace_name>. The index of the first point in a trace is 0.

```
TRACe:DEFine <trace_name>,
   (<numeric_value>|<trace_name>)
```

Creates a new trace with the name specified by the first parameter. The second parameter may be a numeric value indicating the size of the new trace or it may be the name of another trace which is to be copied.

```
TRACe:DELete[:NAME] <trace_name>
```

Deletes the specified trace.

```
TRACe:DELete:ALL
```

Deletes all traces.

```
TRACe:DIRectory?
```

Returns a string containing the names, sizes and limits of all traces up to the first 50 traces. The format of the response is as shown below:

```
<name1,size1,start1, stop1;
name2, size2, start2, stop2;
... , stopn>
```

```
TRACe:FREE?
```

Returns the number of trace memory points in use and the number of trace memory points available. The format of the response is as shown below:

```
<points_available>,<points_in_use>TRACe
:LIMits <trace_name>,<start_index>,
<stop_index>
```

Sets the playback limits of the trace whose name is specified by <trace_name>. The second parameter is the index of the start point and the third parameter is the index of the end point. The value of the end point must be greater than the value of the start point plus eight. Both points must be within the trace boundaries. Trace point indices range from 0 to (size -1).

```
TRACe:LIMits? <trace_name>
```

Returns the playback limits of the trace whose name is specified by <trace_name> in the following format:

```
<start_index>,<stop_index>
```

```
TRACe:MODE <CW|RASTer>
```

Sets the trace playback mode. If CW is selected a fixed 50 MHz scan rate is used and phase accumulation is used for frequency control. Frequency is controlled using the :FREQuency[:CW] command.

CW stands for Continuous Wave and implies the signal being played back is phase continuous like a sine wave. CW mode is useful primarily for playing back standard functions like sinusoid, triangle and square.

Because of the fixed scan rate the 20 MHz filter can be turned on to remove the 50 MHz sampling noise and thus generate spectrally pure functions.

If RASTer is selected the scan rate can be adjusted by the selected reference oscillator. The raster scan frequency is controlled by the :FREQuency :RASTer command. Raster mode would be used to play back arbitrary waveforms which typically define a complex pattern of amplitude verses time.

The user may not want to use the filter in this mode because arbitrary waveforms may have step or pulse patterns that are meant to have sharp edges.

```
TRACe:POINts <trace_name>,
                <numeric_value>
```

Re-sizes the trace whose name is specified by <trace_name>. The new size is specified by the second parameter.

```
TRACe:POINts? <trace_name>
```

Returns the size of the specified trace.

3.3.3.10 TRIGger Subsystem

Refer to Table 3-3h.

`TRIGger:COUNt <numeric_value> (1)`

This command sets the number of times to cycle through a trace after a trigger is received. Default value is 1. The COUNt value ranges from 1 to 1,048,575 for triggered waveforms (`FUNCtion:MODE FIXed`) and from 1 to 65536 for triggered sequences (`FUNC-tion:MODE SEQuence`).

`TRIGger:GATE[:STATe] <ON|OFF>`

This command selects a gated mode of operation when the selected trigger source is external.

`TRIGger[:IMMediate]`

This command immediately triggers the instrument, independent of which trigger source is selected.

`TRIGger:POLarity <POSitive|NEGative>`

This command selects the active trigger level.

`TRIGger:SOURce[:STARt] <EXTernal|INTernal|CHAin|ECHAin|TTLTrg<n>>`

Selects the source of the STARt trigger signal. The STARt trigger signal is used to initiate activity when the instrument is in a triggered mode of operation. It also initiates the start of the first pass through a Sequence in Continuous Mode.

- EXTernal Selects the external TRIG IN BNC as the trigger source.
- **INT**ernal Selects an instrument dependent internal signal as the trigger source.
- CHAin Selects the CHAIN trigger signal from the local bus. This signal is used to receive a trigger signal from an adjacent module to the left in a chain.
- ECHAin Selects the END CHAIN trigger signal from the local bus. This signal is typically used by the first (left-most) module in a chain to receive a trigger signal from the last module in the chain.
- TTLTrg<n> Selects one of the VXIbus TTL Trigger Lines from the backplane. Valid numeric suffixes are in the range 0 through 7.

`TRIGger:SOURce:ADVance <EXTernal|INTernal|CHAin|ECHAin>`

Selects the source of the ADVance trigger signal. The ADVance trigger signal is used to advance a Sequence from segment to segment. The source selection definitions are the same as for the STARt trigger.

`TRIGger:TIMer <numeric_value>(1e-3)`

Sets the period of an internal periodic signal source. The timer signal acts as a trigger when the selected trigger source is INTernal. Default value is 1E-3. It ranges from 2e-7 to 1e4 seconds with 2e-7 resolution.

Table 3-3h. TRIGger Keyword Excerpts From Table 3-2

KEYWORD	PARAMETER FORM	NOTES
TRIGger		
:COUNt	<numeric_value>	1, **1**, 1048575 for traces 1, **1**, 65536 for sequences
:GATE		
[:STATe]	<Boolean_data> (**OFF**)	
[:IMMediate]		
:POLarity	<**POS**itive|NEGative>	
:SOURce		
[:STARt]	<**INT**ernal|EXTernal|CHAin|ECHain|TTLTrg<n>>	<n> = 0 to 7 VXIbus
:ADVance	<**INT**ernal|EXTernal|CHAin|ECHain>	TTLTrigger lines
:TIMer	<numeric_value>	(2e-7, **1e-3**, 1e4)

3.3.3.11 Binary Waveform Transfer

The Model 1385's SCPI command `TRACe[:DATA] <trace_name>,<block>` is used to download user-defined Waveforms from the remote controller to the ARB. Likewise, the query form `TRACe[:DATA]? <trace_name>` is used to upload the waveform data back up to the controller.

In both of these cases, the data block is transferred using the IEEE-488.2 Definite Length Arbitrary Block Data format (see figure 3-3). This format for block data transfer makes it possible to rapidly move the large amount of data required for arbitrary waveforms.

To send a block of waveform data, send an ASCII "#" ($23), then an ASCII encoded digit whose value signifies the number of digits in the byte count, then ASCII encoded digit(s) representing the byte count, then the two byte binary data words (MSB first). The byte count is twice the number of points to be downloaded to the trace.

The byte count must exactly correspond to the number of bytes in the block of data. Each point is composed of two bytes representing a 12-bit unsigned integer between 0 and 4095.

Prior to downloading a Waveform using the `TRACe[:DATA] <trace_name>, <block>` command, the Trace must be first defined. Send the `TRACe:DEFine <trace_name>, <value>` command to define a Trace with a size of `<value>` points under the `name <trace_name>`.

This also presets the Trace Limits for this trace at full size. You may select a segment of this trace for download (or subsequently, for upload) using the `TRACe:LIMits <trace_name>, <start_index>,<stop_ index>` command.

Note

When the block size exceeds the capability of your download/upload application, you may use the TRACe:LIMits feature to break the block up into manageable segments.

The "binary" transfer using this format occurs at a relatively high speed because the binary data is not parsed through the 1385's Command Processor. Instead, the binary data is routed directly to the Trace RAM without processing or limit checking, much like a direct memory access (DMA).

If the waveform limits (size), the byte count or the number of bytes in the `<block>` are not all in numeric agreement, the high speed transfer will be aborted. Any data received after this will be interpreted by the Command Processor as ASCII characters and will cause the Model 1385 to generate many error messages.

3.3.3.12 Shared Memory Waveform Download

Shared Memory Waveform Download: `TRACe[:DATA] <trace_name>,SMEM`. This command transfers the data from the A24 shared memory to the waveform trace memory. You must first set up the waveform using the trace:define command. When the TRACe:DATA <trace_name>,SMEM command it transfers the data in the shared memory directly into the waveform data memory. See section 3.4.6.2.

Table 3-4. Error Messages

Error Number	Message	Error Number	Message
0	"No error"	-222	"Data out of range"
-100	"Command error"	-223	"Too much data"
-101	"Invalid character"	-224	"Illegal parameter value"
-102	"Syntax error"	-230	"Data corrupt or stale"
-103	"Invalid separator"	-231	"Data questionable"
-104	"Data type error"	-240	"Hardware error"
-105	"GET not allowed"	-241	"Hardware missing"
-108	"Parameter not allowed"	-250	"Mass storage error"
-110	"Command header error"	-251	"Missing mass storage"
-111	"Header separator error"	-252	"Missing media"
-112	"Program mnemonic too long"	-253	"Corrupt media"
-113	"Undefined header"	-254	"Media full"
-114	"Header suffix out of range"	-255	"Directory full"
-120	"Numeric data error"	-257	"file name error"
-121	"Invalid character in number"	-258	"Media protected"
-123	"Exponent too large"	-260	"Expression error"
-124	"Too many digits"	-261	"Math error in expression"
-128	"Numeric data not allowed"	-270	"Macro error"
-130	"Suffix error"	-271	"Macro syntax error"
-131	"Invalid suffix"	-272	"Macro execution error"
-134	"Suffix too long"	-273	"Illegal macro label"
-140	"Character data error"	-274	"Macro parameter error"
-144	"Character data too long"	-275	"Macro definition too long"
-148	"Character data not allowed"	-276	"Macro recursion error"
-150	"String data error"	-277	"Macro redefinition not allowed"
-151	"Invalid string data"	-278	"Macro header not found"
-158	"String data not allowed"	-280	"Program error"
-160	"Block data error"	-281	"Cannot create program"
-161	"Invalid block data"	-282	"Illegal program name"
-168	"Block data not allowed"	-283	"Illegal variable name"
-170	"Expression error"	-284	"Program currently running"
-171	"Invalid expression"	-285	"Program syntax error"
-178	"Expression data not allowed"	-286	"Program run time error"
-180	"Macro error"	-300	"Device specific error"
-181	"Invalid outside macro definition"	-310	"System error"
-183	"Invalid inside macro definition"	-311	"Memory error"
-184	"Macro parameter error"	-312	"PUD Memory lost"
-200	"Execution error"	-314	"Save/recall memory lost"
-201	"Invalid while in local"	-315	"Configuration memory lost"
-202	"Settings lost due to rtl"	-330	"Self test failed"
-210	"Trigger error"		

3.3.4 IEEE-488.2 Common Commands

The *CAL? self calibrate query, the *TST? self test query, and the *TRG command are discussed elsewhere in this manual (along with their equivalent SCPI query or command).

The *CAL? query is equivalent to the SCPI CALibration[:ALL]? query. The self calibration query is discussed in Appendix A of this operator's manual.

The *TST? query is equivalent to the SCPI TEST[:ALL]? query. The self test query is discussed in Appendix B of this operator's manual.

The *TRG is an IEEE Common Command used to provide a properly sequenced trigger and execute to an addressed IEEE-488.2 device. It triggers the Model 1385 via the VXI data bus.

The previous paragraphs describe in detail the three most commonly used IEEE Common Commands. Table 3-5 briefly describes the messages mandated by the IEEE-488.2 standards, plus optional commands supported by the Model 1385.

3.4 Model 1385 Operation

The following paragraphs describe messages for various modes of operation for the Model 1385.

Before beginning, review the data in paragraphs 3.4.1, 3.4.2, and 3.4.3.

3.4.1 Output Terminations

Each output connector must be properly terminated during its use to minimize signal reflection or power loss due to an impedance mismatch. Figure 3-4 shows proper 50Ω termination for the MAIN OUT

Table 3-5 IEEE 488.2 Common Commands

Command	Function	Description
*CAL?	Calibration Query	Starts self-cal, places pass/fail response in output queue
*CLS	Clear Status Command	Clears Status Data Registers, forces OCIS/OQIS
*ESE	Standard Event Status Enable	Sets Event Status Enable Register bits
*ESE?	Standard Event Status Enable Query	Returns contents of Event Status Enable Register
*ESR?	Standard Event Status Register Query	Returns contents of Event Status Register
*IDN?	Identification Query	Identifies devices over the system interface
*OPC	Operation Complete Command	Requires oper. comp. message in Event Status Reg.
*OPC?	Operation Complete Query	ASCII '1' in dev. out. queue when operations complete
*RCL	Recall Command	Restores device setup from local memory
*RST	Reset Command	Resets the device
*SAV	Save Command	Stores current device setup to local memory
*SRE	Service Request Enable Command	Sets Service Request Enable Register bits
*SRE?	Service Request Enable Query	Returns contents of Service Request Enable Register
*STB?	Read Status Byte Query	Returns status and master summary status bytes
*TRG	Trigger Command	Initiates a properly sequenced trigger and execute
*TST?	Self Test Query	Starts self-test, places pass/fail response in output queue
*WAI	Wait - to - Continue Command	Blocks device commands until 'No-Op-Pend' flag is true

connector, with R_S representing the Model 1385 source impedance, R_L representing the termination or load resistance, and R_{IN} representing the receiving instrument input impedance. Table 3-6 lists all the input and output impedances of the Model 1385.

Table 3-6 Input and Output Impedances

Connector	Impedance
CLK IN	2 kΩ shunted by 10 pF
CLK OUT	50Ω, TTL (0 to >2.4 V terminated)
TRIG IN	2 kΩ
POSITION OUT	50Ω, TTL (0 to >2.4 V terminated)
SYNC OUT	50Ω, TTL (0 to >2.4 V terminated)
SWEEP OUT	600Ω
AM IN	10 kΩ
MAIN OUT	50Ω

3.4.2 Input/Output Protection

The Model 1385 provides protection for internal cicuitry connected to input and output connectors. Refer to the Specifications in Section 1 of this manual to determine the level of protection associated with each input or output connector.

3.4.3 Power On/Reset Defaults

At power on, or as the result of sending `RESet` or `*RST` (except that the Trace Subsystem is unaffected by a reset), the Model 1385 defaults to the following conditions:

Subsystem	SOURce
Operational Mode	CONTinuous
Trace Mode	CW (vs. RASTer)
Function Mode	FIXed (SEQuence disabled)
Phase Lock	OFF
Phase value	0 degrees
Reference Oscillator	INTernal
Frequency Mode	CW (Sweep Off)
Frequency value	1 kHz (at MAIN OUT)
Amplitude value	2 Vpp (1 Vp)
Offset value	0 Vdc
Function	SINusoid
Main Output	OFF
Waveform Filter	20 MHz ELLiptic, ON
Sync Marker Output	ON, ZCRoss (zero crossing)
Position Marker Output	ON, markers AOFF (all off)
Clock Input/Output	INTernal Input, disabled (OFF)
AM Input	Disabled (OFF)
TTL Trigger Lines	OFF (not input or output)
ECL Trigger Lines	OFF (not input or output)
Trigger Slope	POSitive
Trigger Source	INTernal
Trigger Timer	1 ms
Trigger Count	1
Sweep Mode	ContinuousRESet
Sweep Start value	1 kHz
Sweep Stop value	100 kHz
Sweep Direction	UP
Sweep Spacing	LINear
Sweep Time	1 s
Sweep Count	1
Sequence Advance	SYNChronous
Sequence Length	1
Sequence Dwell	1

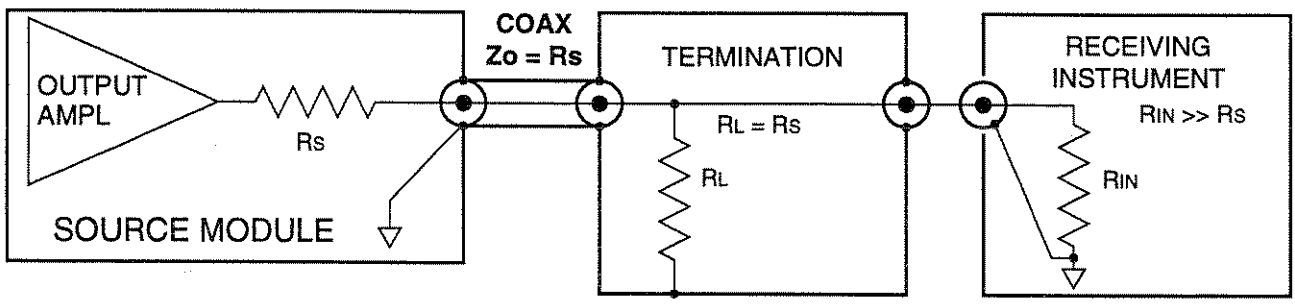


Figure 3-4. Output Termination

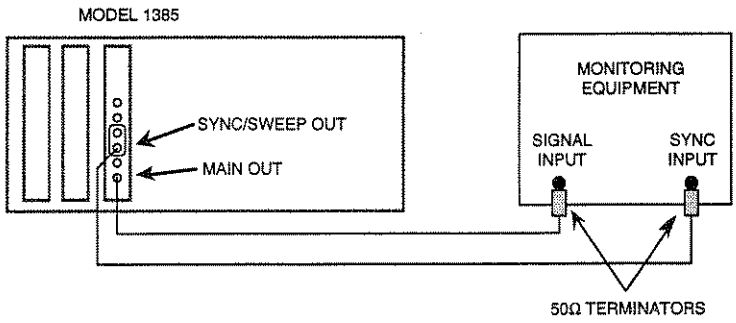


Figure 3-5. Model 1385 Basic Operation Setup

3.4.4 Standard Functions (CW)

"Example 1" in this paragraph shows how to set up the Model 1385 for a continuous, 10kHz, 1.5Vpp sine wave, with −1.5Vdc offset at the MAIN OUT connector. All of the parameters shown here are detailed in earlier paragraphs. Much of the information given here directly transfers to other modes.

Connect the Model 1385's output (terminated) to the device under test, as shown in figure 3-5. Use the SYNC/SWEEP OUT signal as a synchronizing source. Figure 3-5 illustrates the SYNC/SWEEP OUT and MAIN OUT relationships.

Example 1:

The following discussion provides a complete example for Continuous Mode operation of the Model 1385, and it applies to a particular VXIbus system setup as follows:

- IBM PC compatible host.
- National Instruments GPIB card and drivers.
- National Instruments "IBIC" or "WIBIC" application software for interactive GPIB control.
- VXIbus chassis with Wavetek 1320 Resource Manager, or equivalent.
- Wavetek 1391, 1375, 1395 or 1385 acting as the external source in slot 1, set to secondary address 1**.
- Wavetek 1385 acting as the Unit Under Test (UUT) in slot 2, set to secondary address 2.

**Will be used in later examples.

The GPIB card, "gpib0", and the information file "gpib.ini" are set up according to National's instructions. The two devices, "dev1" and "dev2", are set up with the Wavetek 1320's primary address and the secondary address for the modules in slot 1 and slot 2, respectively. In the following programming steps, the information in **bold** is the WIBIC prompt. The remainder of the string is to be typed in by the programmer.

The first line identifies the GPIB card. The second line identifies the Model 1385 UUT. The third line turns on both the MAIN OUT and SYNC/SWEEP outputs.

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "outp on;:mark:sync on"
```

When viewed on an oscilloscope, the MAIN OUT is the default 1 kHz, 2 Vp-p sine waveform, and the synchronization waveform is a TTL level square wave which would result from "zero-crossing" this sine wave with a comparator.

Each of the Model 1385's "Standard" waveforms (DC, HFSQuare, NHSine, NRAMp, PHSine, PRAMp, PRNoise, SINusoid, SQUare, or TRIangle) can now be selected as in the following command:

```
dev2:ibwrt "func pram"
```

The selected Standard function will now be a positive-going ramp. Observe any desired function from the list and compare the waveforms and synchronization signal to Figure 3-6.

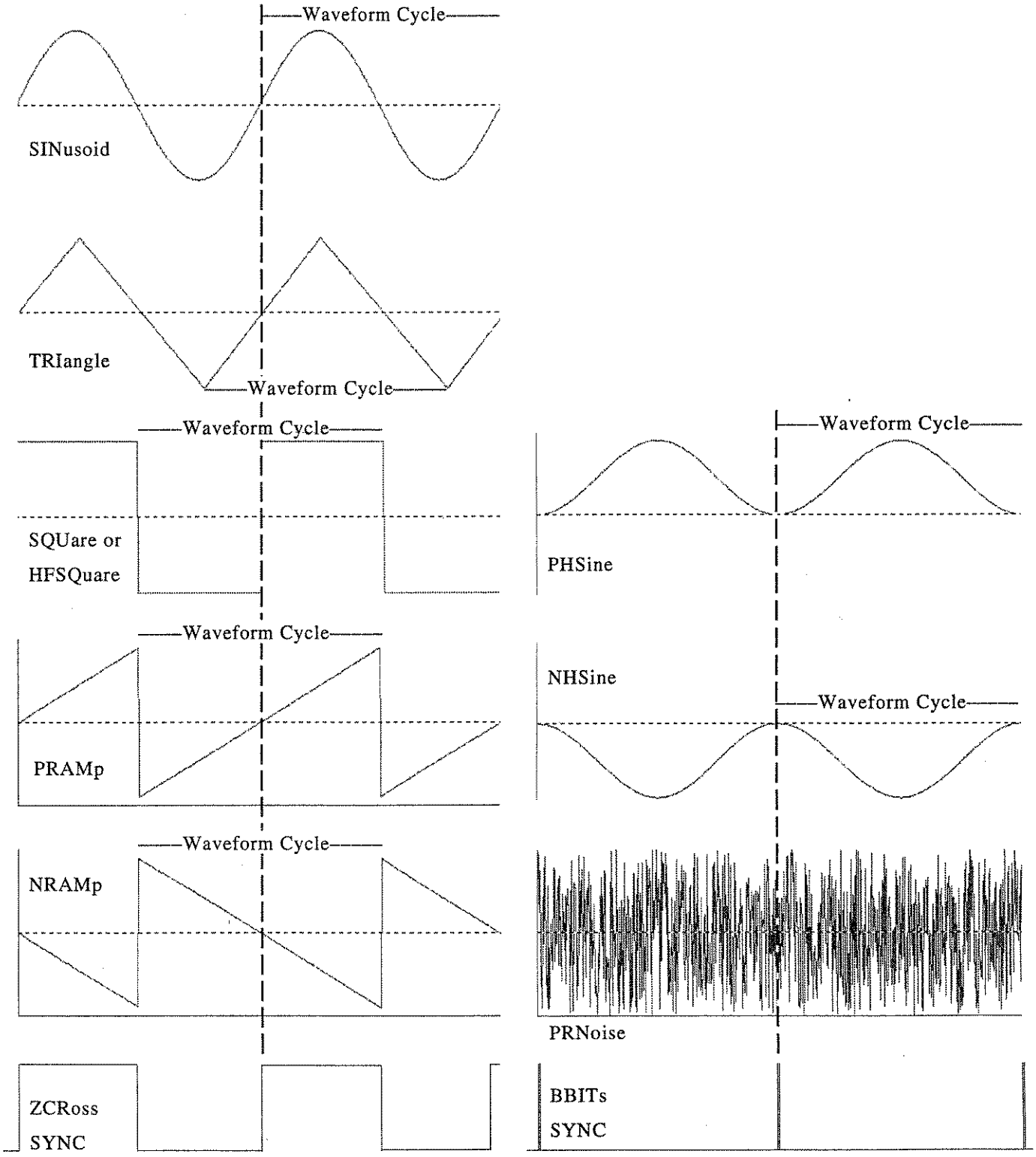


Figure 3-6. Continuous Waveform Characteristics

Selection of the Haversine or Noise functions will require selecting the "BBITs" rather than the "ZCRoss" synchronization signal (see paragraph 3.4.10), as follows:

```
dev2:ibwrt "func phs;mark:sync
                        :sour bbit"
```

Reselect the sine wave when through observing the Standard functions. Set the output signal's Vpp amplitude and Vdc offset to the values for this example by sending the following command:

```
dev2:ibwrt "volt 0.75;volt
                        :offs -1.5"
```

This command sets the "amplitude" of the selected trace to 0.75 Vp (1.5 Vpp) and the "dc offset" to -1.5 Vdc. In reality, any points within the trace set to the minimum value of "001_{hex}" will be at -Vp below the offset value; those set to "800_{hex}" will be at the offset value; and those set to "FFF_{hex}" will be at +Vp above the offset value. It is the combination of the amplitude/offset settings and the instantaneous trace point value that determine the output voltage.

Set the waveform frequency to 10 kHz with the following command:

```
dev2:ibwrt "freq 1e4"
```

3.4.5 Trace Operations and USER Function (RAST)

The Trace Subsystem commands have been described in paragraph 3.3.3.9. The Trace Subsystem is used to define, enter data into, and manipulate arbitrary waveforms. Trace Operations are concerned with the management of the 1385's Trace Memory. Trace Memory is volatile, high-speed RAM organized as 131,072 (optionally 524,288) addressable points.

3.4.5.1 Trace Definition

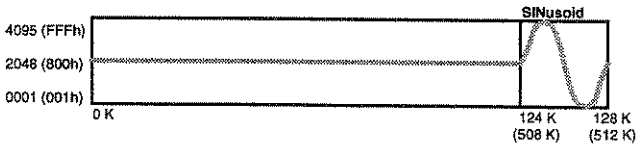
An arbitrary waveform is referred to in the SCPI language as a Trace. A Trace is a block of contiguous points in the Trace Memory which are referenced by a <trace_name>. The "size" of the Trace is the number of points reserved for waveform data. Each data point uses 16 bits, with 12 bits used to describe the "vertical" position of a point (a value of "001_{hex}" will be at the lowest value; a value of "800_{hex}" will be at the median value; and a value of "FFF_{hex}" will be at the maximum value), and the remaining four bits to be used as markers.

The previous paragraph introduced a special case of the Trace, referred to as the Standard Function. Standard Functions have neither their <shape_name>, their waveform data, their size, nor their position in the Trace Memory arbitrarily assigned. When the `FUNCtion <shape_name>` command is processed, and the <shape_name> is one of the 10 Standard Function reserved names, then the appropriate waveform data is calculated and placed in the final 4K points of Trace Memory (unless Trace Memory has less than 4K unreserved points remaining). For Standard Functions, the TRACe MODE is set to CW (see Section 4 for Phase Accumulator theory of operation) and *waveform frequency* is determined by the `[SOUR:]FREQ[:CW|FIX] <value>` command.

Traces defined under the Trace Subsystem and selected for playback using the `FUNCtion USER` command are *arbitrary waveforms*. These Traces are played back in the "Raster" (vs. phase accumulator) mode of waveform synthesis. In Raster Mode, each point in the waveform is accessed sequentially (by the sample time clock) and held for one clock cycle. This assures that each data point value is accessed once each pass through the waveform.

Raster Mode is selected with the `TRACE:MODE RASTer` command and the *sample frequency* is set with the `[SOURce:] FREQuency:RASTer <value>` command. Trace frequency is then determined by dividing the Raster clock frequency by the number of points in the Trace. Conversely, the Trace period is determined by multiplying sample time (time spent at each point is the inverse of sample frequency) by the number of points in the Trace.

When the Model 1385 is powered on, the Trace Memory is blank except for the default sine wave created in the final 4K points. The Trace Mode is CW (phase accumulate), the Function is SINusoid, the sample frequency is 50 MHz, the waveform frequency is 1 kHz, the Sync Marker Source is Zero CRoss, and the 20 MHz Elliptic Filter is on. The Trace Memory is illustrated in the following drawing:



Example 2:

Set up the VXIbus system as described in Example 1. Continue with commands in this example over the next few paragraphs under the "Trace" headings.

The 1385's default configuration can be observed by cycling power, setting up the Model 1385 and an oscilloscope per Figure 3-5, and then sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "outp on;mark:sync on"
```

In this paragraph new Traces will be Defined, which gives them a name and reserves a block of points in Trace Memory according to the new Trace's size. When the first Trace is defined, it goes to the start of Trace Memory (from memory address 0 to memory address *size - 1*). The next Trace defined will be placed directly after the first TRACE, and so on such that *used* memory is always contiguous from the start of memory and *free* memory is contiguous from the end of the last Trace to the end of available memory. This way, the Standard Functions are always available (using the last 4K of free memory) until there is less than 4K of memory remaining. The Model 1385 firmware manages the start and stop addresses of each Trace so that this simple model is maintained, even as Traces are deleted or resized in subsequent Trace Subsystem operations.

Trace names have a maximum length of 12 characters. The names cannot be the names of Standard Functions, as these are *reserved names*. Trace names must start with an Alpha character and may be composed of either upper-case or lower-case characters. The only other characters permitted are the numerals (0-9) and the underscore (_) character. No short-form of the Trace name is permitted. The Trace Directory can hold up to 50 Trace names, although new Traces cannot be defined once the Trace Memory is filled.

The command to define a new Trace has two forms:

```
TRAC:DEF <1st_name>,<2nd_name>
TRAC:DEF <name>,<size>
```

The first command can be used to define a new Trace <1st_name>, and at the same time copy the data from the Trace or Standard Function <2nd_name> to the new Trace. The new Trace will be defined using the size of the Trace <2nd_name>. If <2nd_name> is the name of a Standard Function, the new Trace will be created with a default size of 8K (8192 points).

The second form of the command creates a new Trace with the given name and size. This Trace will not yet have any waveform data. The Trace Memory has default data values of 2048 (800 in hexadecimal), which is half-scale (zero amplitude).

Using the first form of the Trace Define command, the SINusoid Standard Function can be used as a source of data to create the Trace "SIN1" in the following manner:

```
dev2:ibwrt "trac:def sin1,sin"
```

Use the following commands to observe this newly created Trace:

```
dev2:ibwrt "func:user sin1"
dev2:ibwrt "func user"
```

The Model 1385 will make several other programming changes automatically because of the shift from a Standard to a User Function. The Trace Mode will shift from CW to Raster, the Sync Source will shift from ZCRoss to BBITs, and the Filter will turn off (Filter Type remains Elliptic). The Raster (sample clock) frequency will be at its default value of 50 MHz, so the SIN1 output frequency will be 50 MHz divided by 8192 or 6.104 kHz. Verify that the oscilloscope indicates the change to an approximate 6 kHz (~160 µs) sine wave and that the sync signal uses BBITs. Then send the Trace Directory query:

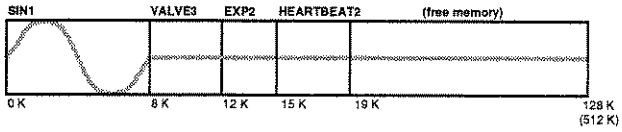
```
dev2:ibwrt "trac dir?"
dev2:ibrd 20
```

The response should be "SIN1,8192,0,8191". This response is interpreted to mean that there is one Trace in memory with the name "SIN1", with a size of 8192 points, a *start address* of 0 and a *stop address* of 8191. The true start and stop address are managed by the instrument firmware, and may not be viewed by the user/programmer. From the user's point of view, it is best to consider each Trace to have a *relative* start address of 0 and a *relative* stop address of (size -1). Within this relative address range, segments of the Trace may be selected for various operations using the Trace Limits (see paragraph 3.4.5.4).

The second form of the Trace Define command can now be used to set up the Traces which will be used in the remainder of this operational description, as follows:

```
dev2:ibwrt "trac:def
valve3,4000;def exp2,3000;def
heartbeat2,4000"
```

Verify that the Trace Memory looks like the following figure by sending the TRAC:CAT? query and verifying that the response message is the names of the four traces.



3.4.5.2 Trace Data

Note that three of the four Traces in the above figure have a name and address space reserved for them, but the waveform data at each point is simply the default half-scale value. This paragraph is concerned with a few of the methods for getting data into the defined traces using the Trace Data commands. These methods may be characterized as "pre-defined", "user-defined", and "Block Download" (see paragraph 3.4.6).

"Pre-defined" data entry involves the Standard Functions. This is often a convenient starting point from which the user can make alterations. In the previous paragraph the SIN1 waveform was both defined and loaded with an 8K point SINusoid using the Trace Define command and the default size. VALVE3 was defined with a size of 3000 points. This Trace can be given "square wave" data with the command:

```
dev2:ibwrt "trac:data valve3,squ"
```

```
dev2:ibwrt "func:user valve3"
```

The second command selects the USER waveform as VALVE3 for viewing on the oscilloscope. The square wave represents the simplest on/off representation of a valve control signal. The waveform could be given greater complexity through subsequent Trace Data operations, such as Point and Line editing.

Point editing is far too tedious for editing more than a few points, but occasionally it is the only way to get exactly the desired results. For example, the rising edge of the VALVE3 waveform could be "softened" on a point-by-point basis without attempting to define the transition with an expression. Make sure the oscilloscope is externally triggered on the positive-going slope of the BBITs synchronization signal and set the horizontal sweep time to 50 ns/DIV to observe the VALVE3 rising transition. Send the following point data:

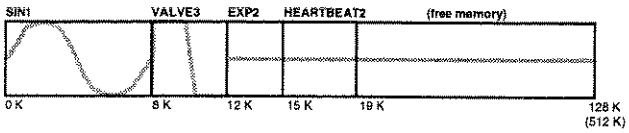
```
dev2:ibwrt "trac:data:poin valve3,
0,1000;poin valve3,1,2000;poin
valve3,2,3000;poin valve3,3,3500;
poin valve3,4,4000;mark:sync:sour
zcr;sour bbit"
```

The first five sub-commands send the data point-by-point, and the last two sub-commands toggle the synchronization "source" from BBITs to ZCRoss and back to BBITs. This restores the BBITs signal which is lost when relative address zero is written into.

Line editing is the second form that the Trace Data command can take. VALVE3 can be used to demonstrate line editing. Make sure the oscilloscope is externally triggered on the positive-going slope of the BBITs synchronization signal and set the horizontal sweep time to 10 µs/DIV to observe the VALVE3 falling transition at approximately four divisions. Since VALVE3 was originally defined as a 4000 point square wave, the first 2000 points should have had the data value of "4095" and the last 2000 points should have the data value of "1". Modify the falling transition with a line segment by sending the following:

```
dev2:ibwrt "trac:data:line
valve3,1900,4095,2100,1"
```

The edited VALVE3 waveform should now appear in Trace Memory as follows:



Block editing is the third and final form that the Trace Data command can take. The Block data is in the form of an IEEE-488.2 *Definite Length Arbitrary Block Transfer* from a computer. Refer to paragraph 3.4.6 for data on the block transfer.

3.4.5.3 Trace Copy, Resize, Rename, and Delete

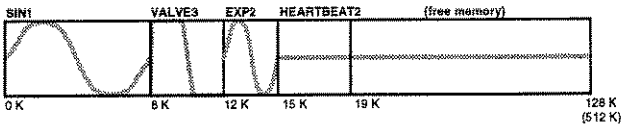
The SIN1 and VALVE3 examples demonstrate Copying a Standard Function to a new Trace. This same process can also Copy an existing USER Function to a new Trace. The destination Trace must be first defined so that it has a name and a size. The destination Trace size may be smaller, equal to, or larger than the source Trace. The Model 1385 uses linear interpolation to remap the data from a source Trace to a destination Trace with a differing size. The interpolation routine attempts to maintain waveshape integrity as much as possible. Copy the 8K SIN1's data to fill the 3K User waveform EXP2 using the following command:

```
dev2:ibwrt "trac[:data] exp2,sin1"
```

This maps the 8K data points into a smaller 3K waveform. View EXP2 with the command:

```
dev2:ibwrt "func:user exp2"
```

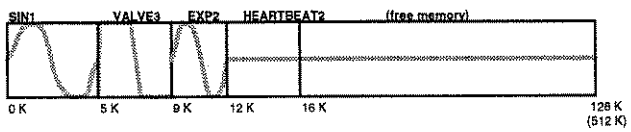
Verify that the EXP2 waveform now appears in Trace Memory per the following drawing:



Waveform Resizing is accomplished by using the TRACe POINts (not to be confused with the Trace Data Point) command to specify a different number of points in a given Trace. View SIN1 with the `FUNC:USER SIN1` command. With the oscilloscope horizontal trace set to 20 µs/DIV, the waveform will be approximately eight divisions long. Resize SIN1 from 8192 points to 5120 points, as follows:

```
dev2:ibwrt "trac:poin sin1,5120"
```

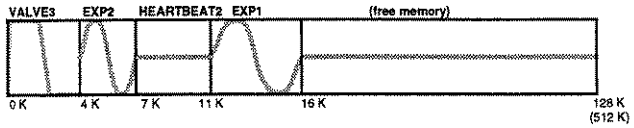
Verify that SIN1 is now approximately five divisions long. The Trace Memory should now be as follows:



SIN1 can be renamed EXP1 by first Copying it to the new Trace EXP1 and by Deleting SIN1. This is accomplished as follows:

```
dev2:ibwrt "trac:def exp1,sin1;del
sin1;:func:user exp1"
```

The oscilloscope display will be unchanged after the above steps because the new "EXP1" is identical to the old "SIN1". However, either a CATalog or DIRectory query would verify that the Trace Memory is now organized as follows:



Note that the Model 1385 manages the start and stop addresses of the Traces by "filling in" any free address space created by deleting or downsizing Traces. Traces will always occupy contiguous address space starting at the beginning of the Trace Memory.

3.4.5.4 Trace Limits

Recall that each Trace in Trace Memory can be thought of as a block of points with a relative start address of zero and a relative stop address of (size - 1). (The reference data in this manual often refers to "addresses" as "trace point indices"). The Limits command allows the user to select a *segment* within a Trace using this relative addressing. Subsequent Trace Subsystem operations on this Trace then apply to the selected segment and not the entire Trace.

NOTE

The Trace Delete command is an exception, it deletes the entire Trace, not just the selected segment. The start and stop Limits must be included in the range of zero to (size - 1), and the stop Limit must be at least five points greater than the start Limit. When a new Trace is Defined, the start and stop Limits default to zero and (size - 1).

As an example, Trace Limits can be used to create a HEARTBEAT2 waveform a segment at a time. This way more complex Traces can be created with the Model 1385's simple waveform editing capability. Select a segment within the HEARTBEAT2 waveform and view it as follows:

```
dev2:ibwrt "trac:lim heartbeat2,
1600,2400"
```

```
dev2:ibwrt "func:user heartbeat2"
```

Simulate the Q-R-S Wave portion of a PQRST heartbeat signal using the Positive Haversine:

```
dev2:ibwrt "trac:data heartbeat2,
phs"
```

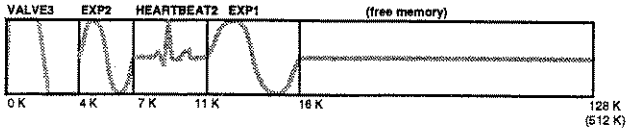
View the entire waveform by resetting the Limits:

```
dev2:ibwrt "trac:lim heartbeat2,
0,3999"
```

Note that the Haversine was written into only the selected segment, and the remainder of the Trace is still at the default half-scale value. Complete the heartbeat waveform with line segments, as follows:

```
dev2:ibwrt "trac:line heartbeat2,
800,2048,1000,2248;line heartbeat2,
1000,2248,1400,1600;line heart-
beat2, 1400,1600,1600,2048;line
heartbeat2,
2800,2048,3000,2248;line
heartbeat2, 3000,2248,3200,2048"
```

The resulting waveform should be a reasonable fascimile of a train of heartbeat pulses when viewed at a slower oscilloscope sweep setting. The Trace Memory should now be configured as follows:



The current Trace Limit settings of all waveforms in Trace Memory can be obtained in one single query using the `TRAC:DIR?` query.

3.4.5.5 Trace Queries

Most commands have a query form which can be sent by adding a question mark (?) directly after the Program Header. This will cause a response indicating the current setting of the parameter. The Trace Subsystem has three key words which have only a query form. These are intended to provide detailed information on the Traces in Trace Memory. Using the figure in the previous paragraph, each of these queries can be demonstrated.

Definite Length Arbitrary Block Data Format

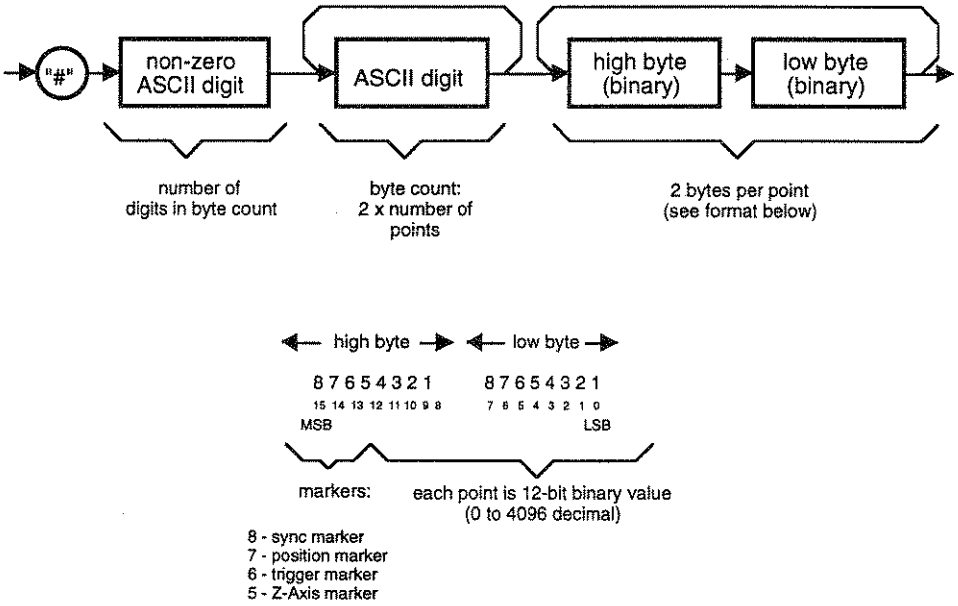


Figure 3-7. Definite Length Arbitrary Block Data Format

The TRACe:CATalog? query returns the names of each user-defined Trace in sequence. The response for the above figure would be:

```
dev2:ibwrt "trac:cat?"
dev2:ibrd 50
"valve3,exp2,heartbeat2, exp1"
```

The TRACe:DIRectory? query returns the names, sizes, and limit settings of each Trace in sequence. The response for the above figure would be:

```
"valve3,4000,0,3999,exp2,3000,0,2999,
heartbeat2,4000,0,3999,exp1,5120,0,5119"
```

The TRACe:FREE? query returns the number of points of Trace Memory in use and the amount free. For a Trace Memory size of 32K, the response for the above figure would be:

```
"16648,16120"
```

3.4.6 Waveform Download Operations

These operations are used to get large amounts of waveform data into or back from Trace Memory, using the host computer. This is the only practical way to work with complex waveforms. Data is entered into the Model 1385's Trace Memory as a "block" of data using the TRACe[:DATA] <trace_name>,<block> command. A block of data can be read back to the computer using the TRACe[:DATA]? <trace_name> query. In both cases, the *IEEE-488.2 Definite Length Arbitrary Block Data format* is used (see the next paragraph). The size of the "block" is determined by the current Trace Limits settings for the Trace using the <trace_name> specified. The Limits commands can be used to divide a large Trace into manageable blocks of data.

3.4.6.1 Definite Length Arbitrary Block Transfer

This format for block data transfer makes it possible to rapidly move the large amount of data required for arbitrary waveforms. Refer to Figure 3-7 for the following discussion.

To send a block of waveform data, send an ASCII "#" ($23), then an ASCII encoded digit whose value signifies the number of digits in the byte count, then ASCII encoded digit(s) representing the byte count, then the two byte binary data words (MSB first).

The byte count is twice the number of points to be downloaded to the trace. The byte count must exactly correspond to the number of bytes in the block of data.

Each data word is composed of two bytes representing a 12-bit unsigned integer between 0 and 4095. The remaining four most significant bits are normally set to zeros, but they may also be set to provide the various waveform markers at specific "relative addresses" within the Trace. As shown in Figure 3-7, the four markers have the following definitions:

Z-AXIS Not used in the Model 1385, the fourth-most significant bit (bit 12 of a 16-bit word) line is reserved for "Z-Axis" or horizontal intensity modulation of an oscilloscope display. This bit could be set at a particular point within a Trace by adding the value "4096" to the 12-bit data value at that point.

TRIGGER The third-most significant bit (bit 13 of a 16-bit word) line is reserved for an internal triggering signal used for intermodule triggering (see paragraph 3.4.13). This bit could be set at a particular point within a Trace by adding the value "8192" to the 12-bit data value at that point.

POSITION The second-most significant bit (bit 14 of a 16-bit word) line is reserved for an internal signal used to drive the MARKER POSITION output. This bit could be set at a particular point within a Trace by adding the value "16384" to the 12-bit data value at that point.

SYNC The most significant bit (bit 15 of a 16-bit word) line is reserved for an internal signal used to drive the SYNC/SWEEP output. This bit could be set at a particular point within a Trace by adding the value "32768" to the 12-bit data value at that point.

Note

Although the programmer/user may set markers within a Trace at the time that data is created for the block transfer, this action is not always necessary. The markers can be created by the Model 1385 with the appropriate SCPI commands at the time of their application.

Note

For example, a Trace with data supplied by a block transfer using 12-bit data values and no markers will not have a BBITs synchronization signal when first "played back". However, the operator can cause the 1385 to create these sync signals by simply programming the Sync Source to ZCRoss and then back to BBITs.

Prior to downloading a waveform using the TRACe [:DATA] <trace_name>,<block> command, the Trace must be Defined. Send the TRACe:DEFine <trace_name>,<size> command to define a Trace of "size" points under the trace name. This also presets the Trace Limits for this trace at full size.

You may select a segment of this trace for download (or subsequently, for upload) using the TRACe:LIMits <trace_name>,<start_ index>,<stop_ index> command.

Note

When the block size exceeds the capability of your download/upload application, you may use the TRACe:LIMits feature to break the block up into manageable segments.

The "binary" transfer occurs at a relatively high speed because the binary data is not parsed through the 1385's Command Processor. Instead, the data is routed directly to the Trace Memory RAM without processing or limit checking, much like a direct memory access (DMA).

If the waveform limits (size), the byte count or the number of bytes in the <block> are not all in numeric agreement, the high speed transfer will be aborted. Any data received after this will be interpreted by the Command Processor as ASCII characters and will cause the Model 1385 to generate many errors.

Several examples of programs written in Quick BASIC and C are given in Appendix E of this manual. These programs create data for a Trace and download the data using the block transfer format.

3.4.6.2 Shared Memory Transfer

The Model 1385's VXIbus Interface card contains 64 kB of "A24" Shared Memory. Shared Memory can be used by VXIbus modules to transfer large amounts of data quickly and efficiently without using Word Serial Protocol. See the VXIbus Specification for details on the VXIbus Shared Memory.

Although the Model 1385 does not support the Shared Memory Protocol, its Shared Memory can be used by the Commander and other instruments. In this case, the issue of interest to the Model 1385 would be creating a mechanism to download waveform data from Shared Memory to Trace Memory.

The mechanism for transfer from Shared Memory to Trace Memory is the pseudo-standard function SMEMory (see [SOUR:]FUNC<shape_name>). The SMEM function serves to operate as a possible source of data to be used with the TRAC[:DATA] <trace_name>,<trace_name> command.

When this form of the Trace Data command is used, the first trace name is the name of the Trace which is to receive the data. The second trace name is the Trace which is to be the source of the data. Data is copied from the source Trace to the destination Trace. The data is resized if necessary to the number of points set by the Trace Limits settings of the destination Trace.

The source Trace can be a Trace in Trace Memory or a Standard Function. When a Standard Function is used, the data is calculated according the function's shape and the destination Trace's size. The exception is the SMEMory Standard Function. When the source trace is the SMEM function, data is copied from Shared Memory to the destination Trace.

It takes two bytes from Shared Memory to make up one 16-bit waveform point (or word). A single data point is defined in Figure 3-7 and paragraph 3.4.6.1. The Shared Memory transfer does not use the ASCII header of the Definite Length Arbitrary Block Data format, but the binary data is the same (use Motorola byte order). Refer to Figure 3-7 and paragraph 3.4.6.1 for detailed information on the binary data word.

When the destination Trace is defined, it is given both a name and a size. At this time the Trace Limits settings include the full Trace. If the Trace Limits are reset to select a segment within the Trace then the Trace has a new size equal to:

(stop limit – start limit + 1)

The current Trace size is the number of points in the selected Trace or Trace segment, and it determines how many bytes of data are copied from Shared Memory. Since it takes two bytes to make a word, Shared Memory is copied starting at its A24 base address and sequentially up to the base address plus *twice* the value (size – 1). The Trace size can range from 5 to 128K points (512K points with the Option).

A 64K byte Shared Memory can contain a maximum of 32K data points. If the Trace has a size greater than 32K points, the Trace Limits can be used to break the waveform up into 32K blocks, and then use multiple downloads to build the waveform up in segments.

Appendix E of this manual provides an example "C" program written for the RadiSys EPC-2 Embedded Controller which writes a ramp pattern into the 1385's Shared Memory and then transfers it to the 1385's Trace Memory under the name of "test".

3.4.7 Non-continuous Modes

The Mode of waveform generation is *CONTinuous* as long as the default setting `INITiate:CONTinuous ON` is not changed. Continuous Mode causes the Model 1385 to output the selected Trace continuously. Changing this setting to OFF sets the Model 1385 to a non-continuous modes of operation. There are two non-continuous modes, Triggered and Gated. The default setting of the command `TRIGger:GATE[:STATe] OFF` determines that the unit will be in Triggered Mode until the command is sent to turn Gated Mode on.

3.4.7.1 Triggered Operation

In the triggered mode, the Model 1385's output remains quiescent (at a dc level) until triggered by the selected trigger source. All Model 1385 functions may be triggered. When triggered, the Model 1385 produces one or more complete cycles of the

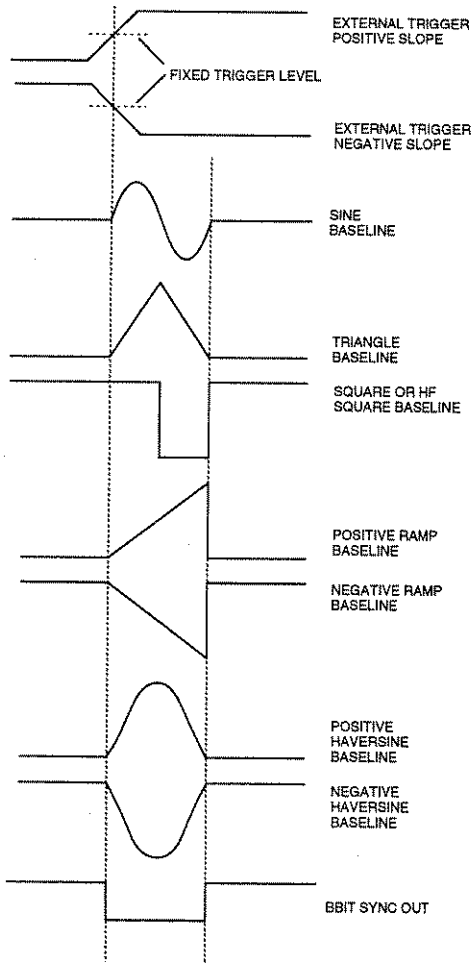


Figure 3-8. Triggered Waveform Characteristics, Count = 1

selected output function, then returns to the quiescent state at the level of the first point in the Trace. See figure 3-8 for the triggered waveform characteristics of the various functions. The following Examples describe how the Model receives a valid *triggering event* by using the INTernal trigger timer, the external TRIG IN, the trigger command, one of the eight TTLTrg<n> VXIbus trigger line, or the CHAin or ECHain signals on the VXIbus Local Bus (paragraph 3.4.13.1).

To set the Model 1385 for the triggered mode, follow the instructions in the following example:

Example 3:

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments. The following discussion provides a complete example for powering up the Model 1385 in its default settings and then modifying those settings to demonstrate a triggered output. In the following programming steps, the information in **bold** is the WIBIC prompt. The remainder of the string is to be typed in by the programmer. This example will continue through all of the paragraphs under "Non-Continuous Modes".

The first line identifies the GPIB card. The second line identifies the Model 1385 UUT. The third line sets the UUT's output on and turns on the SYNC OUT. The fourth line changes the sine wave's 1kHz default frequency to 10kHz and changes the synchronization signal from ZCRoss to BBITs. The fifth line changes the generator Mode from Continuous to Triggered.

```
:ibfind gpib0
```

```
gpib0:ibfind dev2
```

```
dev2:ibwrt "outp on;:mark:sync on"
```

```
dev2:ibwrt "freq 1e4;mark:sync:
sour bbit"
```

```
dev2:ibwrt "init:cont off"
```

A 0.1 ms SINusoid waveform cycle should appear once every 1 ms, with a dc baseline between cycles.

Internal Trigger

The trigger slope parameter (`TRIG:POL <POS|NEG>`) has no effect on an internal trigger source, but the remaining Trigger Subsystem commands will effect the internally triggered output.

To trigger the generator internally, set up the internal trigger frequency to a value lower than the function generator (CW) frequency. Set up the generator as in the Continuous example:

```
dev2:ibwrt "res"
```

```
dev2:ibwrt "outp on;mark:sync on"
```

```
dev2:ibwrt "freq 1e4;volt 0.75;
volt:OFFS -1.5;:init:cont off"
```

The result will be a 0.1 ms,1.5 Vpp (into 50Ω) sine wave riding on a dc baseline level of -1.5V, triggered at a 1 kHz rate. The SINusoid function, the INTernal Trigger Source, and the 1 ms Trigger Timer setting are all default settings. Change each of these settings as follows:

```
dev2:ibwrt "trig:sour ext"
```

The triggered waveform disappears, leaving the -1.5 Vdc baseline.

Then set the following:

```
dev2:ibwrt "trig:sour int;
tim 2e-3;: func tri"
```

The triggered function is now the TRIangle, and the trigger time is increased to 2 ms. Note that the trigger baseline is at the negative peak of this waveform.

External Trigger Input

In external trigger, the Trigger Timer setting will have no effect on the triggered waveform. The trigger slope determines whether the instrument triggers on the positive- or negative-going portion of the input signal.

First perform the steps in the previous paragraph. Next, select the external trigger input by sending the command:

```
dev2:ibwrt "trig:sour ext"
```

The oscilloscope should show a dc baseline at the previously noted level. Prepare to connect an external signal to the TRIG IN connector to trigger the 1385. To trigger on the positive-going trigger slope (rising edge), send the command:

```
dev2:ibwrt "trig:slop pos"
```

To trigger on the negative-going trigger slope (falling edge), send the command:

```
dev2:ibwrt "trig:slop neg"
```

The maximum specified trigger rate is 5 MHz.

The minimum specified pulse width is 20 ns. The trigger threshold (2 kΩ input) is a fixed value of approximately +1.2 V for TTL signal level compatibility. Select a signal source which can provide a TTL square wave with a frequency up to 5MHz. Set the external generator for approximately 1kHz. Connect the external TTL signal to the Model 1385's TRIG IN BNC. Verify that the triggered TRIangle function returns.

VXIbus TTL Trigger Lines Input

The Model 1385 may also be triggered from the VXIbus TTL Trigger lines. The triggering signal must be placed on the selected TTLTrigger line from another source within the VXIbus chassis.

The following commands provide a complete example for one module triggering another module using a TTL Trigger line.

The first line resets the UUT. The second line sets the UUT's output on and turns on the SYNC OUT. The third line causes the 10kHz sine wave to be internally triggered at a 1kHz rate. The fourth line selects TTLTrg<0> as the trigger source, and the output goes quiescent. The fifth line identifies the external source in slot 1. The sixth line causes the external source to drive TTLTrg<0> with a signal at its 1kHz default frequency.

```
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
dev2:ibwrt "freq 1e4;mark:sync:
sour bbit;:init:cont off"
dev2:ibwrt "trig:sour ttlt0"
dev2:ibfind dev1
dev1:ibwrt "outp:ttlt0 on"
dev1:ibfind dev2
```

Trigger COUNt Command

The Trigger Count command allows the operator to select a number of cycles that are generated following each trigger event. When in Triggered Mode (both `INIT:CONT` and `TRIG:GATE[:STAT]` are `OFF`), and the COUNt is at its default value of "1", single cycle waveform triggering as shown in Figure 3-8 occurs. However, setting the COUNt to a higher value, such as "5" shown in Figure 3-9, the Model 1385 operates as a burst generator.

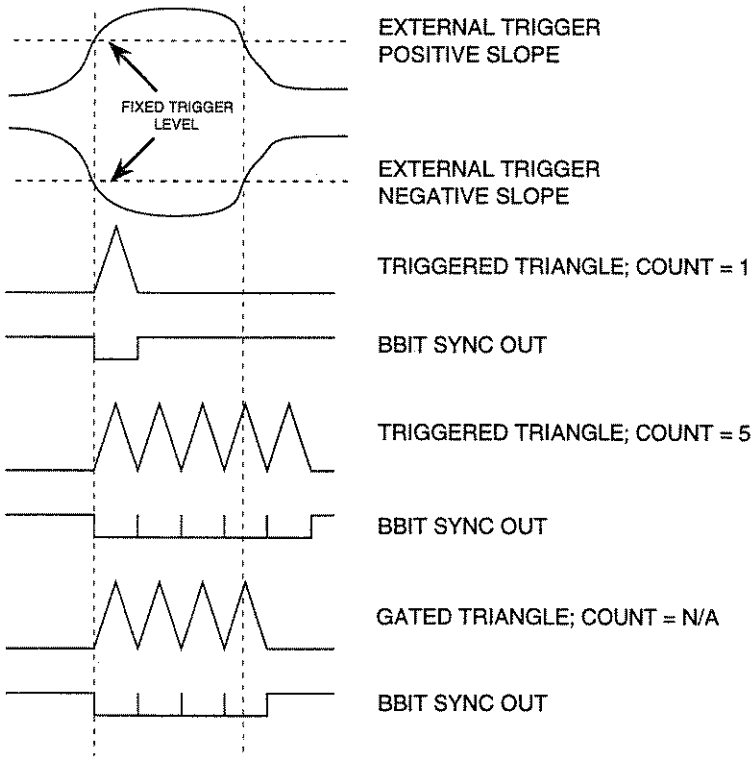


Figure 3-9. Gate/Burst Waveform Characteristics

The trigger COUNt is programmable up to 1,048,575 for Traces and up to 65,536 for segments of a Sequence (see paragraph 3.4.8). To view triggered operation with COUNt set to >1, continue with the sequence of commands from Example 3, as follows:

```
dev2:ibwrt "trig:sour ext;:func tri"
dev2:ibwrt "mark:sync:sour bbit"
```

Connect the external signal source (1 kHz TTL square) back to the Model 1385's TRIG IN. Verify that the triangle and BBITs sync waveforms correspond with Figure 3-9 for "COUNt = 1". Send the following:

```
dev2:ibwrt "trig:coun 5"
```

Verify that the triangle and BBITs sync waveforms correspond with Figure 3-9 for "COUNt = 5".

IEEE-488 and VXIbus Trigger Commands

To trigger the generator using the IEEE 488 bus or the VXIbus, set up the generator to follow on with Example 3 from the previous paragraph. Trigger the generator by sending either the 488.2 `*TRG` or 488.1 `GET` command over the 488 bus, or the word serial `TRIGger` over the VXIbus. The `*TRG` command is mandated to be recognized by the Model 1385.

The `GET` command causes the Commander to send the VXIbus word Serial `TRIGger` command to addressed devices which support `TRIGger` and do not have their DIR bit cleared to 0 (see VXIbus System Specification). Trigger slope does not apply when using these as the trigger source, and the bus commands only have effect when the Model 1385 is in a triggered mode of operation.

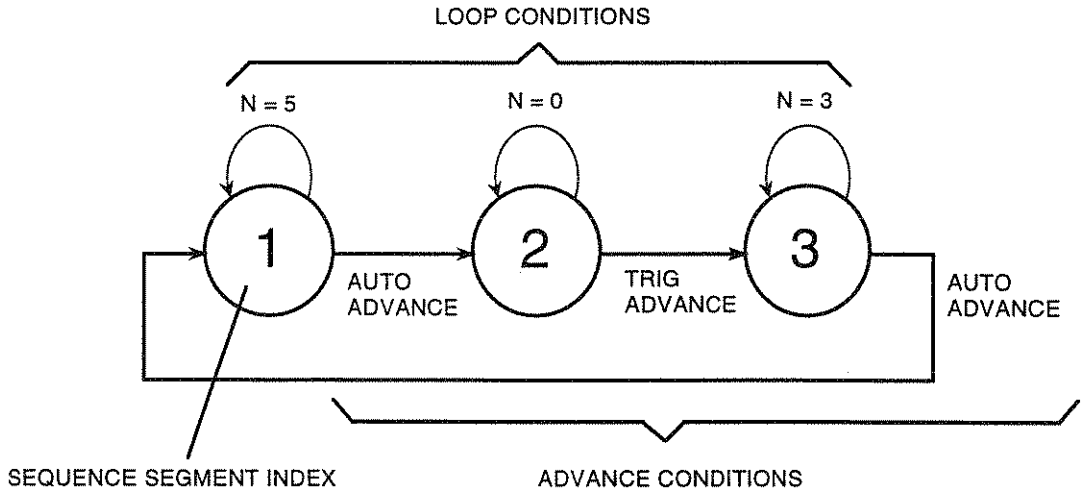
Proceed with Example 3 by first removing the external signal from the Model 1385's TRIG IN. Increase the trigger count so that a single burst can be seen and then send the following commands to trigger the generator:

```
dev2:ibwrt "trig:coun 1e3"
dev2:ibwrt "*TRG"
dev2:ibwrt "trig"
```

Set the 1385 back to external triggering by sending the command:

```
dev2:ibwrt "trig:coun def"
dev2:ibwrt "trig:sour ext"
```

Reconnect the external signal at the TRIG IN connector and verify triggered operation with COUNt = 1.



NOTE:
If N = 0, then the advance is triggered
If N ≠ 0, then the advance is automatic

Figure 3-10. CONTinuous Sequence State Diagram

3.4.7.2 Gated Operation

Gated operation is identical to the triggered operation, except the output from the Model 1385 starts from the quiescent state, produces continuous waveforms for the duration of the trigger signal, then returns to the quiescent state. There are 2 modes for gated operation, SYNChronous and ASYNchronous. SYNChronous will complete the last cycle of the waveform when the trigger is deasserted, ASYNchronous will not. All waveforms may be gated.

To view gated operation, set up the Model 1385 per Example 3 and the previous paragraph. Switch from triggered to gated operation with the following command:

```
dev2:ibwrt "trig:gate on"
```

Vary the frequency of the external TTL square wave and verify that the gated waveform is "on" half the time (plus the completion of the last cycle) and "off" half the time.

3.4.8 Sequence Operation

Refer to the SCPI command reference information given in paragraph 3.3.3.5 under "[SOURce:]SEQuence. . ." for detailed command information. A sequence is a means of adding more capability to the Model 1385 to accomplish even more complex tasks. A Sequence links together from two to 4096 waveforms, as shown in Figure 3-10. A State Diagram best displays the logic involved in setting up a Sequence using the 1385. Traces stored in Trace Memory are linked together as waveform *segments* in the Sequence.

Each segment can be repeated a number of times depending upon its Loop Count setting. When the Loop Condition is met by completing the Loop Count, the Sequence advances to the next segment. Sequences can be run in both the CONTinuous and the TRIGgered Mode. When in CONTinuous, the Sequence is always restarted when the final segment has completed playback.

When in TRIGgered, the Sequence is restarted under the same conditions that single waveforms are triggered (see paragraph 3.4.7). Paragraphs 3.4.8.1 and 3.4.8.2 give a tutorial on using Sequences in CONTinuous and TRIGgered Modes.

3.4.8.1 CONTinuous Sequencing

Refer to Figure 3-10 and the following discussions for CONTinuous waveform Sequencing.

Create Waveforms for a Sequence

In this tutorial a sine wave, a triangle wave, and a negative-going ramp will be created and assigned as Segment 1, Segment 2, and Segment 3 in a Sequence corresponding to Figure 3-10. Then the segment Loop and Advance Conditions will be set up according to the figure and it will be "played back" in CONTinuous mode. The SCPI commands to accomplish this are given in "Example 4", which follows this paragraph and continues on as a tutorial throughout all discussions on Sequencing.

Example 4

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the Model 1385 in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

Create the 3 waveform segments as follows:

```
dev2:ibwrt "trac:def seg1,1000;def
seg2,4000;def seg3,2000"
dev2:ibwrt "trac:data seg1,sin;data
seg2,tri;data seg3,nram"
```

The waveforms can be verified by playing each one back:

```
dev2:ibwrt "func user"
dev2:ibwrt "func:user seg1"
dev2:ibwrt "func:user seg2"
dev2:ibwrt "func:user seg3"
```

Create the Sequence

First define the Sequence length (the number of segments in the Sequence, from 2 to 4096). Use a length of 3 per Figure 3-10 for this tutorial:

```
dev2:ibwrt "seq:leng 3"
```

This sets the segment index to a range of "0" to "2". Then assign an index number to the 3 segments:

```
dev2:ibwrt "seq:func seg1,0;func
seg2,1;func seg3,2"
```

Set Up the Sequence Loop and Advance Conditions

The Loop Count (DWELl) is programmable from 1 to 1,048,576 repetitions on each segment, or as "continuous". Continuous looping is set by programming the Loop Count to "0". Using Figure 3-10 as a guide, program the following:

```
dev2:ibwrt "seq:dwel 5,0;dwel 0,1;
dwel 3,2"
```

This sets "SEG1", the segment assigned to index 1, to loop 5 times; "SEG2" to loop continuously; and "SEG3" to loop 3 times.

There are several Advance Conditions to concern yourself about with the Model 1385:

1. The loop count is set 0 and the ADVance is set to SYNChronous. This will cause the segment to repeat until the trigger is asserted. After the trigger, the waveform will complete that cycle and advance to the next segment.
2. The loop count is set to 0 and the ADVance is set to ASYNchronous. This will cause the segment to repeat until the trigger is asserted. After the trigger, the waveform will advance to the next segment.
3. The loop count is set to a non-zero number and the STARt is set to TRIGgered. The segment will repeat "n" number of times and then remain at a quiescent level until a trigger is received.
4. The loop count is set to a non-zero number and the STARt is set to AUTOmatic. The segment will repeat "n" number of times and then advance to the next segment.

The Sequence State Diagram shown in Figure 3-10 has three segments, each with Loop and Advance Conditions set up according to the three steps above. Segment 1 (the sine wave SEG1) is set up to automatically branch to segment 2 after completing 5 cycles. This corresponds to step 1 in the preceding discussion. Likewise, segment 2 (the triangle) is set up to loop continuously until triggered (step 2), and segment 3 (the negative ramp) is set up to run for three cycles and then branch back to segment 1 on a trigger. Currently, all three segments are set for AUTOmatic ADVance per the default. Program the three Advance Conditions using the following command:

```
dev2:ibwrt "seq:STAR trig,1;adv
trig,2"
```

At this time, leave the SEQuence TRIGger MODE as SYNChronous and the SEQuence TRIGger SENSe as EDGE.

Play Back the Sequence

The 1385's current output should be a continuous negative-going ramp with a 40 µs period and a BBITs synchronization signal. The default STARt and ADVance TRIGger SOURce (see Trigger Subsystem) is a 1 kHz signal from the internal trigger source generator. The STARt SOURce is used to trigger single Traces (as opposed to Sequences), to start a "continuous" SEQuence, and to periodically restart a "triggered" Sequence (see paragraph 3.4.8.2). The ADVance SOURce is used to branch from one segment to the next in a Sequence, after the Loop and Advance Conditions are met for that segment.

In this tutorial example, the STARt SOURce and the ADVance SOURce will be the EXTernal TRIG IN connector. Set an external generator for a 1 kHz TTL square wave per the previous examples under "Non-Continuous Modes".

Send the following commands:

```
dev2:ibwrt "trig:sour ext;sour:adv
ext"
```

This sets both SOURces to EXTernal. Connect the external signal to the TRIG IN connector at this time. Then select the Sequence as the function output:

```
dev2:ibwrt "func:mode seq"
```

This should get the sequence of waveforms going, with branching from segment to segment under control of the external triggering signal. Modify the equipment interconnect given in Figure 3-5 by disconnecting the 1385's SYNC/SWEEP OUT from Channel 2 of the oscilloscope. Use a BNC "tee" to connect the external generator's 1 kHz square to both the TRIG IN and to Channel 2 of the oscilloscope.

Set the oscilloscope's vertical mode to "alternate" to observe both channels, and internally trigger from Channel 2, negative trigger slope. Set the horizontal time base to 0.2 ms/DIV and adjust the trigger hold-off as necessary to get a stable display of the sequence centered in the display. Note the following:

1. After a low-to-high transition of the external square, the sequence starts with five sine waves (SEG1).
2. Immediately following the sine waves, the triangle (SEG2) starts and runs continuously up to the next low-to-high transition of the triggering signal. The final triangle is completed before branching.
3. After the last triangle, the three negative ramps (SEG3) are completed. This ends the sequence at the last point in the ramp, a negative dc level. It holds the negative value until a subsequent low-to-high trigger transition restarts the sequence.

Modify Segment 2 as follows:

```
dev2:ibwrt "seq:dwel 6,1"
```

Re-adjust the trigger hold-off as necessary to stabilize the oscilloscope display. Now the triangle should stop after six cycles, resulting in a low dc level between segment 2 and the start of segment 3.

Advance Triggering Commands

This tutorial has demonstrated Trigger Subsystem commands which set up the trigger sources to start and advance a Sequence. The Trigger Subsystem also can control the TRIGger POLarity, as follows:

```
dev2:ibwrt "trig:pol neg"
```

Note that the sequence now advances on high-to-low transitions of the external triggering signal. Set the POLarity back to POSitive:

```
dev2:ibwrt "trig:pol pos"
```

The [SOURce:]SEQuence sub-subsystem has two additional commands which effect the advance trigger. These have been in their default states, as mentioned earlier in this tutorial.

Note that at the end of segment 3 the trigger level is high, and that segment 1 doesn't restart until the trigger goes low, and then makes a low-to-high transition.

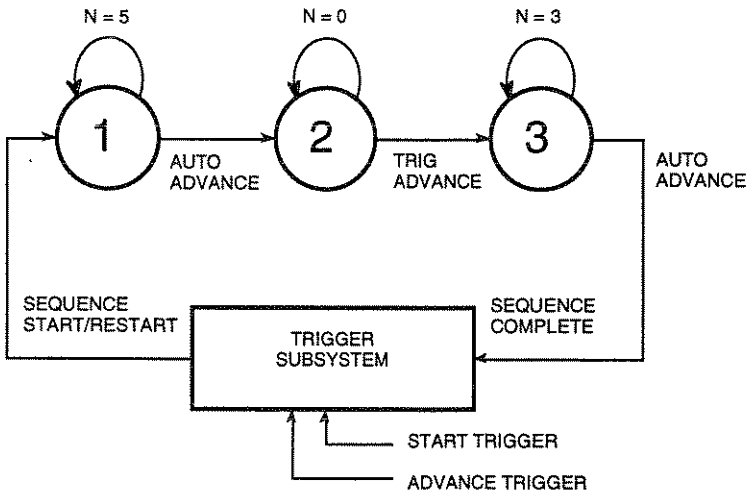


Figure 3-11. TRIGgered Sequence State Diagram

Return segment 2 to its earlier appearance, as follows:

```
dev2:ibwrt "seq:dwel 0,1"
```

Then send the command:

```
dev2:ibwrt "seq:trig:mode asyn"
```

Note that segment 2 now advances to segment 3 at the trigger transition, without first completing the last triangle waveform.

3.4.8.2 TRIGgered Mode Sequencing

Refer to Figure 3-11 and the following discussions for TRIGgered Mode waveform Sequencing.

In this part of the Example 4 tutorial, the advance trigger will be the internal trigger generator, and the start trigger will come from the external trigger source. For this part of the tutorial, it is best to use a 1375 or 1385 SYNC OUT (ZCRoss) as the external generator so that the two frequencies can be different, but still "in sync" when the programmed frequencies are harmonically related.
See "Example 1" for information on programming "dev1". Send the following:

```
dev2:ibwrt "trig:sour:adv int"
```

The oscilloscope display may become unstable. Switch to triggering from Channel 1 and adjust trigger level and hold-off as necessary for a stable display of the sequence. Set the external generator to a frequency of 200 Hz. Set the oscilloscope to 1 ms/DIV to see two cycles of the external trigger and 5 "cycles" of the sequence. Then send the following to enter triggered generator mode:

```
dev2:ibwrt "init:cont off;:seq:
STAR auto,2"
```

The sequence will now have its restart cycle triggered by low-to-high transitions of the 200 Hz signal.

This completes Example 4. Disconnect the test equipment.

3.4.9 AM/SCM Operation

This paragraph provides a tutorial, Example 5, which takes the operator through the SCPI programming steps to demonstrate Amplitude Modulation of the 1385.

Example 5

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the Model 1385 in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

Connect the Model 1385 Unit Under Test (UUT, slot 2) according to Figure 3-5 to display the sine wave output on the oscilloscope. This sine wave will be the *carrier* for the Amplitude Modulated signal. A second signal is required to drive the AM IN input connector on the Model 1385 to provide the AM modulation *envelope*. This signal is external to the UUT and may come from the VXIbus module in slot 1 (see example 1) or from a signal generator outside the VXIbus chassis. This tutorial will assume a Wavetek Model 1375 or 1385 is located in slot 1. First, the carrier signal should be set to a higher frequency than the modulation signal:

```
dev2:ibwrt "freq 1e5"
dev2:ibfind dev1
dev1:ibwrt "res"
dev1:ibwrt "outp on;:volt 0.2"
```

Modify the setup of Figure 3-5 by connecting the ARB OUT (MAIN OUT) of the slot 1 module to the AM IN of the 1385 UUT, using a BNC "tee" to connect the modulation signal to Channel 2 of the oscilloscope. Observe both channels, triggering on the 1 kHz signal on Channel 2. Enable the AM input at the UUT with the following:

```
dev1:ibfind dev2
dev2:ibwrt "am on"
```

The signal on Channel 1 should be the 100 kHz carrier from the UUT Amplitude Modulated approximately 100% by the 1 kHz signal at the AM IN. Verify Suppressed Carrier Modulation (SCM) by sending the following:

```
dev2:ibwrt "am:mode scm"
```

The signal on Channel 1 should change to SCM.

This completes Example 5. Disconnect the equipment.

3.4.10 Sync/Position Markers

The Sync Marker, when enabled, appears at the SYNC/SWEEP output connector of the Model 1385 in all modes except Frequency Sweep (see paragraph 3.4.11.1). The Sync Marker is used to synchronize to the waveform start/stop point of the signal at the MAIN OUT. It could also be used as an auxiliary TTL frequency output at the selected waveform frequency. The Sync Marker can be programmed as "ZCRoss" or "BBITs". The ZCRoss form produces a synchronization signal which would result from passing the signal at the MAIN OUT through a *zero-cross detector*. The sync is a TTL high whenever the MAIN OUT waveform data is above half-scale (800 hex), and a TTL low when below half-scale. This works well for most of the Standard Waveforms, which are symmetrical in time and amplitude. However, arbitrary waveforms generally do not have symmetry and may have multiple zero-crossings. Then the BBITs form should be used, which provides a narrow pulse at the start/stop point.

The Sync Marker has been demonstrated in previous examples. See Figure 3-5 and Example 1 for Continuous Mode waveform synchronization, and Figures 3-8 and 3-9 and Example 3 for Non-continuous Modes waveform synchronization.

The following tutorial, Example 6, takes the operator through the SCPI programming steps to demonstrate the use of the Position Marker of the 1385.

Example 6

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the Model 1385 in its default states by sending the following:

```
:ibfind gpib0

gpib0:ibfind dev2

dev2:ibwrt "res"

dev2:ibwrt "outp on;:mark:sync on"
```

Connect the Model 1385 Unit Under Test (UUT, slot 2) according to Figure 3-5 to display the sine wave output on the oscilloscope. Note the use of the ZCRoss Sync Marker on Channel 2 of the oscilloscope to provide waveform synchronization.

Remove the cable at the SYNC/SWEEP output of the UUT and connect it to the POSITION output. The oscilloscope will lose sync. Set the oscilloscope to trigger internally from Channel 1. Note that the POSITION output is a TTL low.

The Position Marker requires programming to appear at the POSITION output. The `[SOURce:]MARKer:POSition:POINt<trace_name>,<point_index>` command will be used to set position markers.

First, create a Trace:

```
dev2:ibwrt "trac:def sin1,sin"

dev2:ibwrt "func:user sin1"

dev2:ibwrt "func user"
```

This "clones" the sine function with the default size of 8192 points. Provide a synchronization pulse starting at the "90°" phase point of the sine wave using the Position Marker with the following commands:

```
dev2:ibwrt "mark:pos:poin sin1,2047; poin sin1,2048;poin sin1,2049"
```

This creates a TTL positive pulse 3 samples wide covering points 2047, 2048, and 2049 of the sine wave, which has a relative address range of 0 to 8191. It was created 3 samples wide for easier visibility on Channel 2 of the oscilloscope. By switching the oscilloscope trigger source from Channel 1 to Channel 2, the MAIN OUT can be viewed as a cosine (90° starting phase).

This completes Example 6. Disconnect the test equipment.

3.4.11 Frequency Sweep

The Model 1385 provides two mechanisms (other than direct programming) for rapid control of the generator's instantaneous frequency. These are covered in the next two paragraphs.

3.4.11.1 Sweep Generator

The Sweep Generator is controlled by a group of commands under the Source Subsystem which start with "`[SOURce:]SWEep`". When the command "`[SOURce:]FREQuency:MODE SWEep`" is sent, changing the Frequency Mode from its default value of "`CW`" to "`SWEep`", the Sweep Generator is enabled. The following tutorial, Example 7, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1385's Sweep Generator.

Example 7

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the Model 1385 in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on"
```

Connect the Model 1385 Unit Under Test (UUT, slot 2) according to Figure 3-5 to display the sine wave output on the oscilloscope. Then send the following:

```
dev2:ibwrt "freq:mode swe"
```

The oscilloscope display should show the sine wave being swept from a lower frequency to a higher frequency. Additionally, the signal on Channel 2 should now be a voltage level indicating instantaneous sweep position. All Sweep Generator settings are at their default values, as follows:

Start Frequency	1 kHz
Stop Frequency	100 kHz
Manual Frequency	1 kHz
Sweep Count	1
Sweep Direction	UP
Sweep Spacing	LINear
Sweep Time	1 s
Sweep Mode	CRESet

The Start and Stop Frequency settings indicate that the Sweep Generator is set to sweep between 1 kHz and 100 kHz. Since the DIRection is UP, the Sweep Generator resets to 1 kHz and LINearly sweeps UP to 100 kHz, and then resets to the Start value. The process is repeated once each second. Adjust the Start and Stop Frequency values with the following

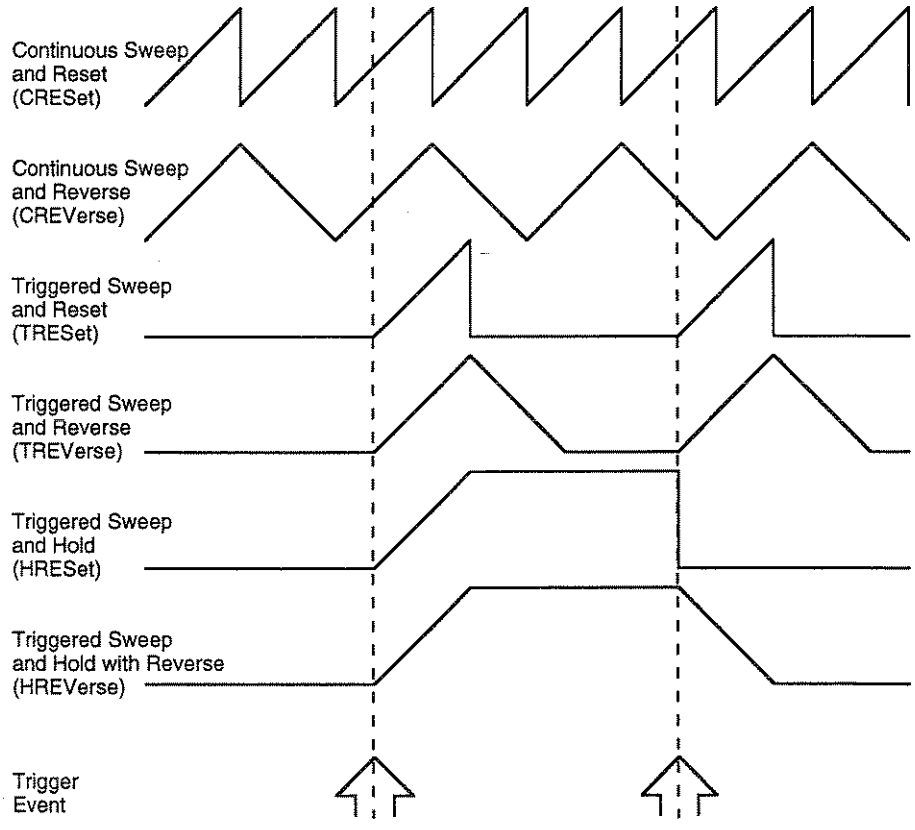


Figure 3-12. Sweep Mode Characteristics.

command and note the change in frequencies on the oscilloscope:

```
dev2:ibwrt "freq:star 5e2;stop 5e4"
```

Sweep Direction reverses the higher and lower frequencies as follows:

```
dev2:ibwrt "swe:dir down"
```

Set the DIRection back to normal and then change the "spacing" from LINear to LOGarithmic:

```
dev2:ibwrt "swe:dir up"
```

```
dev2:ibwrt "swe:spac log"
```

The SWEEP OUT signal on Channel 2 is not changed, and remains a linear indication of position within the sweep relative to sweep time. However, the swept sine wave on Channel 1 should be spending more of the sweep time at lower frequencies and less at higher frequencies. A LOGarithmic sweep will spend equal time per octave (and decade) of frequency coverage. Select the sweep time with the following:

```
dev2:ibwrt "swe:time 2"
```

The time from Start to Stop Frequency should now be 2 seconds.

The various Sweep Modes available in the Model 1385 are depicted in Figure 3-12. The figure shows how frequency changes as a function of time, using LINear SPACing and with the DIRection set to UP. The default Mode is CRESet (Continuous sweep and RESet). The sweep is Continuous because it proceeds continuously, without the need of a triggering signal. A single sweep starts at the Start Frequency (for "UP" DIRection), sweeps to the Stop frequency over the selected Sweep Time, and then immediately RESets back to the Start.

The difference between the CRESet and CREVerse Sweep Modes is evident after sending the following:

```
dev2:ibwrt "swe:mode crev"
```

Note that instead of immediately resetting to the Start, frequency sweeps back down from Stop to Start, again at the rate set by the Sweep Time.

The remaining sweep modes are triggered sweeps. Set the Trigger Timer to 5 seconds and the Sweep Mode to TRESet with the following:

```
dev2:ibwrt "trig:tim 5;:swe:mode
tres"
```

Compare the sweep action to the figure for TRESet Sweep Mode. Set the Sweep Mode to TREVerse with the following:

```
dev2:ibwrt "swe:mode trev"
```

Compare the sweep action to the figure for TREVerse Sweep Mode. Set the Sweep Mode to HRESet with the following:

```
dev2:ibwrt "swe:mode hres"
```

Compare the sweep action to the figure for HRESet Sweep Mode. Set the Sweep Mode to HREVerse with the following:

```
dev2:ibwrt "swe:mode hrev"
```

Compare the sweep action to the figure for HREVerse Sweep Mode.

The final Sweep Mode is MANual. In this mode the sweep action stops and the frequency value can be programmed directly to values between the Start and Stop Frequencies. The difference between this Sweep Mode and going back to CW Frequency Mode, is that the SYNC/SWEEP output continues to output a voltage proportional to position within the sweep. When this voltage is used to drive the horizontal axis of an X-Y display device, and the Sweep Generator is driving a frequency selective device such as a filter, the user can manually locate a point in the device's response and determine its exact frequency.

Continue on with Example 7 to the next paragraph demonstrating the Frequency List.

3.4.11.2 Frequency List

The Frequency List provides an additional tool for the programmer to achieve frequency agility. The List provides frequency "hopping" between preprogrammed fixed frequency settings, rather than a contiguous sweep between start/stop settings.

The main advantages are simplicity and speed of programming. Frequency List supports approximately 2000 setting changes per second, which improves the approximate 50 settings per second that can be obtained by sending frequency setting commands over the VXIbus.

The mechanism for hopping from the current setting in the List to the next setting is the trigger event. The trigger event can come from any of the available trigger sources defined in paragraph 3.4.7.1.

This is especially useful in an ATE system environment, where the command to advance from one setting to the next can come from external equipment signals at the TRIG IN input, or through the system controller as a *TRG or TRIGger command.

Frequency values are first entered into the Frequency List. Use the `[SOURce:]LIST:POINts <value>` command to set the active size of the list, where `<value>` ranges from 1 to 1024 points. Enter frequency values into the list with the `[SOURce:]LIST:FREQuency <value>,<index>` command, where `<value>` is the frequency value to be set at the list position determined by the `<index>` (which ranges from 0 to the active size minus 1). Then the `[SOURce:] FREQuency:MODE LIST` command is used to start frequency hopping.

Set the following commands to make a simple Frequency List:

```
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
dev2:ibwrt "trig:tim 1"
dev2:ibwrt "list:freq 1e4,0;freq 2e4,1;freq 3e4,2;freq 4e4,3;freq 5e4,4"
dev2:ibwrt "list:poin 5"
dev2:ibwrt "freq:mode list"
```

The sine wave should hop from one frequency on the list to the next once per second, the internal trigger timer interval. This completes Example 7.

3.4.11.3 Frequency SMPI

Even faster frequency changes are possible using [SOURce:]FREQuency:MODE SMPI (SMPI is an acronym for Shared Memory Phase Increment). In this mode of operation the output frequency is determined by the contents of the lower three words (48bits) of shared memory.

Upon receipt of this command the Model 1385 sets up the currently selected trace for Phase accumulated mode of operation and the Phase Increment is initialized for an output frequency of 1kHz. For this to work properly the size of the trace must be a binary (1,2,4,8,...) multiple of 4096 points.

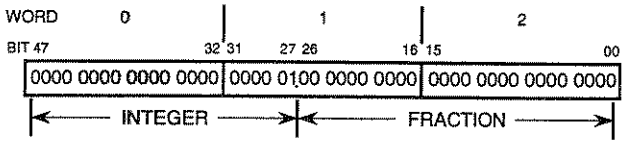
The output frequency is updated by placing the desired Phase Increment value in the lower three words of Shared Memory and sending a Word Serial Command with the value 0x0003. The receipt of this Word Serial Command causes the Model 1385 to Synchronously update the Phase Increment.

When this operation has been completed, the Model 1385 resets the Write Ready bit to 1. A new Phase Increment value should not be written to Shared Memory until the Write Ready bit has been set.

The value of the Phase Increment can be calculated using the following equation:

PI = (Output Frequency * Trace Size)/50e6

The Phase Increment must be formatted into a 48-bit value with the upper 22 bits to the left of the decimal point and the lower 26 bits to the right of the decimal point. The diagram below represents a Phase Increment of 1.0.



The 48-bit Phase Increment is placed in Shared Memory with the most-significant word (bits 47-32) of the Phase Increment located in the first word of Shared Memory. The next two words are placed in consecutive order into Shared Memory.

	WORD	BYTE	OFFSET
15	2	00	4
31	1	16	2
47	0	32	0

A24 Shared Memory

This mode of operation allows for sustained frequency updates of over 2000/s.

3.4.12 Clock Input/Output Operation

The 1385's waveform synthesis operates in two modes. "Phase Accumulation" or "CW" Mode operates from a fixed 50 MHz waveform clock, and is used with most of the Standard Functions. "Raster" Mode operates with a variable frequency clock, and is primarily used with arbitrary waveforms (Traces specified by the USER FUNCtion).

These selections are automatic, but may be overridden by the operator. In either case, the waveform Clock is the waveform *sample frequency*, and the corresponding *sample period* is the time spent at

each point selected for waveform playback. The Clock is normally generated internally and available for output at the CLK IN/OUT connector.

The connector may also be an input for an external signal which will control the waveform sample frequency.

The following tutorial, Example 9, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1385's Clock Sources.

Example 9

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the Model 1385 in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

Connect the Model 1385 Unit Under Test (UUT, slot 2) according to Figure 3-5 to display the sine wave output on Channel 1 of the oscilloscope. Trigger the oscilloscope internally from Channel 2. Move the BNC cable from the SYNC/SWEEP output to the CLK IN/OUT connector. Note that there is no TTL level signal on Channel 2. Then send the following:

```
dev2:ibwrt "outp:cloc:freq 1e6"
```

Note that the sine wave turns off and that Channel 2 now has narrow positive pulses occurring at 1 µs intervals. The UUT is now in a "special" mode of operation wherein the SCPI programming required to turn off the MAIN OUT, configure the CLK IN/OUT as an output, set up the Clock Source as RASTer or SYNThesizer, and program the required frequency to get a frequency of 1e6 at the CLK IN/OUT has all been done. As long as the operator remains in this special Clock Mode, the Clock Frequency is programmable with the above command.

Cancel Clock Mode by sending:

```
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

The next set of commands accomplish the same task without going into Clock Mode. This requires more commands, but it has the advantage of leaving the waveform output on.

Send the following:

```
dev2:ibwrt "trac:mode rast;:
outp:cloc:sour rast;:freq:rast 1e6"
dev2:ibwrt "cloc:conf outp"
```

The waveform on Channel 2 will be the narrow clock pulses at 1 µs intervals (1 MHz). The waveform on Channel 1 will be the 4096 point sine wave with a 1 MHz sample frequency, or approximately 244 Hz.

The remainder of the example will deal with using the CLK IN/OUT as an input. Reset the UUT with the following:

```
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

The Clock Output signal on Channel 2 will disappear. Move the cable from CLK IN/OUT to SYNC/SWEEP OUT and trigger on the ZCRoss sync on Channel 2. Channel 1 has the default 1 kHz, 4096 point sine wave. The RESet sets the CLK IN/OUT back to its default configuration, as an INPut. Send the following:

```
dev2:ibwrt "rosc:sour ext"
```

The waveforms on the oscilloscope stop. Connect a TTL level signal from an external source to the CLK IN/OUT connector (the module in slot 1 or an external generator may be used). The frequency of this signal can be in the range of dc to 50 MHz. For a 1 kHz output of the sine wave at the 1385's MAIN OUT, the "CW Mode" clock needs to be 50 MHz. A lower Clock Frequency will scale the sine wave frequency down proportionately.

The remainder of this example assumes that the module in slot 1 is another Model 1385. The following program steps will use a 50 MHz Clock Frequency from "dev1" to drive the VXIbus ECL Trigger Line 1, which will be input to "dev2" as its ROSCillator (reference oscillator) Clock Frequency. Send the following:

```
dev2:ibwrt "rosc:sour eclt1"
dev2:ibfind dev1
dev1:ibwrt "res;:outp:eclt1 on"
```

After the first command the UUT's MAIN OUT should stop, and after the last command the sine wave should return. The 1 kHz sine wave frequency indicates that the UUT is being clocked at 50 MHz.

This completes Example 9. Disconnect the test equipment.

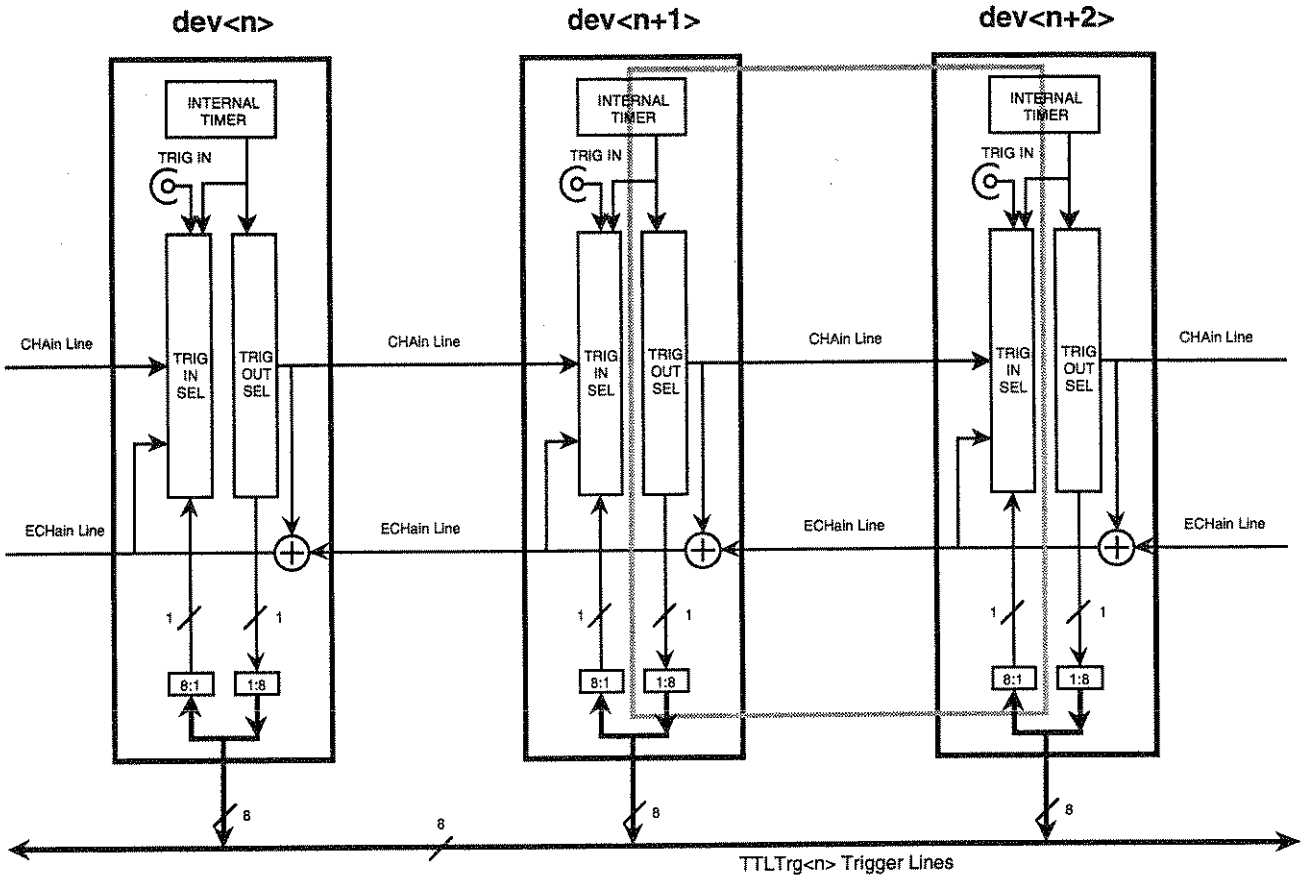


Figure 3-13. Intermodule Triggering Backplane Connections.

3.4.13 Intermodule Operations

3.4.13.1 Intermodule Triggering

Intermodule triggering provides much greater versatility than single module triggering as described in Example 3 (paragraph 3.4.7 and Figures 3-8 and 3-9).

Intermodule triggering utilizes interconnections on the VXIbus backplane. Figure 3-13 diagrams these interconnections. The figure illustrates three Model 1385 modules adjacent to one another (their device numbers are consecutive) in the VXIbus chassis. This is required to "daisy chain" the VXIbus Local Bus from module to module. For the purpose of this discussion on triggering, each module can be thought of as a Trigger Input Selector and a Trigger Output Selector (of course, the waveform generation circuits are positioned between the two).

The eight TTL Trigger lines run the full length of the backplane and do not require Local Bus support. The CHAin and ECHain Lines are part of the Local Bus between two Model 1385s. Each Trigger Input Selector can select between INTernal, EXTernal, CHAin, ECHain, one of eight TTL Trigger lines, and Word Serial trigger inputs to trigger the waveform generator circuits.

Each Trigger Output Selector has several input signals from the waveform generation circuits (not shown at this level, see Figure 3-14) and the INTernal trigger signal.

The Output Selector selects a signal as the Output Trigger Source. This signal is always connected to the CHAin Line and to the input of the next module to the right. It may also be selected to drive the ECHain Line or one of the eight TTL Trigger lines.

Intermodule triggering can be accomplished with the TTL Trigger lines or with the Local Bus CHAin/ ECHain lines. The example programming in this paragraph (Example 10) will demonstrate both methods. The CHAin/ECHain method is more versatile, but it requires that the modules be adjacent for Local Bus operation.

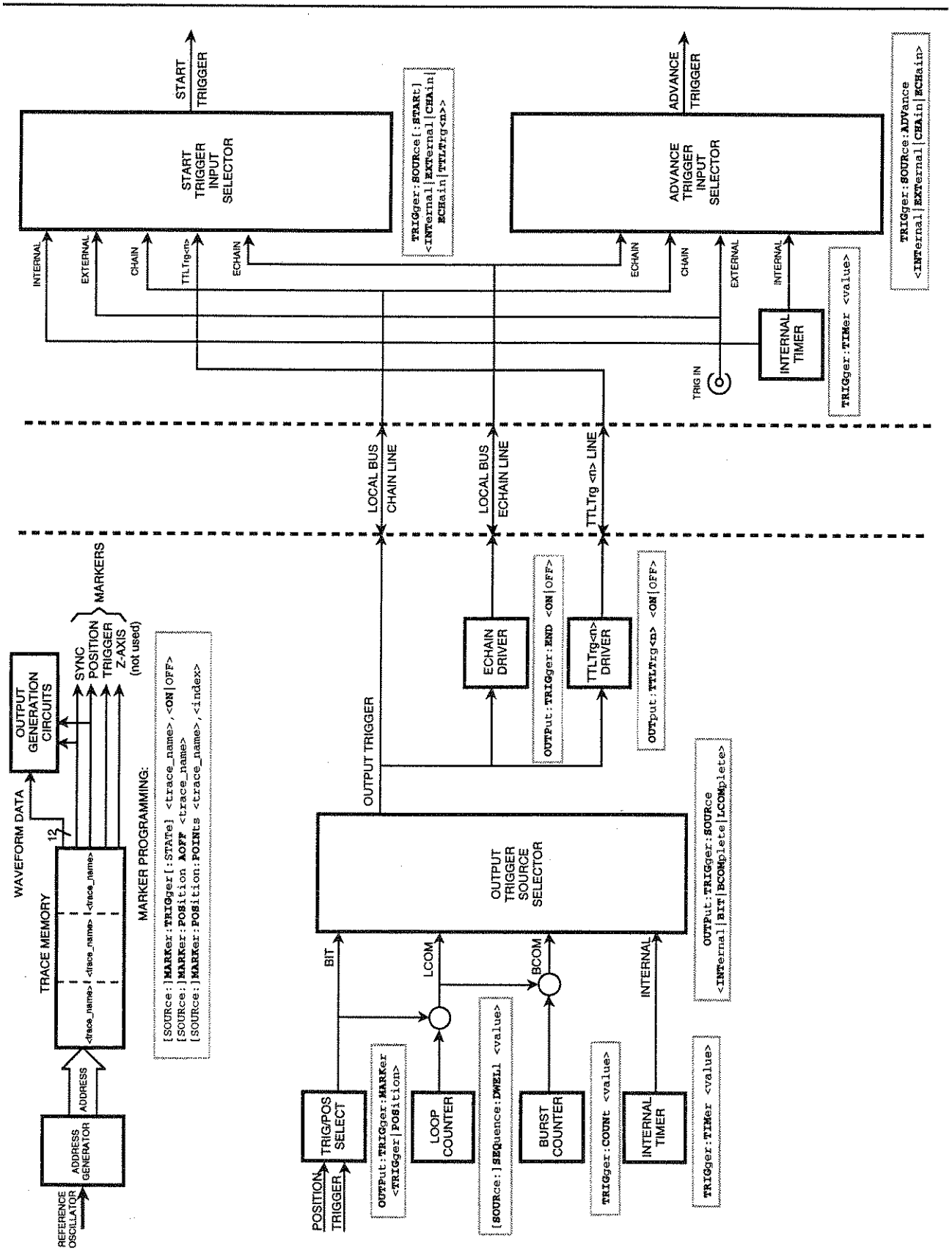


Figure 3-14. Intermodule Triggering Command Reference.

In general, the left-most module will be the timing master. The CHAin Line passes triggers left-to-right one module at a time. The right-most module in the chain will usually drive the ECHain Line back to the left-most module. This general outline is not the only possibility, the fact that the ECHain Line is "wire-ORed" to all trigger outputs suggests that more complex triggering schemes can be implemented.

The shaded box in Figure 3-13 indicates the area which is diagrammed in greater detail in Figure 3-14. Figure 3-14 not only provides a functional block diagram of triggering operations, but also provides SCPI commands which apply to each block. It is intended that the programmer/operator use this figure as a programming aid when dealing with this very complex subject.

Figure 3-14 diagrams the operation of Trigger Output selection and Trigger Source selection. The figure relates more closely to the SCPI commands rather than the physical hardware.

The selected Reference Oscillator (ROSCillator) clocks the Address Generator/Mode Control block, which in turn outputs Addresses to Trace Memory. These Addresses are updated at the rate of the waveform sample frequency. Selected Traces are accessed according to their <trace_names> included in the SCPI programming for waveforms and/or Sequences.

On a Trace-by-Trace basis, the programmer can choose to set or not set Markers within waveforms. These Markers can be used as the source for triggering signals, either directly or through gating with signals from the Loop or Burst Counters.

Trace Memory has 16 bits of data output. 12 bits are used for the waveform Digital to Analog Converter (DAC). The remaining 4 bits are Marker lines, which can be set true on selected points in a Trace.

The SYNC Marker can be set up to provide either the BBITs or ZCRoss synchronization signal to drive the SYNC/SWEEP output.

The POSITION Marker can have selected points set true in a given Trace. The POSITION Marker drives the POSITION output, and can be selected to *qualify* the OUTPUT TRIGGER.

The TRIGGER Marker can be turned on or off for a given Trace. The TRIGGER Marker, which is set to appear at the end of a Trace, can be chosen to *qualify* the OUTPUT TRIGGER.

The Z-AXIS Marker is not used in the Model 1385. Below the Trace Memory block in the figure, note the three SCPI commands which are used to define the POSITION and TRIGGER Markers.

The `OUTP:TRIG:MARK <POS|TRIG>` command selects which of the two Markers are used to drive the "BIT" signal. The BIT signal is one of the Trigger Sources applied to the Output Trigger Source Selector. The BIT signal is true for selected "bits" within selected Traces. The BIT signal contains the Trace-specific position information of the Marker that produced it.

When a Sequence is defined, the `SEQ:DWEL <value>` command is used to determine how many times to repeat (Loop on) a given segment before passing to the next. During playback this repeat count is controlled by the Loop Counter on a Trace-by-Trace basis.

The output of the Loop Counter is false except during the last cycle of the Trace using a repeat count. The BIT signal is *qualified* with (ANDed with) the Loop Count Complete signal to produce the LCOM (Loop Complete) signal. The LCOM signal is the same as the BIT signal, except that only the bits that occur in the final cycle of sequence loops remain.

The command `TRIG:COUN <value>` determines how many times a Trace or complete Sequence repeats each time it is triggered. The trigger count is controlled by the Burst Counter.

The Burst Count Complete is true during the last cycle of a Trace or Sequence using a trigger count. Both Loop and Burst Complete are used to qualify the BIT signal, producing the BCOM (Burst Complete) signal. This is true for the last loop of the last burst of the selected Trace(s).

The INTernal trigger rate generator, under control of the `TRIG:TIM <value>` command, is a final input to the Output Trigger Source Selector.

The command `OUTP:TRIG:SOUR` selects among the various trigger sources to produce the Output Trigger. The Output Trigger is sent to the Local Bus CHAin line and to the input trigger selectors in the next module to the right in the VXIbus chassis.

It can also be selected to drive the Local Bus ECHain line using the `OUTP: TRIG:END ON` command. The Output Trigger is sent to the TTLTrg<n> Driver, where it can be selected to drive one of the eight TTL Trigger lines using the `OUTP:TTLT<n> ON` command.

The selected Output Trigger passes through the backplane to the next Model 1385 in the trigger chain. There are two input trigger selectors on a Model 1385, one to generate the Start Trigger and one to generate the Advance Trigger.

In addition to the Output Trigger from the previous module, the TRIG IN connector and the Internal Timer are inputs to the trigger source selectors. Note that the TTLTrg line can be used to generate a Start Trigger, but not an Advance Trigger.

The following tutorial, Example 10, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1385's Intermodule Triggering operation.

The tutorial assumes a Model 1385 in slot 1 using secondary address 1, and another Model 1385 in slot 2 using secondary address 2. The 1385 in slot 1 will drive and receive the ROSCillator on ECLT1, and will drive the Local Bus CHAin and TTL Trigger lines.

The 1385 in slot 2 will receive the ROSCillator on ECLT1, will receive the trigger on the CHAin and TTLT1 lines, and will drive the Local Bus ECHain line with its trigger output. Both modules will be providing waveform outputs at their MAIN OUT connectors.

Example 10

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the UUT in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

Set up the other Model 1385 in its default states by sending the following:

```
dev2:ibfind dev1
dev1:ibwrt "res"
dev1:ibwrt "outp on;:mark:sync on"
```

Verify that both modules are generating a 1 kHz sine wave at the MAIN OUT and a ZCRoss sync at the SYNC/SWEEP OUT.

The first task recommended for intermodule triggering is to put all modules in the triggering "chain" on the same waveform clock. This assures the closest possible signal coupling for proper timing. One module (if a group of adjacent modules is used, select one near the middle of the group) is selected as the Reference Oscillator source. This module will drive its waveform clock to ECLT1. All modules, including the reference source, will receive their Reference Oscillator from ECLT1. This way, the waveform sample frequency will be controlled by the chosen reference source. Program the modules as follows:

```
dev1:ibwrt "outp:eclt1 on;:rosc
:sour eclt1"
dev1:ibfind dev2
dev2:ibwrt "rosc:sour eclt1"
```

Verify that both modules' outputs remain unchanged. Then set up the waveform which will be used on "dev2" as follows:

```
dev2:ibwrt "trac:def
sin100,100;data sin100,sin"
dev2:ibwrt "func:user sin100;:
func user"
```

Verify "dev2's" output is now a 500 kHz (2 µs) sine with BBITs sync. Then set up the waveforms which will be used on "dev1" as follows:

```
dev2:ibfind dev1
dev1:ibwrt "trac:def
ramp100,100;def tri100,100;data
ramp100,pram;data tri100,tri"
dev1:ibwrt "func:user ramp100;:
func user"
```

Verify "dev1's" output is now a 500 kHz (2 µs) positive ramp with BBITs sync. Then send:

```
dev1:ibwrt "func:user tri100"
```

Verify "dev1's" output is now a 500 kHz (2 µs) triangle with BBITs sync.

Reduce the sample frequency from 50 MHz to 10 MHz to get triggering signals within the VXIbus bandwidth specification for the TTL Trigger lines:

```
dev1:ibwrt "freq:rast 1e7"

dev1:ibfind dev2

dev2:ibwrt "freq:rast 1e7"
```

NOTE

In this example each module in the chain was programmed to a Raster Frequency below the VXIbus bandwidth limit for TTL Trigger lines.

As an alternative, the Raster Frequency could remain at 50 MHz for higher waveform frequencies, and the programmer would have to ensure that any triggering signals that use the backplane were set to be several samples wide.

Verify both modules output 100 kHz (10 µs) signals. Set up a trigger repeat count for the "dev2" waveform:

```
dev2:ibwrt "trig:coun 3"
```

Set up a Sequence on "dev1" as follows:

```
dev2:ibfind dev1

dev1:ibwrt "seq:func ramp100,0;
func tri100,1"

dev1:ibwrt "seq;leng 2"

dev1:ibwrt "seq;dwel 2,0;dwel 3,1"
```

The advance condition for both segments remains defaulted to AUTOmatic. Set the Sequence to "Loop" three times per start trigger:

```
dev1:ibwrt "trig:coun 3"
```

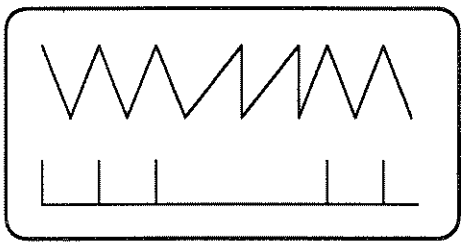
Start the sequence:

```
dev1:ibwrt "func:mode seq"
```

Use the oscilloscope's trigger hold-off to verify a sequence alternating between two ramps and three triangles. Set up the Position Marker as a synchronization signal for proper viewing of the Sequence, and set up the trigger source output to the backplane, as follows:

```
dev1:ibwrt "mark:pos:poin
tri100,50"
```

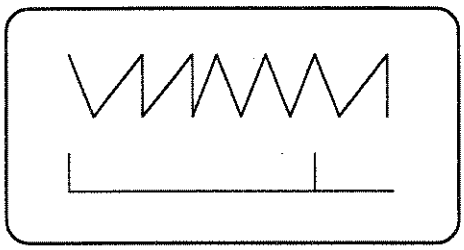
Switch the cable from the SYNC/SWEEP connector to the POSITION connector. Note that sync pulses occur at the triangle positive peaks. The oscilloscope display should be as follows:



Program the Trigger Marker Source as follows:

```
dev1:ibwrt "outp:trig:mark
pos;sour lcom"

dev1:ibwrt "outp:ttlt1 on"
```

This command selects the Position marker as the BIT trigger signal. The BIT signal (identical to the position marker on the oscilloscope) is then ANDed with the triangle's Loop Complete signal to produce the "LCOM" trigger signal, which occurs at the positive peak (last point in Trace) of the last triangle in each pass through the Sequence, as follows (the BIT, LCOM, and BCOM signals cannot be displayed on the oscilloscope):

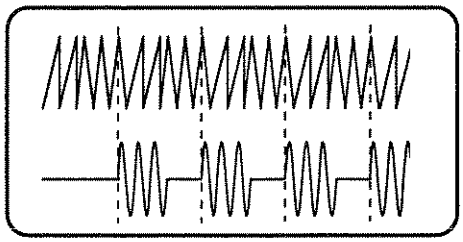


Finally, the LCOM signal is placed on the VXIbus TTL Trigger line 1 with the last command.

Now, set up "dev2" to be triggered from the LCOM signal:

```
dev1:ibfind dev2

dev2:ibwrt "trig:sour ttlt1;:
init:cont off"
```

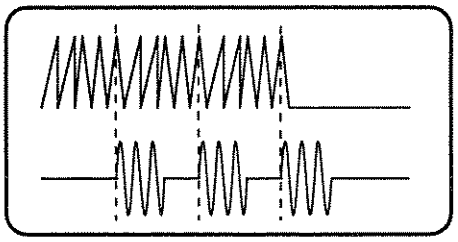
Move the cable from the POSITION output of "dev1" to the MAIN OUT of "dev2". Observe the triggered waveforms.



Next modify the Sequence so that it is triggered, as follows:

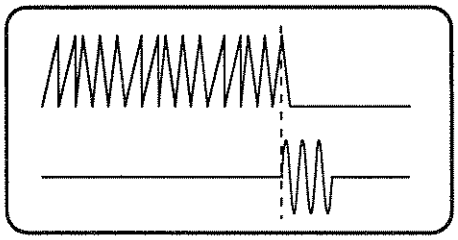
```
dev2:ibfind dev1
dev1:ibwrt "trig:sour:star int;:
trig:tim 3e-4;:init:cont off"
```

Trigger internally from Channel 1 and use trigger hold-off to stabilize the display. The waveforms should now appear as follows.



The top trace is the "two ramp/three triangle" Sequence repeated three times. The lower trace shows the three sine burst triggered by the LCOM qualified position marker at the peak of the last triangle in each "loop" of the triangle segment. To change from LCOM to BCOM send the command:

```
dev1:ibwrt "outp:trig:sour bcom"
```



The BCOM qualified position marker occurs only at the selected point in the final segment of the bursted Sequence. Verify that the same waveform is generated by using the Local Bus CHAin line instead of the TTLTrg<1> line by sending the following:

```
dev1:ibfind dev2
dev2:ibwrt "trig:sour cha"
```

The waveform on the oscilloscope should not change. Note that "dev1" did not need to be set up to drive the CHAin line; the selected output trigger source always drives the CHAin line.

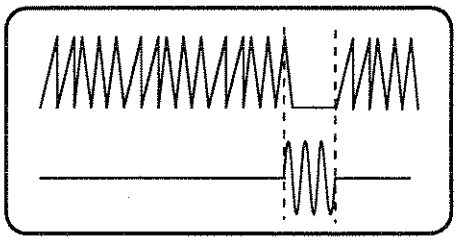
As a final part in this example, "dev2" will be set up to drive the ECHain line back to "dev1", using its trigger marker. Then "dev1" will run the Sequence and trigger "dev2". Following its trigger "dev2" will run its waveform and then trigger "dev1". This way activity can be sustained, alternating back and forth between the two modules. Send the following commands:

```
dev2:ibwrt "mark:trig sin100,on;:
outp:trig:mark trig;sour bcom;end
on"
dev2:ibfind dev1
dev1:ibwrt "trig:sour:star ech"
```

The waveforms may turn off. If so, re-initiate Sequence Start on "dev1" with the following"

```
dev1:ibwrt "trig"
```

The resulting output should appear as follows.



This completes Example 10. Disconnect the test equipment.

3.4.13.2 Intermodule Phase Lock

The following tutorial, Example 11, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1385's Phase Lock operation. The tutorial assumes a Model 1385 in slot 1 using secondary address 1, and a Model 1385 in slot 2 using secondary address 2.

Example 11

Refer to "Example 1" for general information regarding the use of National Instrument's WIBIC application for GPIB control of VXIbus instruments.

Set up the UUT in its default states by sending the following:

```
:ibfind gpib0
gpib0:ibfind dev2
dev2:ibwrt "res"
dev2:ibwrt "outp on;:mark:sync on"
```

Set up the other Model 1385 in its default states by sending the following:

```
dev2:ibfind dev1
dev1:ibwrt "res"
dev1:ibwrt "outp on;:mark:sync on"
```

Phase Lock operation requires that the modules be adjacent to one another in the VXIbus chassis because the Local Bus is used for phase synchronization signals. One module is designated the "Master" and the other (or others) is designated the "Slave". The modules must all be using the Master as the Reference Oscillator for their waveform sample clock. Additionally, the waveform Traces of all modules must have the same number of points and must be played back at the same clock frequency.

For this example "dev1" will be the Master and "dev2" will be the Slave. The default sine waves will be used.

Program the following to set the two modules to the same Reference Oscillator and waveform clock frequency:

```
dev1:ibwrt "outp:eclt1
on;:rosc:sour eclt1"
dev1:ibfind dev2
dev2:ibwrt "rosc:sour eclt1"
```

Modify the setup of Figure 3-5 by connecting the MAIN OUT from the Master to Channel 1 of the oscilloscope. Externally trigger the oscilloscope from the Master's SYNC/SWEEP output. Use positive slope for the external trigger and note that the sine wave on Channel 1 starts at its positive-going zero crossing. This corresponds to 0° phase of a sine wave.

Connect the Slave's MAIN OUT to Channel 2 and note that its starting phase is fixed at some arbitrary phase relationship to the Master. Initiate Phase Lock as follows:

```
dev2:ibwrt "phase:lock on"
dev2:ibfind dev1
dev1:ibwrt "phase:lock on"
```

The two sine waves should be in phase. Modify the Slave's phase relationship with the following:

```
dev1:ibfind dev2
dev2:ibwrt "phase 90"
```

The waveform on Channel 2 should now have its starting phase at its positive peak (cosine wave). Experiment with other phase angles.

This completes Example 11. Disconnect the test equipment.

4.1 FACTORY REPAIR

Wavetek maintains a factory repair department for those customers not possessing the necessary personnel or test equipment to maintain the instrument. If an instrument is returned to the factory for calibration or repair, a detailed description of the specific problem should be attached to minimize the turnaround time.

4.2 CALIBRATION

Calibration is the process of *Scheduled Maintenance* as described in this section of the Model 1385 manual. Through Calibration, the unit is certified to be operational and within the specifications listed in Section 1 of this manual. The Calibration is valid over a specified *Calibration Interval*. After the interval (typically 1 year), the operator returns the unit to the metrology laboratory for Calibration. Units returned at the scheduled interval, without a failure description, may be calibrated and returned to the operator using the procedures in this section of the manual.

Start the Calibration with the *Performance Verification Procedure* following immediately in this section. Performance Verification tests the unit vigorously to the specifications in Section 1, using external test equipment and signals at the unit's input and output connectors. There are Performance Verification Data sheets at the end of this section which are intended to be copied and used to record the data values from the verification test. Completed Performance Verification Data sheets with no out-of-tolerance readings is sufficient for certification of Calibration and return to the operator.

If there are out-of-tolerance readings, perform the Alignment Procedure later in this section. After successful completion of alignment, complete the Performance Verification Procedure.

If the Alignment Procedure cannot be run successfully, or if the Performance Verification Data still has out-of-tolerance readings after alignment, then the instrument should be returned to the factory for repair.

4.3 REQUIRED TEST EQUIPMENT

The test equipment required to perform the Performance Verification Procedure and the Alignment Procedure is listed in table 4-1.

Table 4-1. List of Test Equipment

Equipment	Qty	Specifications
VXI Reqmts	1	VXI chassis.
	1	VXI slot-0 controller.
	1	VXI "C" size extender board (Optional-see text).
Multimeter	1	Wavetek 1271 or equivalent.
Oscilloscope	1	Tektronix 2465 or equivalent.
Freq. Counter	1	HP 5334A Universal Counter.
Voltage Source	1	-10 Vdc to +10 Vdc.
Adaptors	1 ea.	BNC female to banana jacks, BNC "tee".
Coaxial cables	3	BNC male connectors, RG58U cable.
Probes	2	10 MΩ.
50Ω Terminations	2	Feedthrough, 0.1% accuracy, 2W.

4.4 PERFORMANCE VERIFICATION PROCEDURE

The Performance Verification Procedure is given in the following paragraphs. These step-by-step procedures outline the equipment setup and interconnect. Once set up for a reading, the various parameter settings and limits are given in the specified data table in the Performance Verification Data sheets at the end of this section.

4.4.1 Procedure

The Performance Verification Procedure requires that the Model 1385 be part of a VXI system as listed in table 4-1. The Model 1385 will be controlled by the Resource Manager (VXI slot-0 controller). The computer/display device can be the Resource Manager (internal host in a Stand-Alone system) or an external host connected to the Resource Manager via the IEEE-488 (GPIB) programming bus.

4.4.2 Connector Termination

The SYNC OUT and Main OUT connectors must be terminated with 50Ω.

4.4.3 Preparation

Allow at least a 30-minute warm-up.

4.4.4 Test Equipment

The following list of test equipment is included for reference only. Substitution of any equipment is allowed as long as adequate measurement accuracy is maintained. Test equipment should always provide at least 5 times more measurement accuracy than the specification of the parameter being tested. Accessories (cables, terminations etc.) can be found in Table 4-1.

Only equipment that has been properly serviced and calibrated (traceable to NIST) according to the manufacturers specifications may be used for calibration.

Equipment	Manufacturer	Model
RF Voltmeter	Boonton	9200B
Oscilloscope	Tektronix	2465B
Universal Counter	Hewlett Packard	5334B

4.4.5 Self Calibration

This step of the calibration uses the internal DC voltage measurement capability to bring the offset and amplitude parameters within specification limits. For a proper calibration, the module cover should be installed. Allow the unit to warm up at least 20 minutes.

1) Initiate the self calibration sequence.

   Remote Cmds: *CAL?

2) Verify the unit returns a zero result indicating calibration was successful. See the self calibration description in Appendix A for an explanation of the procedures and results of the self calibration.

   Remote Query returns value of zero if self calibration was successful. A non-zero result indicates a self calibration failure.

4.4.6 Self Test

This self test uses the internal DC Voltage measurement capability to make sure that offset and amplitude parameters are within specification limits. Allow a warm up of at least 20 minutes.

1) Initiate the self test sequence.

   Remote Cmds: *TST?

2) Verify the unit returns a zero result indicating calibration was successful. See the self test description in Appendix B for an explanation of the procedures and results of the self test.

   Remote Query returns value of zero if self test was successful. A non-zero result indicates a self test failure.

4.4.7 Square Wave Symmetry

Allow the unit to warm up at least 20 minutes.

1) Configure the module as follows:

Function	Square
Frequency	1 MHz
Amplitude	5 Vp
Remote Commands:	OUTP ON
	FUNC HFSQ
	FREQ 1E6
	VOLT 5

2) Check to see that the square wave symmetry is 50% ± 1%.

4.4.8 Square Wave Quality

Allow the unit to warm up at least 20 minutes.

1) Configure the module as follows:

Function	Square
Frequency	10 MHz
Amplitude	5 Vp
Remote Commands:	OUTP ON
	FUNC HFSQ
	FREQ 10E6
	VOLT 5

2) Check to see that the aberrations are less than 5%.

3) Set amplitude to 7.5 Vp and verify rise/fall times < 9.5 ns.

4.4.9 SCM Null

1) Configure the module as follows:

Function:	Sine
Frequency:	1 MHz
Amplitude:	5 Vp
Modulation Mode:	SCM
Remote Commands:	OUTP ON
	FUNC SIN
	FREQ 1E6
	VOLT 5
	AM:MODE SCM
	AM:STATE ON

2) Verify peak to peak amplitude of output signal <75mVpp.

4.4.10 Elliptic Filter Amplitude Flatness Correction

1) Configure Arb Channel module as follows:

Function:	Sine
Frequency:	50 kHz
Amplitude:	5 Vp
Calibration State:	off
Remote Commands:	OUTP ON
	FUNC SIN
	FREQ 5E4
	VOLT 5
	CAL:STATE OFF

2) Measure the amplitude at 50 kHz and use as the reference amplitude.

3) Measure the amplitude error at 7 MHz relative to 50 kHz.
   Remote Commands: FREQ 7E6

   The amplitude should be within 2% of 50 kHz.

4) Measure the amplitude error at 13 MHz relative to 50 kHz.
   Remote Commands: FREQ 13E6

   The amplitude should be within 2% of 50 kHz.

5) Measure the amplitude error at 17 MHz relative to 50 kHz.
   Remote Commands: FREQ 17E6

   The amplitude should be within 2% of 50 kHz.

6) Measure the amplitude error at 19 MHz relative to 50 kHz.
   Remote Commands: FREQ 19E6

   The amplitude should be within 2% of 50 kHz.

4.5 ALIGNMENT PROCEDURE

The procedure given in paragraph 4.5.5 requires that the model 1385 be part of a VXI system as described in table 4-1. The model 1385 will be installed on a C-size VXI extender card and controlled by the Resource Manager (VXI slot-0 controller). The computer/display device can be the Resource Manager (internal host in a Stand-Alone system) or an external host connected to the Resource Manager via the IEEE-488 (GPIB) programming bus.

WARNING

With the covers removed, low voltage dc power supplies are exposed. Do not be misled by the term "low voltage". Under adverse conditions, potentials as low as 50 volts can cause serious injury or death.

WARNING

With the module on an extender card and the covers removed, no chassis cooling air is moving across the components. The ARB Generator has many high-speed digital logic devices and discrete analog circuits which require some cooling air for continuous operation. Do not operate the module in this configuration for more than a few minutes without directing some cooling air across the face of the ARB Board using an external utility fan. Avoid burns - do not touch components in the module.

4.5.1 Self Calibration

The model 1385 performs a Self Calibration in response to the SCPI `CAL[:ALL]?` command or to the IEEE-488.2 `*CAL?` command. The Self Calibration performs only those steps in the "full" Calibration procedure of paragraph 4.5.5 which are detailed in Appendix A. Self Calibration steps are performed by the instrument firmware at any time the operator/programmer sends the appropriate commands following the 30 minute warm-up. Self Calibration sets up various internal interconnections and uses internal time and voltage standards to store "fresh" Alignment Data which is used to optimize the unit's performance accuracy. During the Self Calibration, the unit disconnects all inputs and outputs, and at the end of Self Calibration it restores its current setup.

A Self Calibration is performed just prior to running the Performance Verification Procedure. Therefore, if there is an out-of-tolerance reading, the Self Calibration is not likely to correct it, and the "full" Alignment Procedure in paragraph 4.5.5 should be run.

4.5.2 Semi-Automated Procedure

Model 1385 alignment is partially automated in that it includes the Self Calibration in addition to manual calibration steps.

Note

The completion of the alignment procedure returns the instrument to correct alignment. ***Alignment limits and tolerances are not instrument specifications.*** *Instrument specifications are given in Section 1 of this Manual.*

4.5.3 Preparation

Obtain access to adjustable components by removing the side panel *without* the address switch cutout. Mount the module on an extender card, or plug it into a VXI chassis connector that will allow access during calibration. Provide cooling air across the face of the module using the VXIbus chassis or an external utility fan. Allow at least a 30 minute warm-up.

4.5.4 Connector Termination

When used as test points, the SYNC OUT and PULSE OUT connectors must be terminated with 50Ω.

4.5.5 Alignment Procedure

The following list of test equipment is included for reference only. Substitution of any equipment is allowed as long as adequate measurement accuracy is maintained. Test equipment should always provide at least 5 times more measurement accuracy than the specification of the parameter being tested. Accessories (cables, terminations etc.) can be found in Table 4-1.

Only equipment that has been properly serviced and calibrated (traceable to NIST) according to the manufacturers specifications may be used for calibration.

Equipment	Manufacturer	Model
RF Voltmeter	Boonton	9200B
Oscilloscope	Tektronix	2465B
Universal Counter	Hewlett Packard	5334B

Overview

The calibration of the Model 1385 can be divided into 3 sections. These are; manual adjustments, self calibration and semiautomatic adjustments. The manual and semiautomatic adjustments are intended for alignment only during a full calibration cycle. The self calibration may be invoked at any time by the user for improved accuracy. This will allow the user to correct critical parameters at the time and temperature of use.

The manual adjustments need to be performed prior to installation of the module cover. The remaining calibration can be performed with all covers installed. The self calibration may also be used to insure functionality before installing module cover.

In each step of the calibration there is a description of the remote interface commands required to set up the instrument.

4.5.5.1 Square Wave Symmetry

The square wave symmetry adjustment is performed with potentiometer R509. This adjustment sets the threshold of a comparator monitoring the filtered output of the Waveform DAC. The output of the Waveform DAC is a sinusoid from which the comparator generates a square wave. To make the symmetry adjustment the module must be installed in a VXIbus chassis with its covers removed. With the top cover in place, allow the unit to warm up at least 20 minutes.

1) Configure the module as follows:

Function	Square
Frequency	1 MHz
Amplitude	5 Vp
Remote Commands:	OUTP ON
	FUNC HFSQ
	FREQ 1E6
	VOLT 5

2) Adjust potentiometer R509 until the square wave symmetry is 50% ± 0.2%. Note that this can be easily done by adjusting R509 until the DC voltage at TP26 or TP8 is 0 volts.

4.5.5.2 Square Wave Quality

The adjustments for this parameter are R180 and R198 located on the Main Board (1100-00-3522). To make these adjustments the module must be installed in a VXIbus chassis with its covers removed. With the top cover in place, allow the unit to warm up at least 20 minutes.

1) Configure the module as follows:

Function	Square
Frequency	10 MHz
Amplitude	5 Vp
Remote Commands:	OUTP ON
	FUNC HFSQ
	FREQ 10E6
	VOLT 5

2) Adjust R180 and R198 for minimum aberrations while maintaining acceptable rise/fall time. R180 will control mainly the positive portion of the waveform and R198 will control mainly the negative portion. There will however be some interaction. The goal of these adjustments is a square wave with a smooth transition from the rising and falling edges to it's final value. The typical module will produce rise/fall times < 8.5 ns with aberrations < 255 mVpp.

4.5.5.3 Self Calibration

This step of the calibration uses the internal DC voltage measurement capability to bring the offset and amplitude parameters within specification limits. For a proper calibration, the module covers should be installed. Allow the unit to warm up at least 20 minutes.

1) Initiate the self calibration sequence.
   Remote Commands: *CAL?

2) Verify the unit returns a zero result indicating calibration was successful. See the self calibration description in Appendix A for an explanation of the procedures and results of the self calibration.

   Remote Commands:
   Remote query returns value of zero if self calibration was successful. A non-zero result indicates an self calibration failure.

4.5.5.4 SCM Null

The remaining steps of the calibration uses external equipment to measure parameters beyond the capabilities of the internal measurement system. The results of the external measurements are then returned to the Model 1385 and stored in nonvolatile calibration memory.

1) Configure the module as follows:

Function	Sine
Frequency	1 MHz
Amplitude	5 Vp
Modulation Mode	SCM
Remote Commands:	OUTP ON
	FUNC SIN
	FREQ 1E6
	VOLT 5
	AM:MODE SCM
	AM:STATE ON

2) Verify peak to peak amplitude of output signal < 75mVpp.

3) If out of tolerance, query the unit for it's current SCM zero constant. The constant will be a decimal number between 0 and 4095. There will be a null point in the amplitude vs cal constant curve. The goal of this calibration is to find the null point.
   Remote Commands: CAL:DATA:SCMZ?

4) Program a new constant and measure signal level. Iterate until amplitude is < 75mVpp.
   Remote Command: CAL:DATA:SCMZ <numeric_value>

Note

The following step should usually be postponed until the entire semi-automatic calibration is completed.

5) Store new constant
   Remote Commands: CAL:DATA:STORE

4.5.5.5 Elliptic Filter Amplitude Flatness Correction

This calibration is intended to correct for the frequency response of the elliptic filter used for sine wave generation. The procedure is to measure the amplitude error at 5 frequencies and send the appropriate correction constant to the Model 1385. The Model 1385 will then use these constants to set up a correction vs. frequency table in the non - volatile calibration memory.

The correction table is broken into 4 frequency ranges. The ranges are:

DC to the frequency of index 1;
frequency of index 1 to frequency of index 2;
frequency of index 2 to frequency of index 3;
frequency of index 3 to frequency of index 4;
frequency of index 4 to frequency of index 5.

The calibration involves determining the corrections at the boundaries of these ranges. Each correction point is assigned an index number, a frequency and a correction. The defaults are:

Index	Frequency	Correction
0	DC	1.00
1	7 MHz	1.00
2	13 MHz	0.835
3	17 MHz	1.02
4	19 MHz	0.995
5	20 MHz	1.00

The default corrections are based on a sampling of units and will not guarantee specified performance. The default range boundaries or frequencies usually will not need to be changed.

The corrections are what is determined from this procedure. If acceptable performance is not achieved, then one or more of the frequencies may need to be modified.

If this is the case, set the index frequency in the non-conforming frequency band to the nearest amplitude maxima or minima.

For example, suppose that after calibration it is determined that amplitudes in the 12 to 14 MHz frequency range do not meet specification. By measuring the amplitude vs. frequency in that band it is determined that the maximum amplitude is at 12 MHz.

Change the frequency of index 2 to 12 MHz. If required, the search for the amplitude maxima or minima should be done with the corrections turned off.

1) Configure Arb Channel module as follows:

Function	Sine
Frequency	50 kHz
Amplitude	5 Vp
Calibration State	off
Remote Commands:	OUTP ON FUNC SIN FREQ 5E4 VOLT 5 CAL:STATE OFF

2) Measure the amplitude at 50 kHz and use as the reference amplitude.

3) Measure the amplitude error at 7 MHz relative to 50 kHz. For best results this data point should use the default correction of 1.00 unless its error is > 2 %. If error is > 2% send the appropriate correction to the Model 1385.

Example: 7 MHz amplitude is 1.05 times higher than 50 kHz amplitude. Error is 0.05 (5 %). Correction factor is 1.00/1.05 or ~0.95.

Remote Commands: FREQ 7E6
CAL:DATA:AFC
1,7E6,(correction factor)

4) Measure the amplitude error at 13 MHz relative to 50 kHz. Send the appropriate correction to the Model 1385.

Remote Commands: FREQ 13E6
CAL:DATA:AFC
2,13E6,(correction factor)

5) Measure the amplitude error at 17 MHz relative to 50 kHz. Send the appropriate correction to the Model 1385.

Remote Commands: FREQ 17E6
CAL:DATA:AFC
3,17E6,(correction factor)

6) Measure the amplitude error at 19 MHz relative to 50 kHz. Send the appropriate correction to the Model 1385.

Remote Commands: FREQ 19E6
CAL:DATA:AFC
4,19E6,(correction factor)

7) Measure the amplitude error at 20 MHz relative to 50 kHz. Send the appropriate correction to the Model 1385.

Remote Commands: FREQ 20E6
CAL:DATA:AFC
5,20E6,(correction factor)

8) Store calibration constants. Turn calibration state on.

Remote Commands: CAL:DATA:STORE
CAL:STATE ON

This completes the Calibration Procedure

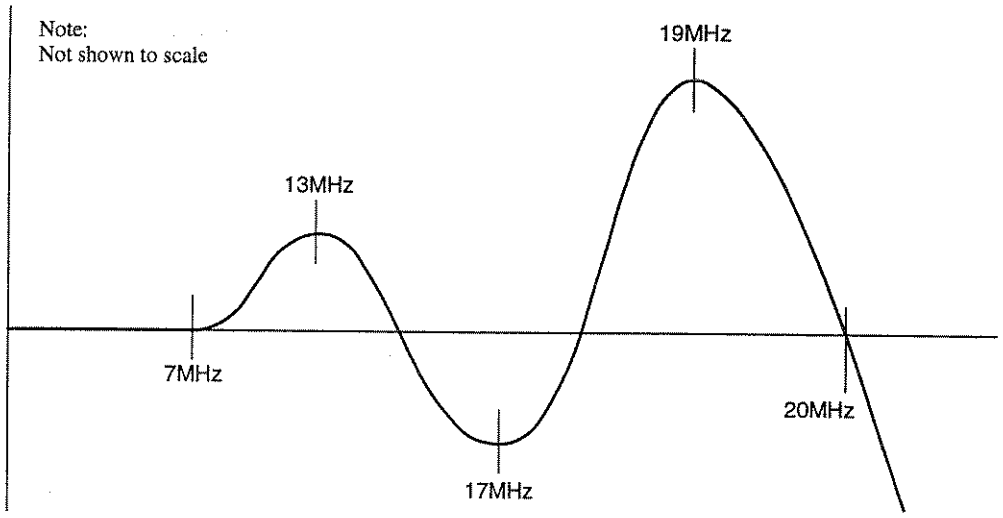


Figure 4-1. Elliptical Filter Response

5.1 DRAWINGS AND PARTS LISTS

We apologize that at the date of publication, suitable Schematics, Assembly Drawings and Parts Lists are still in the course of preparation.

A replacement copy of this manual complete with these drawings and parts lists will be forwarded as soon as they become available.

A.1 Introduction

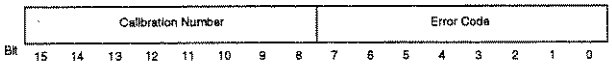
This appendix provides the Operator/Programmer with additional information needed to interpret a non-zero response value to either the `*CAL?` IEEE-488.2 Common Command or to the SCPI `CALibration[:ALL]?` query. This information supplements paragraphs 3.3.3.1 (`CALibration`) and 3.3.4 (`*CAL?`).

A.2 CALibration Query Response

The Calibration query causes an internal self calibration to be performed and a response to be placed in the Output Queue. The response to the `*CAL?` query is an ASCII string representing an integer value. The value of 0 is returned if the self calibration was successful and a non-zero value in the range of 32767 to -32768 is returned if the self calibration was unsuccessful. The interpretation of the value returned in the event of a failed self calibration is user defined.

The `*CAL?` query invokes the same internal self calibration functions and returns the same reponse as the `CALibration[:ALL]?` query.

The value of a 16-bit Self Calibration Status Word is returned in response to the calibration query. The format of the Self Calibration Status Word is shown below:



Self Calibration Status Word

The Self Calibration Status Word is split into two fields, one occupying the lower eight bits and the other occupying the upper eight bits.

The Calibration Number field contains the number of the first sub-calibration in which a failure was detected. Sub-calibration numbers range from 1 to 255. Sub-calibrations are performed in the same sequence as they are numbered.

The Error Code field contains a bit weighted code that is unique to the sub-calibration . The usage of bits 0 and 1 are common to all sub-calibrations involving voltage measurements and have the following meaning:

Error Code Bit	Meaning
0	DVM voltage not settled after 10 measurements
1	DVM voltage over range

For the interpretation of the other bits in this field refer to the paragraph describing the sub-calibration.

If no failures were detected during the self calibration then both the Calibration Number and Error Code fields will be set to zero.

The self calibration saves the state of the instrument prior to starting and restores it after the calibration is complete. While performing the self calibration the output relays remain open so as not to disturb any external devices.

Before beginning the calibration, the Model 1385 is set up with the following configuration:

Output off

Filter off

Sync Marker Off

Master clock output off

Output Attenuator /1 range

Pre-amplifier source from Waveform DAC

The internal clock frequency is set to 50 MHz and the first point in Trace Memory is selected.

The scope of each of the sub-calibrations and their Error Code formats are described as follows.

Calibration 1 Trace Memory/BERR

If the Trace Memory test that was performed at power on indicates a failure, or a bus error was generated when trying to access the ARB Card, then the self test is aborted and the Calibration Number is set to 1. Since so much of the rest of the self calibration is dependent on the proper functioning of the Trace Memory it is pointless to continue until it has been repaired.

Error Code Bit	Meaning
0	This bit may be set but it has no meaning.
1	Trace Memory power on test failed.
2	Bus error on access to hardware.

Calibration 2 AMZERO

This calibration sets the AMZERO DAC to a voltage such that the output voltage does not change while varying the Amplitude Control DAC over its full range. The output of the Waveform DAC is at mid range.

The AMZERO DAC is set to its midrange. The voltage at the PWRAMPLO test point is measured with the Amplitude Control DAC set to its minumum and maximum values, and the difference between the two measurements is taken. The AMZERO DAC is adjusted until this difference is less than 1 mV.

If the initial voltage measured with the Amplitude Control DAC set to its minimum value causes the DVM to indicate an overrange or setting error, then bit 2 is set. If the initial voltage measured with the Amplitude Control DAC set to its maximum value causes the DVM to indicate an overrange or setting error, then bit 3 is set. If more than 20 adjustments are made to the AMZERO DAC value and the difference is still greater than 1 mV, then bit 4 is set. If the AMZERO DAC value diverges (attempts to go higher than 4095 or less than 1) during adjustment, then bit 5 is set.

If any portion of this calibration fails the Calibration Number is set to 2.

Error Code Bit	Meaning
0	Voltage measurement not settled after 10 measurements
1	Voltage measurement over range
2	Voltage overrange with Amplitude Control DAC set to minimum.
4	Voltage overrange with Amplitude Control DAC set to maximum.
5	AMZERO DAC value not settling (after 20 iterations)
6	AMZERO DAC value diverging

Calibration 3 PAZERO

This calibration sets the output of the preamplifier to 0 volts when the amplitude is programmed to 0 volts by adjusting the PAZERO DAC voltage. This calibration affects the DC offset of the SUMBUS driver and allows other calibrations to be more effective since they do not need to compensate for the preamplifier offset voltage.

The PAZERO DAC is set to its mid-range and then adjusted until the voltage measured at the PREAMPTP test point is within 1 mV of 0 volts.

If any portion of this calibration fails the Calibration Number is set to 3.

Error Code Bit	Meaning
0	Voltage measurement not settled after 10 measurements
1	Voltage measurement over range
2	PAZERO DAC value not settling (after 20 iterations)
3	PAZERO DAC value diverging

Calibration 4 Offset Gain

This calibration calculates the gain of the Offset Control DACs.

The Amplitude Control DAC and the Positive and Negative Offset Control DACs are programmed for 0 volts output.

The voltage at the PWRAMPHI test point is measured with the the Positive Offset Control DAC programmed to 1/8 and 7/8 of its range, and the difference between the two measurements is taken. The gain is stored as the ratio between the Δ DAC control value and the Δ output voltage.

$$GAIN = (DAC_{hi} - DAC_{lo})/(V_{hi} - V_{lo})$$

The voltage at the PWRAMPHI test point is measured with the the Negative Offset Control DAC programmed to 1/8 and 7/8 of its range, and the difference between the two measurements is taken. The gain is calculated in the same manner as for the Positive Offset Control DAC.

If this calibration fails the Calibration Number is set to 4.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range

Calibration 5 Offset Offset

This calibration corrects for the offset of the Offset Control DACs by adjusting the OFFSFIN DAC.

The OFFSFIN DAC is set to its mid-range and the output offset voltage is programmed to 0.5 volts. The OFFSFIN DAC is adjusted until the output offset voltage measured at the PWRAMPLO test point is within 1 mV of 0.5 volts. The OFFSFIN DAC value is saved to be used to correct positive output offset voltages.

The OFFSFIN DAC is set to its midrange and the output offset voltage is programmed to -0.5 volts. The OFFSFIN DAC is adjusted until the output offset voltage measured at the PWRAMPLO test point is within 1 mV of -0.5 volts. The OFFSFIN DAC value is saved to be used to correct for negative output offset voltages.

If any portion of this calibration fails the Calibration Number is set to 5.

Error Code Bit	Meaning
0	Voltage measurement not settled after 10 measurements
1	Voltage measurement over range
2	Positive offset correction diverging
3	Positive offset correction not settling
4	Negative offset correction diverging
5	Negative offset correction not settling

Calibration 6 Amplitude Gain

This calibration calculates the gain of the Amplitude Control DAC for the output of the Waveform DAC.

The Amplitude Control DAC and the Positive and Negative Offset Control DACs are programmed for 0 volts output.

The positive and negative peak amplitude voltages at the PWRAMPHI test point are measured with the Amplitude Control DAC programmed to 1/8 and 7/8 of its range. The difference between the peak to peak amplitude voltage in the two ranges is calculated. The gain is stored as the ratio between the Δ DAC control value and the Δ peak to peak output voltage.

$$GAIN = (DAC_{hi} - DAC_{lo})/(Vpp_{hi} - Vpp_{lo})$$

If this calibration fails the Calibration Number is set to 6.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range

Calibration 7 Amplitude Offset

This calibration corrects for the offset of the Amplitude Control circuits by adjusting the MO_AMPL DAC.

The MO_AMPL DAC is set to its midrange and the output amplitude voltage is programmed to 0.5 volts. The MO_AMPL DAC is adjusted until the peak to peak output amplitude voltage measured at the PWRAMPLO test point is within 1 mV of 0.5 volts. The MO_AMPL DAC value is saved to be used to correct output amplitude voltages.

If any portion of this calibration fails the Calibration Number is set to 7.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range
2	Amplitude offset correction value diverging
3	Amplitude offset correction not settling (after 20 iterations)

Calibration 8 High Frequency Square Wave Gain

This calibration calculates the gain of the Amplitude Control DAC for the output of the Square Wave Generator. It follows exactly the same procedure as the Amplitude Gain calculation for the output of the Waveform DAC. The gain correction factor calculated in this calibration is used only when the source of the input signal to the preamplifier is the Square Wave Generator.

If any portion of this calibration fails the Calibration Number is set to 8.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range

Calibration 9 High Frequency Square Wave Offset

This calibration corrects for the offset of the Square Wave Generator by adjusting the AMZERO DAC.

The output of the Square Wave Generator is input to the Preamp. The AMZERO DAC is set to its mid-range and the output amplitude voltage is programmed to 5 volts. The AMZERO DAC is adjusted until the difference between the positive and negative peak amplitude voltages measured at the PWRAMPHI test point is within 1 mV of 0 volts. The AMZERO DAC value is saved to be used to correct output amplitude voltages when the High Frequency Square Wave is in use.

If any portion of this calibration fails the Calibration Number is set to 9.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range
2	Offset correction value diverging
3	Offset correction not settling (after 20 iterations)

Self Test

Appendix B

B.1 Introduction

This appendix provides the Operator/Programmer with additional information needed to interpret a non-zero response value to either the `*TST?` IEEE-488.2 Common Command or to the SCPI `TEST[:ALL]?` query. This information supplements paragraphs 3.3.3.8 (`TEST`) and 3.3.4 (`*TST?`).

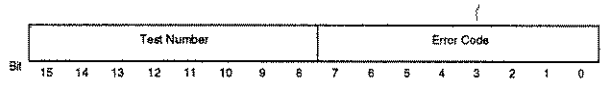
B.2 TEST Query Response

The response to the *TST? query is an ASCII string representing an integer value. The value of 0 is returned if the self test passed and a non-zero value in the range of 32767 to -32768 is returned if the self test failed. The interpretation of the value returned in the event of a failed self test is defined by the module manufacturer.

The *TST? query invokes the same internal self test functions and returns the same reponse as the TEST[:ALL]? query documented below.

TEST[:ALL]?

The TEST[:ALL]? query returns a detailed error code indicating the nature of any failure. The value of a 16-bit Self Test Status Word is returned in response to the TEST[:ALL]? query. The format of the self test status word is shown below:



Self Test Status Word

The Self Test Status Word is split into two fields, one occupying the lower eight bits and the other occupying the upper eight bits.

The Test Number field contains the number of the first sub-test in which a failure was detected. Sub-test numbers range from 1 to 255.
Sub-tests are performed in the same sequence as they are numbered.

The Error Code field contains a bit weighted code that is unique to the sub-test. The usage of bits 0 and 1 are common to all sub-tests involving voltage measurements and have the following meaning:

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	Voltage over range

For the interpretation of the other bits in this field refer to the paragraph describing the sub-test.

If no failures were detected by the self test, then both the Test Number and Error Code fields will be set to zero.

The self test saves the state of the instrument prior to starting and restores it after the test is complete. While performing the self test the output relays remain open so as not to disturb any external devices.

Most of the self test consists of setting up the hardware into a known state, routing a voltage to the 1385's internal DVM and measuring the voltage. The voltages at the input to the DVM are scaled versions of test point voltages.

The self test is performed in stages in order of a circuit's controllabilty from the digital section. DAC's and relays are checked first, amplifiers and filters last. The test stops on the test number of the first failure it detects.

Before beginning any tests the instrument is set up with the following configuration:

Output off

Filter off

Sync Marker Off

Master clock output off

Output Attenuator /1 range

Pre-amplifier source from Waveform DAC

The internal clock frequency is set to 50 MHz and the hardware is set up to select the first point in Trace Memory.

The scope of each of the sub-tests and their Error Code formats are described below.

Test 1 Trace Memory/BERR

If the Trace Memory test that was performed at power on indicates a failure, or a bus error was generated when trying to access the ARB Generator Card, then the self test is aborted and the Test Number is set to 1. Since so much of the rest of the self test is dependent on the proper functioning of the Trace Memory it is pointless to continue until it has been repaired.

Error Code Bit	Meaning
0	This bit may be set but it has no meaning
1	Trace Memory power on test failed
2	Bus error on access to hardware

Test 2 Analog to Digital Convertor

This test checks the operation of the ICL7109 (U21) DVM chip. Channel 7 on the CD4041BE (U20) is selected to apply 0 volts to the input of the ICL7109. The voltage is measured and verified that it is within the range 0 ± 0.002 volts.

If this test fails bit 2 of the Error Code is set to 1 and the Test Number is set to 2.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Analog to Digital Convertor failure

Test 3 Frequency Synthesizer

This test checks the level of the LockDetect output of the MC145148 (U27) to make sure it can detect the locked and unlocked condition of the Phase Locked Loop. The LD output can be monitored directly through bit 2 of the Status Register. The LD signal is not a level. Instead it is mostly high if the loop is locked and mostly low if it is unlocked. In order to test the LD output it is sampled 1000 times and a count is kept of the number of low samples. The ratio of low samples to total samples indictates whether the loop is locked or unlocked.

The Frequency Synthesizer is meant to generate frequencies between 50 and 100 MHz with a loop frequency of 1 kHz when in RASTer mode. A separate loop filter allows the loop frequency to be changed to 1e7/90 Hz (111.111 kHz) to reduce phase jitter at certain frequencies at the expense of decreased frequency resolution. This loop is used in CW mode.

The loop is considered locked if the LD signal is low for less than 50 samples out of 1000. The loop is considered unlocked if more than 400 samples out of 1000 are low

If any portion of this test fails the Test Number is set to 4.

Error Code Bit	Meaning
0	CW loop lock failure
1	CW loop unlock failure
2	RASTer loop lock failure
3	RASTer loop unlock failure

Test 4 Offset Control DACs

This test checks the maximum positive and negative offset voltage levels at the OFFSTAMP test point. The voltage is routed to the ICL7109 (U21) by selecting channel 2 on the CD4041BE (U20).

For all voltage measurements the magnitude of the voltage expected at the OFFSTAMP test point is:

$$V_{expected} = (5.5/7.5) * 4.5 * 0.2659$$

The offset voltage is programmed to its positive maximum. The voltage is measured and its magnitude is verified to be greater than $V_{expected}$ - 10% and less than $V_{expected}$ + 10%.

If the voltage is less than $V_{expected}$ - 10%, then bit 2 of the Error Code is set to 1. If the voltage is greater than $V_{expected}$ + 10%, then bit 3 of the Error Code is set to 1.

The offset voltage is programmed to its negative maximum. The voltage is measured and its magnitude is verified to be greater than $V_{expected}$ - 10% and less than $V_{expected}$ + 10%.

If the voltage is less than $V_{expected}$ - 10%, then bit 4 of the Error Code is set to 1. If the voltage is greater than $V_{expected}$ + 10%, then bit 5 of the Error Code is set to 1.

If any portion of this test fails the Test Number is set to 4.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Positive offset control voltage under-range
3	Positive offset control voltage overrange
4	Negative offset control voltage under-range
5	Negative offset control voltage overrange

Test 5 Amplitude Control DAC

This test checks that the Amplitude Control DAC (U25) is functional and is able to generate control voltages within the expected range. The control voltage is sampled at the AMSIG test point which is routed to the ICL7109 (U21) by selecting channel 1 on the CD4051BE (U20).

The Amplitude Control DAC is programmed to generate the maximum output amplitude voltage. The voltage is measured and verified to be greater than 1.0 volt and less than 1.2 volts. If the voltage is less than 1.0 volt, then bit 2 of the Error Code is set to 1. If the voltage is greater than 1.0 volt + 20%, then bit 3 of the Error Code is set to 1.

The Amplitude Control DAC is programmed to generate the minimum output amplitude voltage. The voltage is measured and verified to be in the range 0.0V ± 0.015V. If the voltage is less than -0.015 volts, then bit 4 of the Error Code is set to 1. If the voltage is greater than +0.015 volts, then bit 5 of the Error Code is set to 1.

If this test fails the Test Number is set to 5.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Maximum amplitude control voltage under-range
3	Maximum amplitude control voltage overrange
4	Minimum amplitude control voltage under-range
5	Minimum amplitude control voltage overrange

Test 6 Preamplifier

This test checks that the voltage at the output of the preamplifier is in the expected range for full scale output amplitude. The preamplifier voltage is routed to the ICL7109 (U21) by selecting channel 4 on the CD4051BE (U20).

This test requires that the address generator chip be set up so that its address lines are all 0 in order to select the first point in Trace Memory. The contents of this point can then be used to set the output voltage of the preamplifier through the Waveform DAC. The Preamplifier signal source is set to the output of the Waveform DAC.

The voltage difference between the positive and negative peaks should be greater than:

$$V_{expected} = 5.5 * 2.0 * (1.0/5.0)$$

and less than $V_{expected}$ + 20%. The offset voltage should be less than 15% of $V_{expected}$.

The Amplitude Control DAC is set to its full scale output voltage by programming it with 0xFFF. The first point in Trace Memory is programmed with the code 0xFFF to select the positive peak output of the Waveform DAC. The voltage is measured and stored as V_{pos}.

The first point in Trace Memory is programmed with the code 0x001 to select the negative peak output of the Waveform DAC.

The voltage is measured and stored as V_{neg}. The first point in Trace Memory is programmed with the code 0x800 to select the midrange output of the Waveform DAC. The voltage is measured and stored as V_{offs}.

The difference between the positive and negative peak voltages is calculated as shown below:

$$V_{diff} = V_{pos} - V_{neg}$$

V_{diff} is verified to be greater than $V_{expected}$ and less than $V_{expected}$ + 20%. If the voltage is less than $V_{expected}$, then bit 2 of the Error Code is set to 1. If the voltage is greater than $V_{expected}$ + 20%, then bit 3 of the Error Code is set to 1.

The magnitude of V_{offs} is verified to be less than 15 % of $V_{expected}$. If the offset is too high on the positive side, then bit 4 of the Error Code is set to 1. If the offset is too high on the negative side, then bit 5 of the Error Code is set to 1.

If any portion of this test fails the Test Number is set to 6.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Peak voltage spread under-range
3	Peak voltage spread overrange
4	Offset voltage positive over-range
5	Offset voltage negative over-range

Test 7 Square Wave Generator

This test checks that the Square Wave Generator is functioning by performing the exact same operations as were performed in the Preamplifier test but with the Preamplifier signal source set to the Square Wave Generator.

If any portion of this test fails the Test Number is set to 7.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Peak voltage spread under-range
3	Peak voltage spread over-range
4	Offset voltage positive over-range
5	Offset voltage negative over-range

Test 8 Filters

This test checks DC operation of the Bessel and Elliptic filters by monitoring voltage levels at the output of the Pre-amplifier at the PREAMP test point. It checks that the voltage at the PREAMP test point does not change by more than 1% when the filters are enabled.

At the start of the test the offset voltage is programmed to 0 volts, the amplitude is programmed to its maximum value, and the first point in Trace Memory is programmed with the code 0xFFF. The voltage at the PREAMP test point is measured with the filters off and saved as a reference.

The Bessel filter is enabled and the voltage is measured and verified to be within 1% of the reference. If the voltage is more than 1% lower, then bit 2 of the Error Code is set to 1. If the voltage is more than 1% higher, then bit 3 of the Error Code is set to 1.

The Elliptic filter is enabled and the voltage is measured and verified to be within 1% of the reference. If the voltage is more than 1% lower, then bit 4 of the Error Code is set to 1. If the voltage is more than 1% higher, then bit 5 of the Error Code is set to 1.

If this test fails the Test Number is set to 8.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Bessel filter output more than 1% lower than reference
3	Bessel filter output more than 1% higher than reference
4	Elliptic filter output more than 1% lower than reference
5	Elliptic filter output more than 1% higher than reference

Test 9 Power Amplifier

This test checks the gain of the Power Amplifier and voltage levels at the output of the Power Amplifier at the PWRAMP test point. The voltage at the PWRAMP test point is routed to the ICL7109 (U21) by selecting channel 6 on the CD4051BE (U20).

The gain of the Power Amplifier is checked by measuring the spread between positive and negative peak amplitude voltages at its input and output. The ratio between the two measurements should be equal to the gain. The gain should be 10 ± 5%.

The voltage levels are checked by measuring the spread between positive and negative peak offset voltages at its output. The voltage difference between the positive and negative peaks should be greater than:

$$V_{expected} = (5.5/2.0) * 2.0 * (9.0/79.27)$$

and less than 20% of this value. The offset of the voltage difference should be less than 10% of this value.

Power Amplifier Gain

The offset voltage is programmed to 0 volts and amplitude is programmed to half its maximum to avoid saturating the Power Amplifier.

The first point in Trace Memory is programmed with the code 0xFFF to select the positive peak output of the Waveform DAC. Voltages are the PREAMP and PWRAMP test points are measured and stored.

The first point in Trace Memory is then programmed with the code 0x001 to select the negative peak output of the Waveform DAC. Voltages at the PREAMP and PWRAMP test points are again measured and stored.

The spread between the peak voltages at the PREAMP test point is calculated and stored as V_{pre_diff}. The spread between the peak voltages at the PWRAMP test point is calculated and stored as V_{pwr_diff}. V_{pwr_diff} is then corrected for the scaling resistors at the PWRAMP test point. The ratio between V_{pwr_diff} and V_{pre_diff} is verified to be within the range 10 ± 5%. If this check fails then bit 2 of the Error Code is set to 1.

Power Amplifier Output Levels

The Waveform DAC data inputs are set to 0x800 to set the amplitude voltage to 0 volts, and the Amplitude Control DAC is programmed for 0 volts amplitude.

The Offset Control DAC is programmed to half its positive maximum to avoid saturating the Power Amplifier. The voltage at the PWRAMP test point is measured and stored as V_{pos}.

The Offset Control DAC is programmed to half its negative maximum and the voltage at the PWRAMP test point is measured and stored as V_{neg}.

The difference between the positive and negative peak voltages is calculated and stored as Vdiff as shown:

$$V_{diff} = V_{pos} - V_{neg}$$

V_{diff} is verified to be greater than $V_{expected}$ and less than $V_{expected}$ + 20%. If this check fails then bit 3 of the Error Code is set to 1.

The offset from zero of the center point between the positive and negative peak voltages is calculated and stored as V_{offs} as shown:

$$V_{offs} = (V_{pos} + V_{neg})/2$$

V_{offs} is verified to be less than 10 % of $V_{expected}$. If this check fails then bit 4 of the Error Code is set to 1.

If any portion of this test fails the Test Number is set to 9.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	Gain failure
4	Level failure
5	Offset failure

Test 10 Output Attenuator

This test checks the voltage level at the output of the Output Attenuator stage at the PAOUTTST test point. It identifies problems with the Output Attenuator by placing it in its /1, /2, /4 and /16 ranges. The voltage is routed to the ICL7109 (U21) by selecting channel 0 on the CD4051BE (U20).

At the start of the test the Offset Control DAC is set to 0x800 to give 0 volts of offset, the Amplitude Control DAC is programmed to 0xFFF and the Waveform DAC (U47) data inputs, DB[12:01], are set to 0xB80. This should generate an output voltage that will not saturate the Power Amplifier.

The /1 output attenuator range is selected. This voltage is measured and its value is stored as V_{test}.

The /2 output attenuator range is selected. The voltage is measured and compared to $V_{test}/2$. If the voltage is higher by more than 5% then bit 2 of the Error Code is set to 1. If the voltage is lower by more than 5% then bit 3 of the Error Code is set to 1.

The /4 output attenuator range is selected. The voltage is measured and compared to $V_{test}/4$. If the voltage is higher by more than 5% then bit 4 of the Error Code is set to 1. If the voltage is lower by more than 5% then bit 5 of the Error Code is set to 1.

The /16 output attenuator range is selected. The voltage is measured and compared to $V_{test}/16$. If the voltage is higher by more than 5% then bit 6 of the Error Code is set to 1. If the voltage is lower by more than 5% then bit 7 of the Error Code is set to 1.

If any portion of this test fails the Test Number is set to 10.

Error Code Bit	Meaning
0	Voltage not settled after 10 measurements
1	DVM over range
2	-06 dB attenuator over voltage
3	-06 dB attenuator under voltage
4	-12 dB attenuator over voltage
5	-12 dB attenuator under voltage
6	-24 dB attenuator over voltage
7	-24 dB attenuator under voltage

Test 11 MARKER OUTPUT

This test checks the operation of the Marker output control logic and buffers. The POSITION and SYNC Marker signals are brought into the Status Register, allowing their levels to be read by the CPU.

The generation of the Marker outputs from the upper bits of Trace Memory is tested. The Sync Marker source is selected to come from the state of bit 14 of Trace Memory.

The first point in Trace Memory is programmed with the code 0x0800 to set the SYNC (bit 14) and POSITION (bit 15) bits low. The Status Register is read and the level of the Sync and Position Markers is verifed. If the level of the Position Marker is not low, then bit 0 of the Error Code is set to 1. If the level of the Sync Marker is not low, then bit 1 of the Error Code is set to 1.

The first point in Trace Memory is programmed with the code 0x8800 to set the POSITION (bit 15) bit high. The Status Register is read and the level of the Position Marker is verifed. If the level of the Position Marker is not high, then bit 2 of the Error Code is set to 1.

The first point in Trace Memory is programmed with the code 0x4800 to set the SYNC (bit 14) bit high. The Status Register is read and the level of the Sync Marker is verifed. If the level of the Sync Marker is not high, then bit 3 of the Error Code is set to 1.

Next the generation of the Sync Marker output from the comparator is tested.

The first point in Trace Memory is programmed with the code 0x0000 to set the input to the comparator (U45) low. This in turn causes the level of the Sync Marker output to be low. The Status Register is read and the level of the Sync Marker is verifed. If the level of the Sync Marker is not low, then bit 4 of the Error Code is set to 1.

The first point in Trace Memory is programmed with the code 0x0FFF to set the input to the comparator (U45) high. This in turn causes the level of the Sync Marker output to be high. The Status Register is read and the level of the Sync Marker is verifed. If the level of the Sync Marker is not high, then bit 5 of the Error Code is set to 1.

If any portion of this test fails the Test Number is set to 11.

Error Code Bit	Meaning
0	Position marker not low
1	BBITs Sync marker not low
2	Position marker not high
3	BBITs Sync marker not high
4	ZCRoss Sync marker not low
5	ZCRoss Sync marker not high

TEST:RAM?

This query checks the integrity of Trace Memory by peforming destructive write/read operations. These operations attempt to locate stuck address and data lines down to the signal level. This test is also performed when the instrument is first powered on.

Hardware Configuration

The Trace Memory consists of a pair of 8-bit wide static RAM chips. The chips are paired such that one chip stores the high byte and the other stores the low byte of a 16-bit word. Depending on the configuration, the size of the RAM chip is 128K bytes. Thus the total size of Trace Memory is 128K words or 512K words.

The RAMs are electrically isolated from the CPU bus by the address generator and two 74HCT245 transceivers. The address lines are buffered by the address generator and the data lines are buffered by the 74HCT245s. Write enable and output enable signals are generated by logic on the Arb Engine card.

Firmware Configuration

The Trace Memory test is performed in four stages. The first stage looks for stuck data lines, the second stage looks for shorts between address and data lines, the third stage looks for shorts between data lines, and the fourth stage looks for stuck address lines. A failure at any stage of the test will cause the test to terminate. This means that only the first fault will be reported in a board with multiple faults.

The size of Trace Memory is determined at power on by writing unique patterns to word 0 and word 32768. If the contents of word 0 change, then it is assummed that there are only 32k words of Trace Memory. If the contents of word 0 do not change, then it is assumed there are 128k words of Trace Memory. If the RAM chips are not installed or not accessable, this check will indicate that there are 128k bytes of Trace Memory. The size determined by this check is used by the test functions.

Each subtest is described in detail below.

Subtest 1

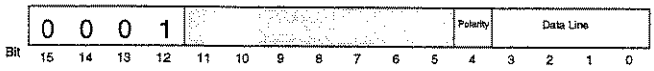
This test checks for shorts or opens on the data line connections.

The first portion of this test checks for data lines that are stuck low. The pattern '0xFFFF' is written to the last word in Trace Memory. The last word in Trace Memory is chosen for this test because when this address is accessed all address lines are high. This prevents a misdiagnosis in the case where a data line is shorted to an address line.

The contents of the last word in Trace Memory are read back and checked for low bits. If any bits are low the position of the first low bit found starting from D0 is recorded and the polarity field is set to 0.

The second portion of this test checks for data lines that are open or stuck high. The pattern '0x0000' is written to the first word in Trace Memory. The first word in Trace Memory is chosen for this test because when this address is accessed all address lines are low. This prevents a misdiagnosis in the case where a data line is shorted to an address line.

The contents of the first word in Trace Memory are read back and checked for high bits. If any bits are high the position of the first high bit found starting from D0 is recorded and the polarity field is set to 1.



Subtest 1 Response Format

If a failure is detected in this subtest, the code 1 is placed in the Test Number field. The encoding of the Data Line and Polarity fields is as shown below. The Bit Position field identifies the stuck data line. The Polarity field records whether the data line was stuck low or high.

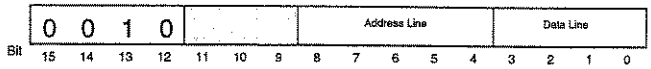
Data Line	CPU Signal	Polarity	Meaning
0	D00		
1	D01	0	Stuck low
2	D02	1	Stuck high
3	D03		
4	D04		
5	D05		
6	D06		
7	D07		
8	D08		
9	D09		
10	D10		
11	D11		
12	D12		
13	D13		
14	D14		
15	D15		

Subtest 2

This tests checks for shorts between address and data lines. This test can only be performed if subtest 1 passed since errors in subtest 1 would cause a misdiagnosis in this subtest.

This test first determines if there is a short between an address line and a data line by writing the pattern '0xFFFF' to address 0. If the pattern read back from RAM has one or more low bits, then an address/data line short is assumed. The first low bit starting from D0 is recorded in the Data Line field.

Next the test determines which address line is shorted to the data line. This is done by walking a high though the address lines until the affected data bit is read as a high. The address line is recorded in the Address Line field.



Subtest 2 Response Format

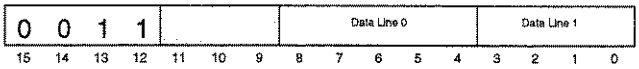
If a failure is detected in this subtest, the code 2 is placed in the Test Number field. The encoding of the Data Line field is the same as used for the Bit Position field in subtest 1. It identifes the shorted data line. The encoding of the Address Line field is as shown below. It identifes which address line is shorted to the data line.

Address Line	CPU Signal Name
1	A01
2	A02
3	A03
4	A04
5	A05
6	A06
7	A07
8	A08
9	A09
10	A10
11	A11
12	A12
13	A13
14	A14
15	A15
16	A16
17	A17

Subtest 3

This tests checks for shorts between data lines. This test can only be performed if subtest 2 passed, since errors in subtest 2 would cause a misdiagnosis in this subtest.

This test determines if there is a short between data lines by walking a '0' through the data lines. It is assumed that the data lines pull low harder than they pull up, so that a data line driving low will prevail over a data line driving high. If the pattern read back from RAM has more than one bit low, then the data lines corresponding to the low bits determine which data lines are shorted.



Subtest 3 Response Format

If a failure is detected in this subtest, the code 3 is placed in the Test Number field. The encoding of the Data Line fields is the same as used in subtest 1. These fields identify data lines which are shorted together.

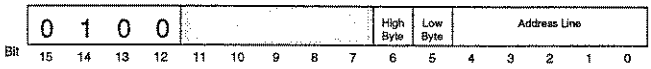
Subtest 4

This test checks for continuity of the address lines to each RAM chip. This test can only be performed if subtest 3 passed since errors in subtest 3 would cause a misdiagnosis in this subtest. Because the address lines are common to both the upper and lower bytes, we need to detect not only which address line has the discontinuity, but also whether that address line is connected to the RAM chip containing the high byte, the low byte or both.

First the pattern 0x0000 is written to word 0. Then a unique pattern is written into both the upper and lower bytes at the series of addresses created when walking a '1' across the address lines. After each write the contents of word 0 are checked to see if they changed. If an address line is stuck, then the pattern read from word 0 will change. The patterns in the upper and lower byte are checked to see if the stuck address line affects only one or both RAM chips.

The addresses are derived by walking a '1' across the address lines of the RAM chips, starting with CPU address line A01. Thus the addresses are in the sequence:

Trace RAM Base Address + {0x0000, 0x0002, 0x0004, 0x0008, ... }



Subtest 4 Response Format

If a failure is detected in this subtest, the code 4 is placed in the Test Number field. The encoding of the Address Line field is the same as used in subtest 2. It identifies the stuck address line. The High Byte field is set to '1' if the high byte RAM chip was affected by the stuck address line. The Low Byte field is set to '1' if the low byte RAM chip was affected by the stuck address line.

SCPI Conformance Information

Appendix C

C.1 Introduction

Appendix C contains additional information for the Standard Commands for Programmable Instrumentation (SCPI) that is not covered in detail in the model 1385's Command Reference, Section 3, paragraph 3.3.3 of this manual. It also supplies the "SCPI Conformance Information" as required by the specification.

C.2 Reference Information

Low Level Interface Commands

The Arb responds to the following low level VXIbus interface commands. These are issued by its commander. Refer to the VXIbus specification for further information.

ANOP Abort Normal Operation

This command causes the Arb to cease its normal operation as fast as possible. It resets the *ready* bit in the Status register to 0. The unit is reconfigured to the default state; the power-up memory test is NOT performed. It has the same effect as ***RST** and ***CLS**.

BNOP Begin Normal Operation:

This commands the Arb to begin normal operation. The *ready* bit in the Status register is set to 1 and the device is ready to receive data.

CLE Clear

Upon receiving the clear command, the Arb removes all output data in the VXI Data Low register. The Arb will discard all data due for output from operations currently in progress and become ready to receive a new command; the input and output queues are cleared. The *read ready* bit in the Response register will be reset to 0. The *Err** bit of the Response register will be de-asserted and any pending *read protocol error* command responses will be canceled.

ENOP End Normal Operation

The Arb ends its normal operation in an orderly manner without any time limit constraints. Incoming signals are no longer processed. The *ready* bit of the status register is reset to 0. The current Arb configuration is maintained.

ICOM Identify Commander

The Commander uses this command to tell the Arb its logical address.

RPE Read Protocol Error

The VXIbus commander uses this command to query the cause of the last protocol error.

RPR Read Protocol

The VXIbus commander uses this command to find out what protocols in addition to the Word Serial protocol that the Arb supports. The Arb supports I, I4, EG, and RG.

RSTB Read Status Byte

This command is used by the Commander to read the status byte from the Arb.

Model 1385 SCPI Command Trees

Appendix D provides the programmer/operator with a complete set of SCPI Command trees for the model 1385. These trees are an alternate method (to the Command Table) of constructing syntactically correct SCPI commands.

Model 1385 SCPI Command Table

Table 3-2, SCPI Command Summary, and Table C-1, model 1385 SCPI Command Syntax, provide the programmer/operator with a complete set of SCPI Commands for the model 1385. The Command Table is the primary means of constructing syntactically correct SCPI commands.

C.3 SCPI Conformance Information

This Appendix contains compliance data as required by the SCPI 1992 Specification, Volume 1: Syntax and Style. Specifically, paragraph 4.2.3, Documentation Requirements, specifies the Conformance Information requirements for SCPI products.

C.3.1 Model 1385 SCPI version

The Model 1385 VXIbus Arbitrary Waveform Synthesizer has been designed to comply with SCPI Version 1992.0, dated February 1992.

C.4 Model 1385 SCPI Command Syntax

The SCPI specification, Version 1992.0, defines three types of SCPI commands which may be used in a SCPI product: Confirmed Commands, Approved Commands, and commands which are not part of the SCPI definition.

C.4.1 SCPI Confirmed Commands

Confirmed Commands are those commands which are published in the SCPI 1992 Specification, Volume 2: Command Reference. Refer to Table C-1 for the complete syntax of Model 1385 SCPI commands. Model 1385 Confirmed Commands will be identified with the notation "Confirmed" in the third column.

C.4.2 SCPI Approved Commands

Approved Commands are those commands which have been approved by the SCPI Consortium, but are not published in the SCPI 1992 Specification. Refer to Table C-1 for the complete syntax of Model 1385 SCPI commands. Model 1385 Approved Commands will be identified with the notation "Approved" in the third column.

C.4.3 Commands not part of the SCPI Specification

The SCPI Specification does allow products using the SCPI language to have specialized commands included which arenot yet listed in the SCPI specification. Refer to Table C-1 for the complete syntax of Model 1385 SCPI commands. Any Model 1385 commands which are not in the SCPI specification will be identified with the notation "Not SCPI Approved" in the third column.

C.4.4 Incomplete Command Implementation

The SCPI definition specifies each command completely, and if the command keyword is at the *leaf node*, it specifies the parameter data and query responses. In some cases, a product may not implement all of the choices given in the specification.

For example, when parameter character data is in the form of a list of choices, the product's hardware may not support all of those choices:

```
[SOURce]:FREQuency:MODE
  <CW|FIXed|SWEep|LIST|SENSe>
```

In this example, a complete list of possible Frequency Modes is given. However, the product's feature set may want to have a settable Frequency Mode in order to set a fixed frequency, and another Mode to sweep a frequency. The other choices, LIST and SENSe, may not have any hardware to support them. In this case, the SCPI Syntax Table (see Table C-1) would use footnotes to indicate partial conformance to the SCPI Specification.

Table C-1. Model 1385 Command Summary

KEYWORD	PARAMETER FORM	NOTES
CALibration		Confirmed
[:ALL]?		Confirmed
:DATA	<block>	Confirmed
:AFCorrection	<point>,<frequency>,<gain>	Not SCPI Approved
:AMPLitude		Not SCPI Approved
[:GAIN]	<numeric_value> (0.0, **0.0**, 1000)	Not SCPI Approved
:OFFSet	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:AMZero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:OFFSet		Not SCPI Approved
[:GAIN]	<**POS**itive|NEGative>,<numeric_value>	Not SCPI Approved
:OFFSet	<**POS**itive|NEGative>,<numeric_value>	Not SCPI Approved
:PAZero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:SCMZero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:STORe		Not SCPI Approved
:STATe	<Boolean_data> (**ON**)	Confirmed
INITiate		Confirmed
:CONTinuous	<Boolean_data> (**ON**)	Confirmed
[:IMMediate]		Confirmed
OUTPut		Confirmed
:CLOCk		Not SCPI Approved
:FREQuency	<numeric_value> (1e-1, **1e3**, 1e8)	Not SCPI Approved
:SOURce	<**RAST**er|SYNThesizer>	Not SCPI Approved
:ECLTrg<n>		Confirmed
[:STATe]	<Boolean_data> (**OFF**)	Confirmed
:FILTer		Confirmed
[:LPASs]		Confirmed
:TYPe	<**BESS**el|ELLiptic>	Confirmed
[:STATe]	<Boolean_data>	Confirmed
[:STATe]	<Boolean_data> (**OFF**)	Confirmed
:TRIGger		Not SCPI Approved
:END		Not SCPI Approved
[:STATe]	<Boolean_data> (**OFF**)	Not SCPI Approved
:MARKer	<**TRIG**ger|POSition>	Not SCPI Approved
:SOURce	<**BIT**|BCOMplete|LCOMplete|INTernal>	Not SCPI Approved
:TTLTrg<n>		Confirmed
[:STATe]	<Boolean_data> (**OFF**)	Confirmed
RESet		Confirmed
[SOURce]		Confirmed
:AM		Confirmed
[:STATe]	<Boolean_data> (**OFF**)	Confirmed
:MODE	<**AM**>|<SCM>	Not SCPI Approved
:CLOCk		Not SCPI Approved
:CONFigure	<**INP**ut|OUTPut>	Not SCPI Approved

Table C-1. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:FREQuency		Confirmed
[:CW|FIXed]	<numeric_value> (1e-6, **1e3**, 2.5e7)	Confirmed
:MANual	<numeric_value> (1e-1, **1e3**, 2e7)	Confirmed
:MODE	<**CW**|SWEep|LIST>	Confirmed[2]
:RASTer	<numeric_value> (1e-1, **5e7**, 5e7)	Not SCPI Approved
:STARt	<numeric_value> (1e-1, **1e3**, 2e7)	Confirmed
:STOP	<numeric_value> (1e-1, **1e5**, 2e7)	Confirmed
:FUNCtion		Confirmed
[:SHAPe]	<shape_name>	Confirmed
:SQUare		
:DUTY	<numeric_value>	Not SCPI Approved
:TRIangle		
:SYMMetry	<numeric_value>	Not SCPI Approved
:USER	<trace_name>	Not SCPI Approved
:MODE	<**FIX**ed|SEQuence>	Confirmed[1]
:LIST		Confirmed
:FREQuency	<value>,<list_index> (1e-1, **1e3**, 2e7),(0,1023)	Not SCPI Approved
:POINts	<value> (1, **1**, 1024)	Not SCPI Approved
:MARKer		Confirmed
:POSition		Not SCPI Approved
:AOFF	<trace_name>	Not SCPI Approved
:POINt	<trace_name>, <point_index>	Not SCPI Approved
:SYNC		Not SCPI Approved
:SOURce	<**ZCR**oss|BBITs>	Not SCPI Approved
[:STATe]	<Boolean_data> (**ON**)	Not SCPI Approved
:TRIGger		Not SCPI Approved
[:STATe]	<trace_name>,<Boolean_data> (**OFF**)	Not SCPI Approved
:PHASe		Confirmed
[:ADJust]	<numeric_value> (-180, **0**, 180)	Confirmed
:LOCK	<Boolean_value> (**OFF**)	Not SCPI Approved
:ROSCillator		Confirmed
:SOURce	<**INT**ernal|EXTernal|ECLTrg<n>>	Confirmed[1]
:SEQuence		Not SCPI Approved
:ADVance	<**SYNC**hronous|ASYNchronous>,<list_index>	Not SCPI Approved
:CRATio	<numeric_value>,<list_index>	Not SCPI Approved
:DWELl	<numeric_value>,<list_index>	Not SCPI Approved
:FUNCtion	<trace_name>,<list_index>	Not SCPI Approved
:LENGth	<numeric_value>	Not SCPI Approved
:SEGMent	<trace_name>,<numeric_value>,	
	<numeric_value>, <**AUTO**matic|TRIGgered>,	
	<**SYNC**hronous|ASYNchronous>,<list_index>	Not SCPI Approved
:STARt	<**AUTO**matic|TRIGgered>,,<list_index>	Not SCPI Approved
:SWEep		Confirmed
:COUNt	<numeric_value> (1, **1**, 1e6)	Confirmed
:DIRection	<**UP**|DOWN>	Confirmed

Table C-1. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:SPACing	<**LIN**ear|LOGarithmic>	Confirmed
:TIME	<numeric_value> (30e-3, **1**, 1e3)	Confirmed
:MODE	<CRESet|TRESet|HRESet|CREVerse|TREVerse|HREVerse|MANual>	Confirmed[1]
:VOLTage		Confirmed
[:LEVel]		Confirmed
[:IMMediate]		Confirmed
[:AMPLitude]	<numeric_value> (0, **1**, 5.5)	Confirmed
:OFFSet	<numeric_value> (-5.5, **0**, 5.5)	Confirmed
STATus		Confirmed[3]
:OPERation		Confirmed[3]
:CONDition		Confirmed[3]
:ENABle		Confirmed[3]
:ENABle?	<NRf>	Confirmed[3]
[:EVENt]		Confirmed[3]
:PRESet		Confirmed[3]
:QUEStionable		Confirmed[3]
:CONDition		Confirmed[3]
:ENABle		Confirmed[3]
:ENABle?	<NRf>	Confirmed[3]
[:EVENt]		Confirmed[3]
SYSTem		Confirmed
:DATE	<year>,<month>,<day>	Confirmed
:ERRor?		Confirmed
:TIME	<hour>,<minute>,<second>	Confirmed
:VERSion?		Confirmed
TEST		Confirmed
[:ALL]?		Confirmed[1]
:RAM?		Confirmed[1]
TRACe		Confirmed
:CATalog?	<trace_name>,<trace_name>,. . .	Confirmed
:DATA	<trace_name>,<block>|<trace_name>	Confirmed
:LINE	<trace_name>,<point_index1>,<point_value1>,<point_index2>,<point_value2>	Confirmed
:POINt	<trace_name>,<point_index>,<point_value>	Not SCPI Approved
:DEFine	<trace_name>,<numeric_value>|<trace_name>	Confirmed
:DELete		Confirmed
[:NAME]	<trace_name>	Confirmed
:ALL		Confirmed
:DIRectory?	<name>,<size>,<limit>,<limit>. . .	Not SCPI Approved
:FREE?	<numeric_value>,<numeric_value>	Confirmed
:LIMits	<trace_name>,<start_index>,<stop_index>	Not SCPI Approved
:MODE	<**CW**|RASTer>	Not SCPI Approved
:POINts	<trace_name>,<numeric_value>	Confirmed

Table C-1. Model 1385 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
TRIGger		Confirmed
:COUNt	<numeric_value> (1, **1**, 1048575)	Confirmed
:GATE		Not SCPI Approved
[:STATe]	<Boolean_data> (**OFF**)	Not SCPI Approved
[:IMMediate]		Confirmed
:POLarity	<**POS**itive|NEGative>	Not SCPI Approved
:SOURce		Confirmed[1]
:ADVance	<**INT**ernal | EXTernal | CHAin | ECHain>	Not SCPI Approved
[:STARt]	<**INT**ernal|EXTernal|CHAin|ECHain|TTLTrg<n>>	Not SCPI Approved
:TIMer	<numeric_value> (2e-7, **1e-3**, 1e4)	Confirmed

Notes:

1. Device dependent parameter character data.
2. Incomplete implementation; at least one parameter not supported per SCPI specification.
3. STATus Subsystem commands operate per the specification, but the physical Status Registers are not implemented in the hardware.

D.1 Command Trees

SCPI Command Trees provide a visual alternative to Command Tables as an aid in building syntactically correct commands to operate the Model 1385. The following figures, Figure D-1 through D-8, illustrate the complete command set. Figure D-1 below shows the ROOT NODE and the various Subsystems. This figure is to be used to take the programmer/operator to the other figures, which generally illustrate one Subsystem at a time.

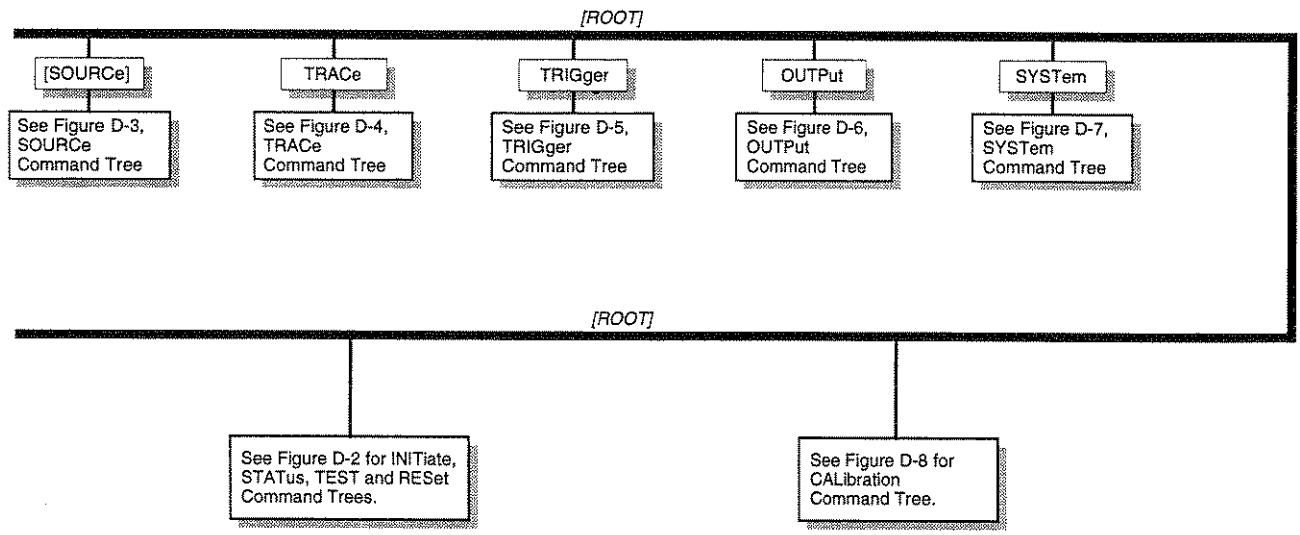


Figure D-1. Subsystems (Root Node).

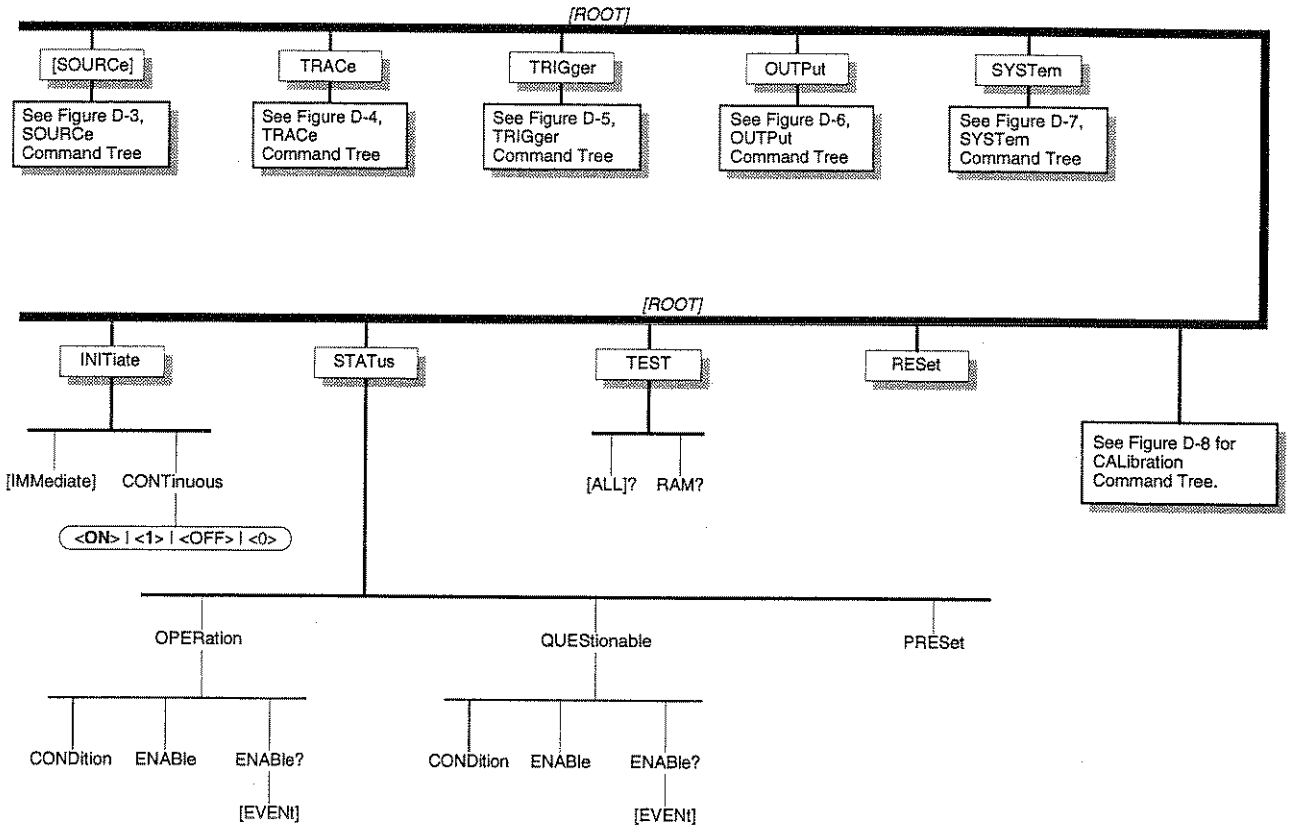


Figure D-2. INITiate, STATus, TEST, and RESet Subsystems.

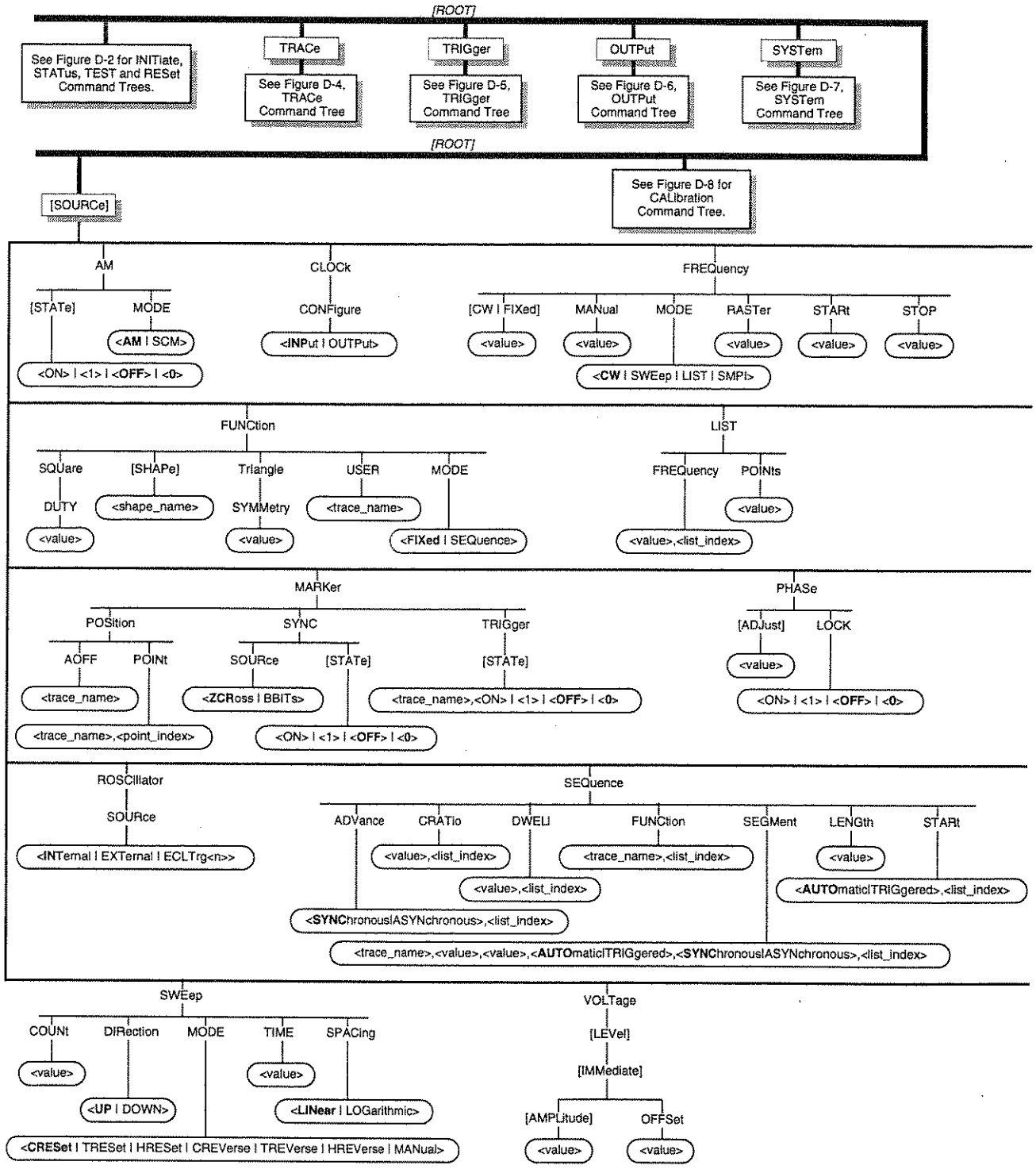


Figure D-3. SOURce Subsystem.

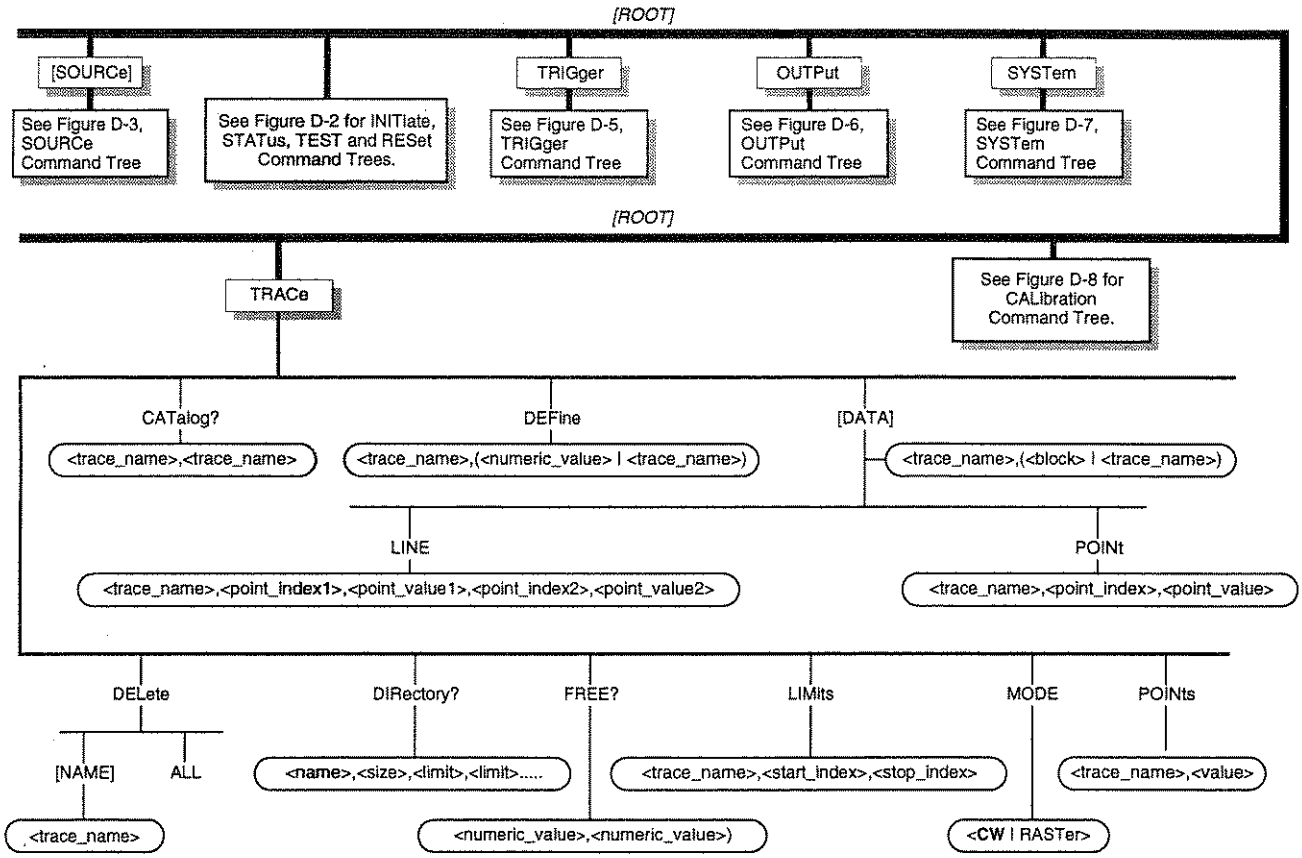


Figure D-4. TRACe Subsystem.

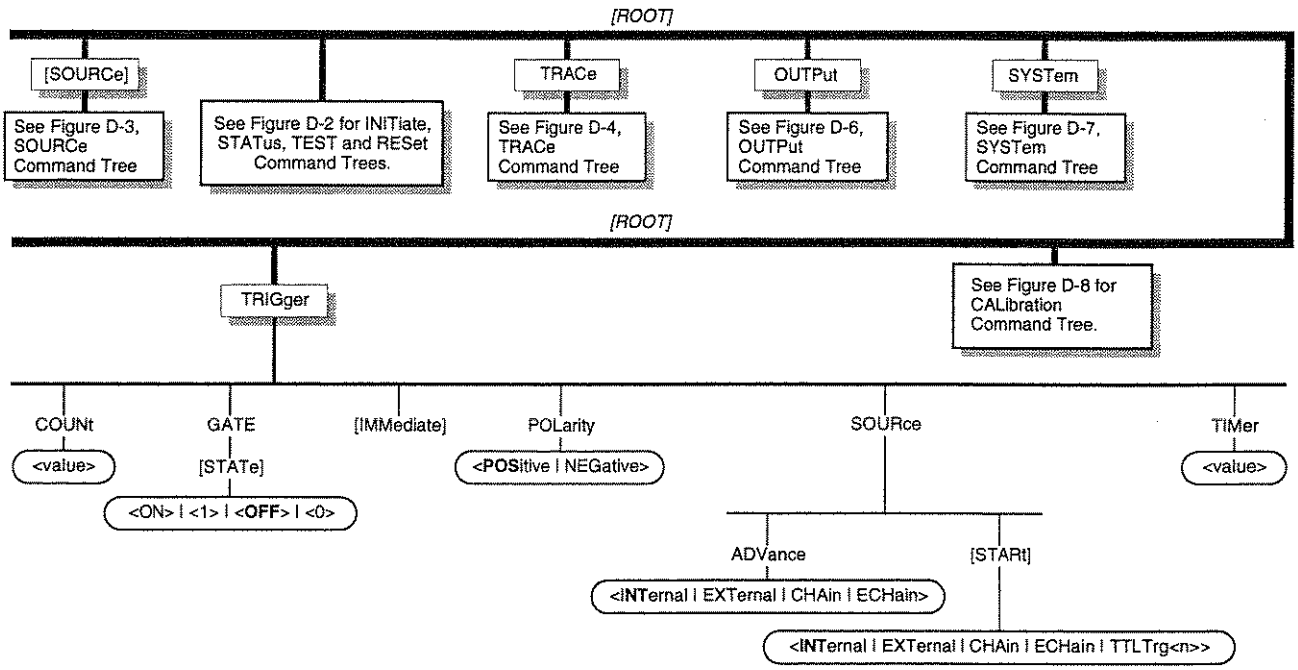


Figure D-5. TRIGger Subsystem.

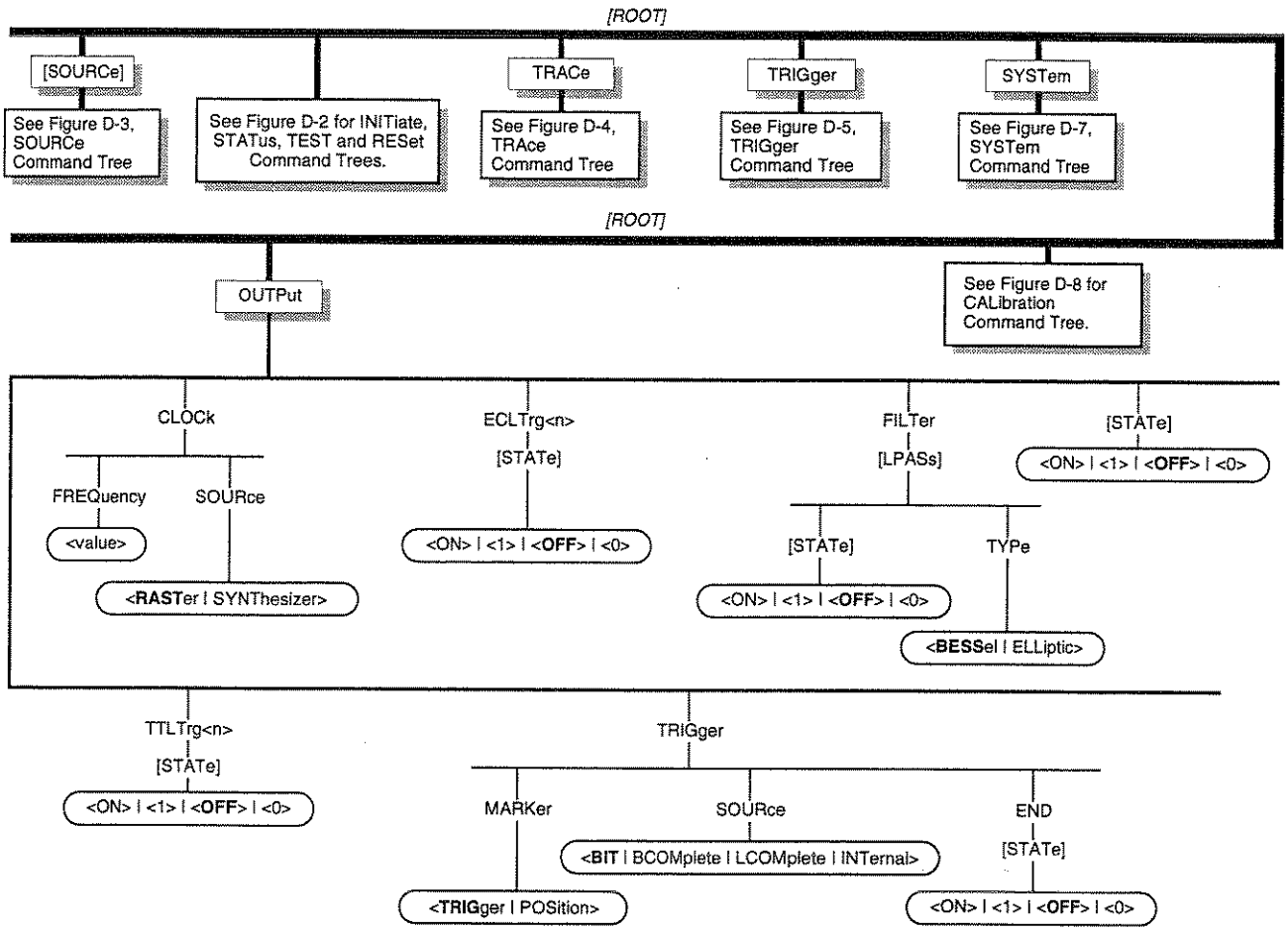


Figure D-6. OUTPut Subsystem.

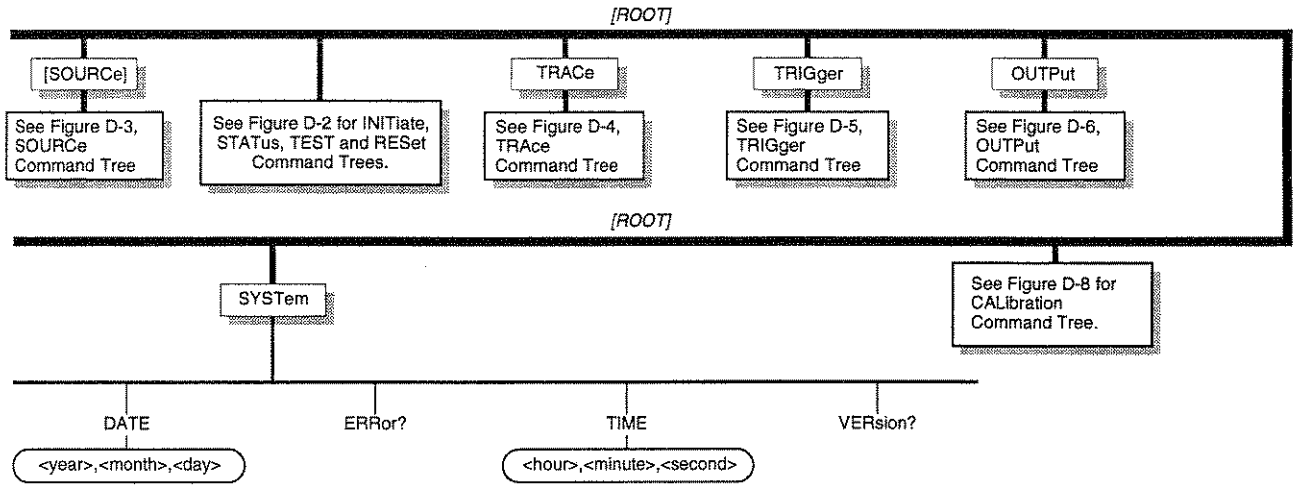


Figure D-7. SYSTem Subsystem.

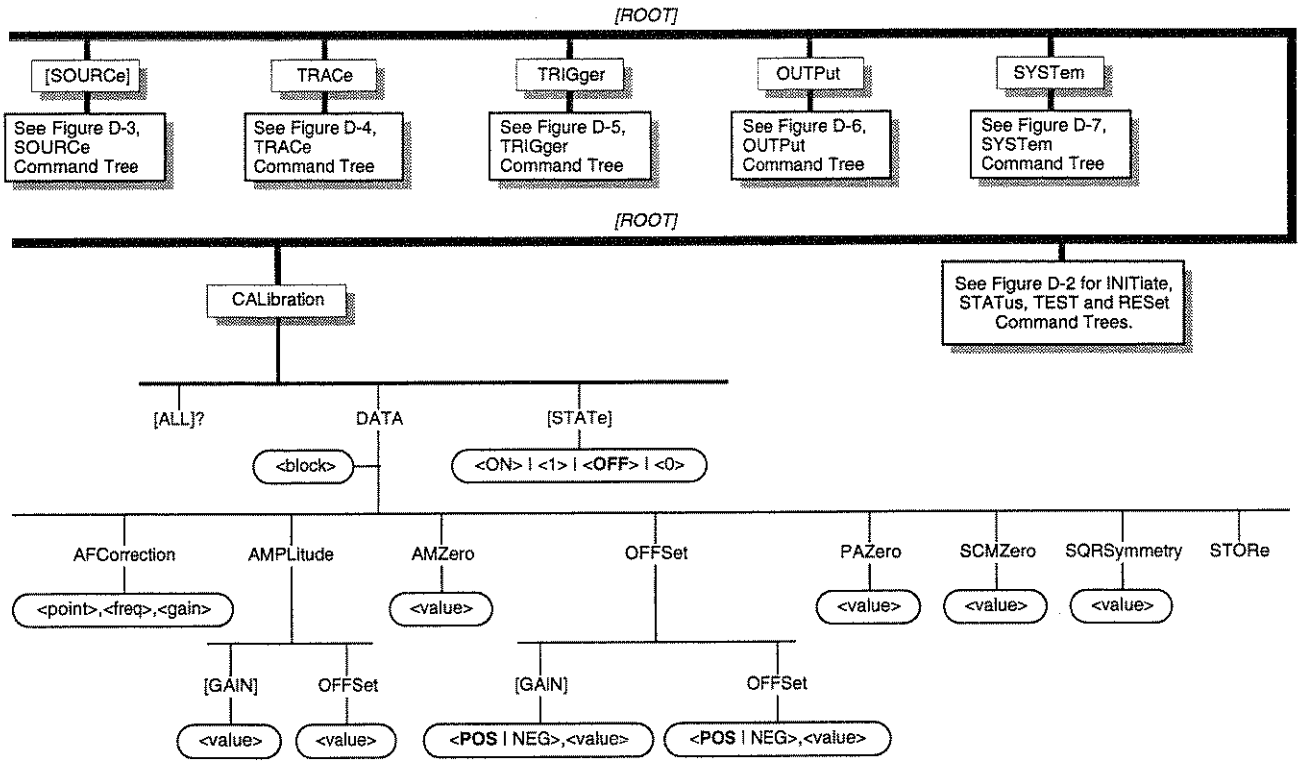


Figure D-8. CALibration Subsystem.

E.1 Introduction

This appendix provides the Operator/Programmer with programming examples to create arbitrary waveforms on the Model 1385.

The following four programs use Wavetek's WaveTest™ Basic language running on an external PC GPIB controller. a GPIB to VXIbus Interface card is used in the VXIbus chassis. Any programming language that is acceptable to the user's controller will work. The first two programs use the factory pre-defined waveforms to supply data to the trace. Complete SCPI keywords are used, with the approved abbreviation in capitals.

The fifth example in this appendix illustrates a "C" program running on a VXIbus embedded controller.

The last example just quotes the SCPI commands which, when sent to the 1385, will define and output a four-waveform sequence

E.1.1 Example 1

SQUARE WAVE; 512 Points, 100 KHz, 10 Vpp, Continuous

```
10   ! This program creates a 512
     point square wave, WAVE1;
     100 kHz; 10 Vp-p.
20   Print @ 10102:"TRACe:DEFine
     WAVE1,512"
30   Print @ 10102:"TRACe:DATA
     WAVE1,SQUare"
40   Print @
     10102:"SOURce:FUNCtion:
     USER WAVE1;:SOURce:FUNCtion
     USER"
50   Print @ 10102:"SOURce:
     FREQuency:CW 100E3"
60   Print @ 10102:"SOURce:
     VOLTage:LEVel: IMMediate:
     AMPLitude 5"
70   Print @ 10102:"OUTPut:STATe ON"
80   End
```

The number "10102" is the complete GPIB address of the Model 1385 in the VXIbus chassis, when addressed from the PC. The first number is the GPIB card, the next two digits is the GPIB primary address (the GPIB/VXIbus Interface card), and the last two numbers are the secondary address (the Model 1385 module). In this case, the address "10102" is the instrument at primary address "01" and secondary address "02" on port "1".

In line 20 the trace name and size is defined as WAVE1 and 512 points using the Trace Subsystem.

Line 30 selects the data source for WAVE1 as the predefined shape SQUare, using the Trace Subsystem. The optional DATA command is used in this line for documentation clarity (the command would have set the factory predefined shape if sent as "`TRACe WAVE1, SQUare`").

In line 40, using the Source Subsystem, trace WAVE1 is selected for output.

Line 50 uses the optional SOURce and CW commands for documentation clarity. They were not needed for the command to function properly ("`FREQuency 100E3`" would have set the frequency).

Line 60 shows the optional SOURce, LEVel, IMMediate, and AMPlitude commands for documentation clarity ("`VOLTage 5`" would have set the amplitude). Unless another subsystem is selected as the first SCPI keyword in the command line, the default is the Source subsystem. The FREQuency sub-command is assumed by the Arb to be in the Source subsystem as no other subsystem was entered on the command line. Line 50 sets the frequency to 100 kHz, line 60 sets the amplitude to 5V peak.

Line 70 turns the output relay ON. The STATe keyword in the Output subsystem is optional and shown for documentation clarity ("`OUTPut ON`" or "`OUTPut 1`" would have turned the output on).

E.1.2 Example 2

SINEWAVE; 200 points, 20 kHz, 2Vpp Internal Trigger 1kHz

```
10  !This creates WAVE2, a 200
    point, 20 kHz sine at 2Vp-p;
    triggered at 1kHz.
20  Print @10102:"TRACe:DEFine
    WAVE2,200"
30  Print @10102:"TRACe:DATA
    WAVE2,SINusoid"
40  Print @10102:"FUNCtion:USER
    WAVE2;:FUNCtion USER"
50  Print @10102:"FREQuency:CW
    2E4"
60  Print @10102:"VOLTage 1"
70  Print @10102:"TRIGger:TIMer
    1E-3"
80  Print @10102:"TRIGger:SOURce
    INTernal"
90  Print @10102:"INITiate:
    CONTinuous OFF"
100 Print @10102:"OUTPut ON"
110 End
```

Lines 20 through 60, and 100 are similar to the square wave in the previous example.

Line 70 sets the trigger frequency to 1 kHz. This can only be done when using the internal trigger.

Line 80 selects the Start Trigger source as internal. This is a default value, but is shown for documentation.

Line 90 selects the Arb triggered operating mode.

E.1.3 Example 3

ARBITRARY WAVEFORM, sine wave 8 points,

This sine wave data is transfered using the arbitrary length block data format. The frequency and amplitude are set at the default values (1 kHz and 1 Vpk).

Lines 10 through 100 fill an array of point values:

```
10  !Enter the number of data
    points.
20  Data_points = 8
30  !Fill the array
40  Step_val = 2*Pi/Data_points
50  X = 0
60  FOR I = 1 TO Data_points
70  Arb_data = (SIN(X)+1)*4095/2
80  X = X+Step_val
90  Waveform_data[I] = Arb_data
100 NEXT I
```

Lines 110 through 260 create the waveform data:

```
110 Data_points = Data_points*2
120 Data_point$ =
    STR$(Data_points)
130 Length = LEN(Data_point$)
140 Length$ = STR$(length)
150 Data_block$ = " "
160 Header_block$ = "#"
170 Header_block$ = Header_block$
    & Length$
180 Header_block$ = Header_block$
    & Data_point$
190 FOR I = 1 TO Data_points/2
200 High_byte =
    Waveform_data[I]\256
210 Low_byte =
    MOD(Waveform_data[I],256)
220 High_byte$ = CHR$(High_byte)
230 Low_byte$ = CHR$(Low_byte)
240 Data_block$ = Data_block$ &
    High_byte$
250 Data_block$ = Data_block$ &
    Low_byte
260 NEXT I
```

Lines 270 through 290 are the output to the Arb. The new trace is named WAVE3:

```
270 Print@10102:"OUTPut ON"
280 Print@10102:"TRACe:DEFine
    WAVE3,8"
290 Print@10102:"TRACe:DATA
    WAVE3,";Header_block$;Data_block$
300 End
```

Lines 240 and 250 assemble the high byte/low byte in the correct order.

Line 280 allocates eight points in memory to be reserved for trace WAVE3.

The "Header_block$" in line 290 creates the "#216" required for the 16 data bytes required for an eight point trace. "Data_block$" is the 16 bytes, in high byte/low byte order, needed to create the eight point sine wave trace (refer to paragraph 3.3.3.11 for Definite Length Arbitrary Block format).

Lines 10 through 100 can be replaced with an algorithm selected by the user to create a different waveform or change the number of points, etc.

E.1.4 Example 4

ARBITRARY WAVEFORM, HEARTBEAT, 32 POINTS

```
10  ! Defines the Model 1385
    address
20  Vxi_address = 901
40  ! Enter the number of data
    points
50  Data_points = 32
70  ! Fill the waveform_data array
80  ! Wave form _data defined as
    an array of 32 elements
100 Label 1:
110 DATA
    2048,2048,2048,2048,2048,2300,2200,1800
120 DATA
    1500,2300,3100,3900,3400,2900,2400,1900
130 DATA
    2048,2048,2048,2048,2048,2048,2048,2048
140 DATA
    2048,2048,2048,2048,2048,2048,2048,2048
150 RESTORE Label1
160 READ Waveform_data
180 ! Sets up the header string
    for arbitrary waveform
    download
200 Data_points = Data_points*2
210 Data_points$ =
    STR$(Data_points)
220 Length = LEN(Data_points$)
230 Length$ = STR$(Length)
240 Data_block$ = ""
250 Header_block$ = "#"
260 Header_block$ = Header_block$
    & Length$
270 Header_block$ = Header_block$
    & Data_points$
290 ! Splits up the waveform_data
    value into high byte
300 ! and low byte and converts to
    character representation
320 FOR I = 1 TO Data_points/2
330 High_byte = Waveform_data[I]/
    256
340 Low_byte =
    MOD(Waveform_data[I],256)
350 High_byte$ = CHR$(High_byte)
360 Low_byte$ = CHR$(Low_byte)
370 Data_block$ = Data_block$ &
    High_byte$
380 Data_block$ = Data_block$ &
    Low_byte$
390 NEXT I
410 ! Sets up the Model 1385 and
    downloads the string to the
    ARB
430 PRINT @Vxi_address:"*rst"
440 PRINT
    @Vxi_address:"trace:del:all"
450 PRINT @Vxi_address:"volt 5"
460 PRINT @Vxi_address:"outp:filt
    on"
470 PRINT @Vxi_address:"trace:mode
    raster"
480 PRINT @Vxi_address:"trace:def
    heartbeat,32"
490 PRINT
    @Vxi_address:":trace:data
    heartbeat,";Header_block$;Data_block$
500 PRINT @Vxi_address:"outp on"
```

E.1.5 Example 5

SHARED MEMORY DOWNLOAD in 'C'

This example uses a RadiSys EPC2 slot 0 controller.

```
#define EPC2

#define NO_ERROR    0
#define TIMEOUT     2000


#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>
#include <string.h>
#include <sys\types.h>
#include <sys\timeb.h>
#include <c:\epconnec\include\busmgr.h>
#include <c:\epconnec\include\epc_err.h>
#include <c:\epconnec\include\mds.h>
#include <c:\epconnec\include\olrm.h>


/*************************************************************************

    int smem_download(DEVICE *devices);

    The DEVICE structure contains the logical address and offset register for   a
VXIbus device.  devices[] is an array of DEVICE structures.  Each Model 1385 found
by the Resource Manager during system power on is added to the devices[] array.

*************************************************************************/
smem_download(devices)
DEVICE  *devices;
{
int     error;

longi,                      /* Generic loop counter                      */
        size,               /* Size of trace in points                   */
        bytes;              /* Size of trace in bytes                    */

unsigned char   ulas,       /* Unit's Logical Address                    */
                buffer[100];    /* Message buffer                        */

unsigned long   offset,     /* Base address of A24 Shared Memory         */
                timeout;        /* Timeout for EPC I/O functions         */

unsigned short  *data;      /* Pointer to dynamically allocated memory*/

    timeout = TIMEOUT;      /* Initialize timeout to 2 seconds           */
```

```
    ulas = devices[0].las; /* Logical address of first 1385          */
    offset=devices[0].offset;  /* Shared Memory offset of first 1385 */
    offset <<= 8;          /* Convert offset to A24 address          */


/*
*   Prompt user for trace size
*/

    printf("Enter trace size > ");
    gets(buffer);

    if(sscanf(buffer, "%ld", &size) != 1)
        {
        printf("Error. Invalid integer");
        return(-111);
        }
    else
    if((size < 5) || (size > 32700))
        {
        printf("Error. Invalid trace size");
        return(-111);
        }

/*
* Set EPC access mode for Motorola byte order and A24 non-supervisory
data
*/
    if(error = EpcSetAccMode(BM_MBO | A24ND))
        return(error);

/*
* Reset the Model 1385, delete all traces and turn the output on
*/
    strcpy(buffer, "*rst;:trace:del:all;:outp on");
    if(error = EpcWsSndStr(ulas, buffer, strlen(buffer), NULL, timeout))
        return(error);

/*
* Define a 'size' point trace named 'test'. Initialize it with a square wave
* pattern and select it for output.
*/
    sprintf(buffer, "trace:def test,%d;data test,squ;sel test", size);
    if(error = EpcWsSndStr(ulas, buffer, strlen(buffer), NULL, timeout))
        return(error);

    bytes = size << 1;      /* Calculate number of bytes in trace */

    if(!(data = (unsigned short *)malloc(bytes))
        {
        printf("malloc error\n");
        return(NO_ERROR);
        }
```

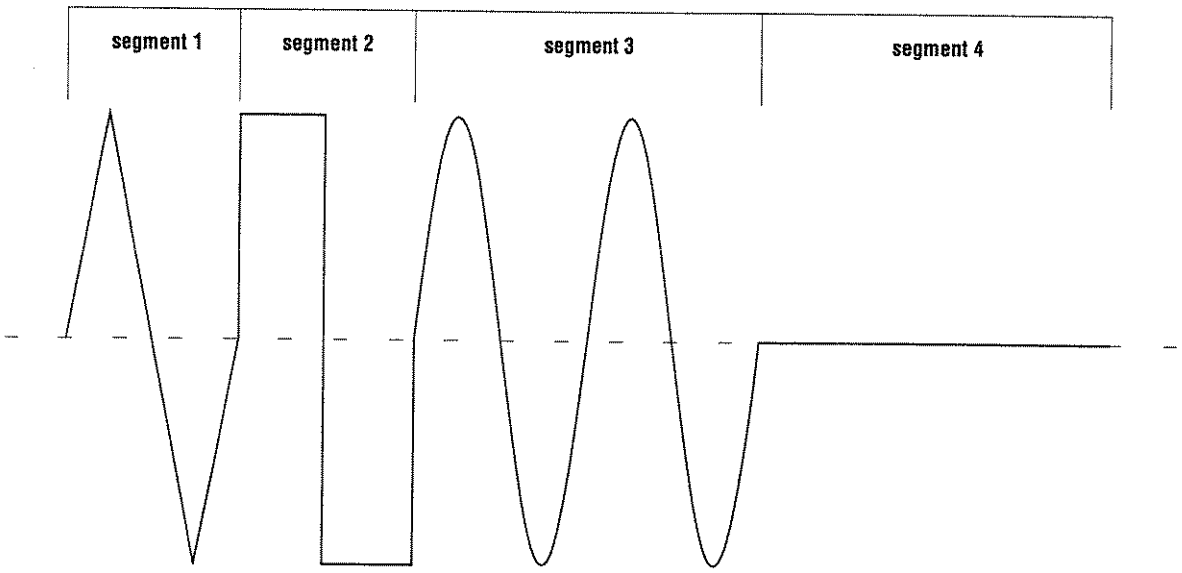
E.1.5 Example 5 *(Contd.)*

```
    for (i=0; i<size; i++)
    data[i] = (i * 4095)/(size - 1); /* Create a ramp pattern      */

    error = EpcToVme(BM_W16 | BM_FASTCOPY, (char far *)data, offset, bytes);

    free(data);

    if(error)
        return(error);

/*
* Initialize 'test' with data contained in Shared Memory.  Turn the Marker
* output on and turn on the position marker for point 0 of 'test'.  Trigger
* scope on rising edge of Position Marker output.
*/
    strcpy(buffer, "trac:data test,smem;:mark on;:mark:pos:poin test,0");
    if(error = EpcWsSndStr(ulas, buffer, strlen(buffer), NULL, timeout))
        return(error);

    return(NO_ERROR);
}
```

E.1.6 Example 6

Sequence Example

Creating a 4-segment sequence.



First create the required waveforms.

TRACE:DEFINE DAN,1000	Define a 1000 point waveform named "DAN"
TRACE:DEFINE MIKE,1000	Define a 1000 point waveform named "MIKE"
TRACE:DEFINE JOHN,1000	Define a 1000 point waveform named "JOHN"
TRACE:DEFINE BRUCE,10	Define a 1000 point waveform named "BRUCE"

Then define the waveforms

TRACE:DATA DAN,TRI	Make the waveform "DAN" be a Triangle
TRACE:DATA MIKE,SQUARE	Make the waveform "MIKE" be a Square
TRACE:DATA JOHN,SIN	Make the waveform "JOHN" be a Sin
TRACE:DATA BRUCE,DC	Make the waveform "BRUCE" be a DC

Next setup the sequence

The format of the Segment command is <Trace Name>,<Repeat Count>,<Clock Divider Ratio>
<Start Condition>,<End Condition>,<Segment Index>

SEQ:SEGM DAN,1,1,AUTO,SYNC,0	
SEQ:SEGM MIKE,1,1,AUTO,SYNC,1	
SEQ:SEGM JOHN,2,1,AUTO,SYNC,2	See Note 1
SEQ:SEGM BRUCE,1,100,AUTO,SYNC,3	See Note 2

Lastly output the sequence

OUTPUT ON	Turn the Main output on
SEQ:LENG 4	Set the number of segments to 4
FUNC:MODE SEQ	Put the 1385 in Sequence mode

Note 1:

The 2 in the Repeat count causes the sinewave to play 2 times in the 3rd segment

Note 2:

The 100 in the Clock Divider Ratio causes the clock to be divided by 100. This will make the waveform 100 times slower.

WAVETEK TEST & MEASUREMENT DIVISION — SALES AND SERVICE REPRESENTATIVES WORLDWIDE

ARGENTINA
Reycom Electronica SA
B de Irigoyen 972 2DO Piso C,
1304 Buenos Aires
Tel: 54 1 300 2018
Fax: 54 1 300 2010

AUSTRALIA
Scientific Devices Pty. Ltd
PO Box 63, 2 Jacks Road,
South Oakleigh, Victoria 3167
Tel: 61 3 579 3622
Fax: 61 3 579 0971

AUSTRIA
Wavetek Gesellschaft mbH.
Pharos Haus
Nordbahnstrasse 36/TOP 1.4
A-1020 Vienna, Austria
Tel: 43 1 21451 10
Fax: 43 1 21451 09

AUSTRIA
Walter Rekirsch, und Co.
Vertriebs KG, Obachgasse 28,
1220 Wien
Tel: 43 1 259 7270
Telex: -----
Fax: 43 1 259 7275

BANGLADESH
Sisco International Ltd.
13/2 Toyenbee Circular Road,
Mitijheel Commercial Area,
GPO Box 2545
Tel: 880 2 241 491
Telex: 950 632157 MITBJ
Fax: 880 2 833 041

BENELUX
Air Parts International BV
PO Box 255, Kalkovenweg 12,
2400 AG Alphen aan den Rijn
Tel: 31 1 72 422 455
Telex: 844 39564
Fax: 31 1 72 421 022

BOLIVIA
Electronic Marconi SRL
Calle Yanacocha 337,
Cajon Postal 143, La Paz
Tel: 591 2 352 574
Fax: 591 2 314 540

BRAZIL
Sistronics Instrumentacao
E Sistemas LTDA
Av. Adolfo Pinheiro 1000-9 AND
CEP 04734-022, Sao Paulo - SP
Tel: 55 11 247 5588
Fax: 55 11 523 8457

CANADA
Interfax
5947 Chemin St.,
St. Laurent,
Quebec H4S 1B6
Tel: (514) 336-0392
Fax: (514) 336-9607

CHILE
Avantec Ltda
PO Box 1087
Fidel Oteiza 1921 - Of. 1106
Providencia - Santiago
Tel: 56 2 341 1021
Fax: 56 2 341 1020

CHINA
Wavetek Corporation;
Room 2701, Citic Building
No. 19 Jianguomenwai Dajie
Beijing 100004
PRC
Tel: 86 10 6592 8044
Fax: 86 10 6500 8199

DENMARK
Teleinstrument A/S,
Christiansholmsgade
8700 Horsens
Tel: 45 76 251 818
Fax: 45 75 615 658

ECUADOR
Dimatronic SA,
Junin 430 Y Cordova
Guayaquil
Tel: 593 4 301 863
Fax: 593 4 563 664

EASTERN EUROPE
Wavetek GmbH
A-1020 Wien, Pharos Haus,
Nordbahnstraße 36, Top 1.4,
Austria
Tel: 43 1 214 51 10
Fax: 43 1 214 51 09

EGYPT & MIDDLE EAST
Computek Test & Measurement
Instruments Division
23, Amer Street, Dokki,
Giza, Cairo.
Tel: 202 3602234
Fax: 202 349 8683

FINLAND
Finn Metric OY
Scandia Metric Group
PO Box 4, Riihitonuntie 2
FIN-02201 Espoo
Tel: 358 0 423 911
Fax: 358 0 426 967

FRANCE
Wavetek S. A.
34 rue Necker
42000 Saint-Etienne, France
Tel: 33 77 47 89 00
Fax: 33 77 47 89 70

FRANCE
M. B. Electronique
606 Rue Fourny-BP31, Z.I. de Buc,
78533 BUC CEDEX
Tel: 33 1 39 56 81 31
Telex: 842 695414
Fax: 33 1 39 56 53 44

GERMANY
WAVETEK GmbH
Gutenbergstraße 2-4,
85737 Ismaning
Tel: 49-89-99641-0
Fax: 49-89-99641-160

GREECE
American Technical
Enterprises SA
PO Box 3156
Agiou Konstantinou 39,
Athens 10210.
Tel: 30 1 524 0620
30 1 524 0740
Telex: 863 216046 ATE GR
Fax: 30 1 524 9995

GUATEMALA
Ceica, SA
20 Calle 16-03, Zona 10,
01010 Guatemala City
Tel: 502 2 335 682
Tel: 502 2 335 683
Tel: 502 2 335 691
Fax: 502 2 335 690

HONG KONG (1)
Wavetek Hong Kong Ltd.
3A, HKPC Building,
78 Tat Chee Avenue,
Kowloon, Hong Kong
Tel: 852 2788 6221
Fax: 852 2788 6220

HONG KONG (2)
Euro Tech (Far East) Ltd
18/F., Gee Chang Hong Centre,
65 Wong Chuk Hang Road,
Hong Kong
Tel: 852 814 0311
Telex: 780 72449 EFELD HX
Fax: 852 873 5974

ICELAND
Rafhus HF
Fiskislod 94
101 Reykjavik
Tel: 354 1 621 616
Fax: 354 1 627 366

INDIA
Technical Trade Links
Deodahar Centre,
424 Marol Maroshi Road,
Andheri (East), Bombay 400 059
Tel: 91 22 850 2412
91 22 850 2204
Telex: 011 79261 TTLIN
Fax: 91 22 850 6719

INDONESIA
C. V. Schmidt Mitra Indonesia
Delta Bldg., Block A, No. 30,
JL Suryo Pranoto No 1-9,
Jakarta 10160.
Tel: 62 21 380 7845
Fax: 62 21 380 7847

IRELAND
Euro Electronic Systems
Unit 1, Sandyford Park,
Sandyford Industrial Estate
Dublin 18.
Tel: 353 12 952 326
Fax: 353 12 952 246

ISRAEL
DAN-EL Technologies Ltd.
PO Box 13144, Office 60,
Tel-Aviv 661131.
Tel: 972 3 647 8770
Fax: 972 3 647 8771

ITALY
DELO Instruments
Via Piemonte 14
20090 Fizzonasco Pieve E,
Milano
Tel: 39 2 907 22441
Fax: 39 2 907 22742

JAPAN (Calibration Instruments)
Yokogawa Electric Corporation
155 Takamuro-Cho
Kofu-Shi, Yamanashi-Ken 400
Tel: 0552 43 0332
Fax: 0552 43 0399

JAPAN (GPT Instruments)
Toyo Corporation
26-9, Yushima-Cho
Bunkyo-Ku, Tokyo 113
Tel: 81 356 88 6800
Fax: 81 356 88 6900

KENYA
GES Engsales Kenya Ltd.
P O Box 46658,
Waiyaki Way, ABC Place,
Niarobi
Tel: 254 2 441 209
254 2 448 814
Fax: 254 2 448 815

KOREA
Myoung Corporation
25-4 Yeo Eui Do Dong,
Young Deng Po-Ku
PO Box 14,
Young Deng Po-Ku, Seoul
Tel: 82 2 784 9942
Telex: K 24283 MYOUNG
Fax: 82 2 784 2387

KUWAIT
Tareq Company
PO Box 20506 Safat,
13066 Safat
Tel: 965 243 6100
Telex: 2231 KT ZUAITER
Fax: 965 243 7700

LEBANON
Mabek
PO Box 13-5657,
Beirut
Tel: 961 1 864 497
Telex: 22889 LE MABEK
Fax: 961 1 603 129

MALAYSIA
Schmidt Scientific SDN BHD
13th Floor, Wisma Mirama,
Jalan Wisma Putra,
P. O. Box 10592,
50718 Kuala Lumpur.
Tel: 60 3 242 7122
Fax: 60 3 248 5143

MEXICO
Mexitex, S.A.
Porfirio Diaz 53, Col. Del Valle,
APDO, Postal 12-1012
Mexico, D.F. 03100
Tel: 525 575 9929
525 575 0312
525 575 0269
Fax: 525 575 9981

MOROCCO
Minhol SA
64 Rue El Mortada,
Casablanca 02.
Tel: 212 2 25 52 92
212 2 25 55 13
212 2 25 49 69
Telex: 24064M
Fax: 212 2 25 49 92

NEW ZEALAND
Nilsen Technologies (NZ) Ltd.
Unit 1, 59-61 Marsden St.
Lower Hutt.
Tel: 64 4 569 4676
Fax: 64 4 569 4129

NIGERIA
Boladel Doksfad Nigeria Ltd.
8 Eleyele Street,
Eleyele Lay-Out
PO Box 407,
Ile-Ife,
Osun State
Tel: 234 36 232 077

NORWAY
Metric A/S
Scandia Metric Group
Postboks 46, Holmlia
Nordasveien 5
N-1201 Oslo 12.
Tel: 47 22 61 1070
Fax: 47 22 61 7492

PAKISTAN (1)
Mushko Electronics (Pvt) Ltd.
Oosman Chambers,
Abdullah Haroon Road,
Karachi 74400
Tel: 92 21 524 131
Telex: 29094 MUSKO PK
Fax: 92 21 568 2298

PAKISTAN (2)
M/S Intermark (Pvt) Ltd.
Hakimsons Building,
19 West Wharf Road,
PO Box 6159
Karachi 2
Tel: 92 21 201 725
92 21 201 043
92 21 201 045
Telex: 952 823649 YAQINPK
Fax: 92 21 231 0480

PERU
BMP Ingenieros SA
Ave Del Parque Sur No 552,
Lima 41
Tel: 51 14 901 224
Fax: 51 14 757 026

PHILIPPINES
Avesco Marketing Corp.
PO Box 3531,
Manila
Tel: 63 2 912 8881
Telex: 22095 AESPH
Fax: 63 2 912 2999

PORTUGAL
Decada SA
Rua Margarida Palla. 11B
Miraflores, 1495 Alges
Lisboa.
Tel: 351 1 410 3420
Telex: 832 15515 ESPBPC
Fax: 351 1 410 1844

SAUDI ARABIA
Electronic Equipment Marketing Co.
PO Box 3750,
No. 28 Al Baroudi Lane
Thalateen St, Suleimanieh
Riyadh 11481,
Tel: 966 1 477 1650
Telex: 928 401120 ZUHAIRSJ
Fax: 966 1 478 5140

SINGAPORE (1)
Wavetek Asia-Pacific Pte Ltd
51 Goldhill Plaza #14-04/05
Singapore 308900
Tel: 65 356 2522
Fax: 65 356 2523

SINGAPORE (2)
O'Connor's Singapore Pte Ltd
O'Connor's House,
98, Pasir Panjang Road,
Singapore 118516
Tel: 65 473 7944
Telex: RS 21023 OCONSIN
Fax: 65 472 4508

SOUTH AFRICA
Altech Instruments Pty Ltd
PO Box 2097,
Boksburg 1460
Transvaal
Tel: 27 11 914 4525
Fax: 27 11 914 1475

SPAIN
Equipos y Systemas SA
c/o Apolonio Morales 13B,
28036 Madrid
Tel: 34 1 359 0088
Fax: 34 1 359 0298

SWEDEN
Kaliber AB
Maltesholmvägen 136,
Box 4443 S-165 15 Hasselby
Stockholm
Tel: 46 8 380 350
Fax: 4 68 380 320

SWITZERLAND
Computer Controls AG
PO Box 314
Probusweg 2, CH-8057
Zurich
Tel: 41 1 308 66 66
Fax: 41 1 308 66 55

TAIWAN
Quatek Co., Ltd.
3rd Fl., Spring Plaza, 6, Section 3,
Min Chuan E. Road.,
Taiwan, R.O.C.
Tel: 886 2 501 7065
Fax: 886 2 509 5329

THAILAND
Schmidt Scientific (Thailand) Ltd.
69/5 Soi Seuksavittaya
North Sathorn Road,
Bankok 10500.
Tel: 662 236 7784
Fax: 662 237 2627

UNITED ARAB EMIRATES
Al Sanani Trading Establishment
Old Passport Road,
PO Box 47187,
Abu Dhabi
Tel: 971 2 319161
Telex: 958 23966 SANANIEM
Fax: 971 2 31554

UNITED KINGDOM
Wavetek Test & Measurement Division
52 Hurricane Way,
Norwich Airport,
Norwich, Norfolk NR6 6JB,
England
Tel: 44 1603 404 824
Fax: 44 1603 483 670

UNITED STATES OF AMERICA
- **Wavetek Test & Measurement Division** (Service Only)
c/o Wavetek CATV Division
5808 Churchman Bypass,
Indianapolis, IN 46203
Tel: 317 788 9351
Fax: 317 782 4607

- **North American Sales & Service**
9045 Balboa Avenue,
San Diego, CA 92123
Tel: 619 279 2200
Fax: 619 450 0325

VENEZUELA
Onimex C.A.
2 Avenue Entre,
3 & 4 Transversal,
Los Palos Grandes,
Apartado Postal 61421
Caracas 1062
Tel: 58 2 285 8641
Fax: 58 2 285 8417

WEST INDIES
WISCO International Ltd.
690 West 28th Street,
Hilea FL 33010,
USA
Tel: 305 888 1676
305 888 1677
Fax: 305 887 1624

INTERNET WORLDWIDE NET SITE
http://www.wavetek.com

FOR CUSTOMERS IN COUNTRIES NOT LISTED, PLEASE CONTACT WAVETEK TEST AND MEASUREMENT DIVISION IN THE UNITED KINGDOM:

Wavetek Test & Measurement Division
52 Hurricane Way,
Norwich Airport,
Norwich, Norfolk NR6 6JB,
England
Tel: 44 1603 404 824
Fax: 44 1603 483 670

WAVETEK CORPORATION
Corporate Office
11995 El Camino Real,
Ste. 301, San Diego,
CA 92130, U.S.A.2310
Tel: 1 619 793 2300
Fax: 1 619 793 2310