WAVETEK

OPERATION AND
MAINTENANCE MANUAL

Model 1395/1395 Mate
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 Revision C, 6/95
Manual Part Number 1006-00-0699

WARRANTY

Wavetek warrants that all products manufactured by Wavetek conform to published Wavetek specifications and are free from defects in materials and workmanship for a period of one (1) year from the date of delivery when used under normal conditions and within the service conditions for which they were furnished.

The obligation of Wavetek arising from a Warranty claim shall be limited to repairing, or at its option, replacing without charge, any product 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 non-conforming defective parts or components to a condition as warrantied within a reasonable time after receipt thereof, Buyers shall be credited for their value at the original purchase price.

Wavetek must be notified in writing of the defect or nonconformity within the Warranty period and the affected product returned to Wavetek's factory or to an authorized service center within (30) days after discovery of such defect or nonconformity.

For product warranties requiring return to Wavetek, products must be returned to a service facility designated by Wavetek. Buyer shall prepay shipping charges, taxes, duties and insurance for products returned to Wavetek for warranty service. Except for products returned to Buyer from another country, Wavetek shall pay for return of products to Buyer.

Wavetek shall have no responsibility hereunder 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 and not in accordance with instructions furnished by Wavetek.

Exclusion of Other Warranties

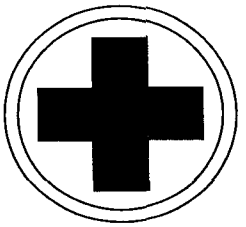
The Warranty described above is Buyer's sole and exclusive remedy and no other warranty, whether written or oral, is expressed or implied. Wavetek specifically disclaims the implied warranties of merchantability and fitness for a particular purpose. No statement, representation, agreement, or understanding, oral or written, made by an agent, distributor, representative, 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 Wavetek employee. Under no circumstances shall Wavetek be liable for any direct, indirect, special, incidental, or consequential damages, expenses, losses or delays (including loss of profits) based on contract, tort, or any other legal theory.

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)

CONTENTS (Continued)

TABLES

ILLUSTRATIONS

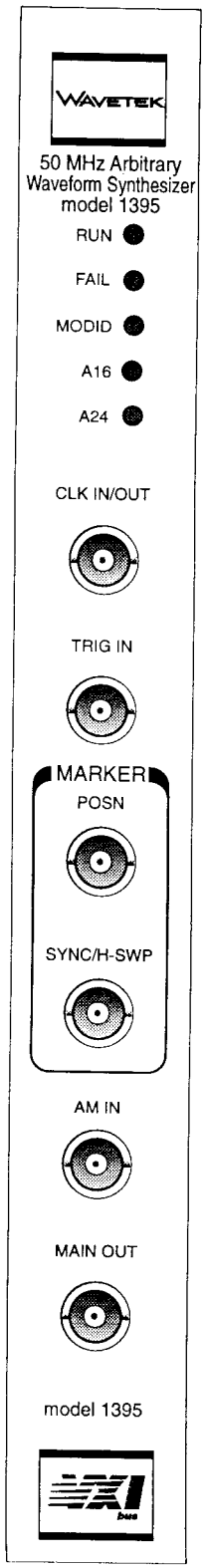


Figure 1-1. Model 1395 50 MHz VXIbus Arbitrary Waveform Synthesizer

Specifications Section 1

1.1 THE MODEL 1395

The MODEL 1395 is a high performance Synthesized Arbitrary Waveform Generator (ARB) with the following main features:

- Up to 50 MHz Sampling Frequency
- 12 Bit Vertical Resolution
- 32K points (128K optional) Horizontal Resolution
- Intermodule Triggering, Summing and Phase Control
- Waveform Linking and Looping
- 64K bytes Shared Memory for fast data transfer
- 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 μHz to 25 MHz; or arbitrary waveforms from 5 points minimum to 32K (128K) 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.

The main waveform output provides up to 15 Vp-p into 50Ω (30 Vp-p, open circuit). Waveform dc offset or dc output is also provided up to ±7.5 V into 50Ω (±15 V into open circuit).

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 1395 waveform synthesizer is a single-slot "C" size VXIbus module. Using any manufacturer's VXIbus chassis, the Model 1395 can be controlled using the SCPI language and the appropriate controller.

Multiple ARBs may be linked and operated together inside one VXIbus chassis. Series operation is provided by full support of the VXIbus SUMBUS protocol. A signal programmed at the output may be sent to the SUMBUS, or signal present at the SUMBUS may be summed into the model 1395 output. 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 1395 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 function or used to name an arbitrary waveform.) One to fifty arbitrary waveforms (traces) may be stored by name in volatile 32,768 point (optionally 131,072 point) RAM memory. Each trace has 12 bits vertical resolution, and from 5 points to the maximum number of points in the waveform memory horizontal resolution.

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, sinc and dc, obvi-

ously, 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 1395 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 10 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**.

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

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:

Linked 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 1395 Arb has four output signals on the front panel: the function output, the position marker, the sync marker, and the sample clock. The Arb also provides a clock to the selected VXIbus backplane ECLTRG line, and a trigger output to the VXIbus Local Bus or to the 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.

SYNC MARKER/H-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 H-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 (±500 mV, open circuit) proportional to sweep position between selected start and stop limits is provided to drive the horizontal axis of a 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 or segment within a SEQuence, 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 or 0.1 mHz 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.

SUMBUS OUTPUT (to VXI Backplane):
Analog signals at the 1395's MAIN OUT may also be summed into the VXIbus SUMBUS line with a fixed scale factor (see Intermodule Analog Summing, paragraph 1.2.14). A full amplitude 15 Vpp signal at the MAIN OUT results in a 75 mApp signal driving the 25Ω SUMBUS line. SUMBUS driver specifications are:

Scale Factor:	5 mA/V (5 mApp signal at the SUMBUS line for each Vpp MAIN OUT).
Accuracy:	±(6% + 2.5mA)
Load Impedance:	25Ω ±2% (VXIbus specification)
Output Impedance:	> 10 kΩ in parallel with < 20 pF
Compliance:	±1.2 V minimum
Bandwidth:	> 50 MHz (limited by the backplane)

1.2.4.2 Inputs

The Model 1395 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:	±15 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 1395. The TTL Trigger line has a VXI specification limit of 12.5 MHz maximum and 30 ns minimum pulse width. Additionally, the 1395 module has a practical limit of 5 MHz maximum for a trigger input signal.

If another 1395 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 1395. 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.

SUMBUS INPUT (from VXI Backplane):
Analog inputs on the VXIbus SUMBUS line may be summed into a model 1395 MAIN OUT with a selection of scale factors (see Intermodule Analog Summing, paragraph 1.2.14). With no SUMBUS attenuation selected, a 1.875 Vpp (75 mA driving 25Ω) signal on the SUMBUS line will drive the MAIN out to its full-scale amplitude of 15 Vpp. SUMBUS receiver specifications are:

Scale Factor (1:1 atten):	8 V/V (8 Vpp out at MAIN OUT for each Vpp input at the SUMBUS).
Accuracy:	±(6% + 200mV + 2.5mA)
Input Impedance:	> 10 kΩ in parallel with < 20 pF
Bandwidth:	> 50 MHz

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. If 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, ≤ 15 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 - 1μHz to 20 MHz.

Square - 1μHz to 25 MHz.

Haversines - 1μHz to 20 MHz.

Other Standard Waveforms - 1μHz to 2 MHz.

Resolution – 8 digits limited by 1 μHz, 5 digits when >20 MHz; 5 digits when in Triggered or Gated modes, or when the selected function is USER vs. a Standard function.

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.

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 32K (128K 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 15Vp-p into 50Ω
	0.03 to 30Vp-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
> 2.500 & ≤ 7.500 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:	± 7.5Vdc into 50Ω
	± 15Vdc 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

Number of Waveform Segments: 2 to 4

Segment Loop Count	1 to 65,535 or continuous.
Start Conditions:	Requires a trigger to Start a SEQuence. Uses the Word Serial command or any selected start trigger event.
Advance Conditions:	Segment Loop Count complete; Loop continuously until selected advance trigger event true; Loop done and advance trigger true.
Advance Trigger Types:	Event - Trigger must transition to the true state to qualify as an event. Trigger event is latched. Level - Trigger must be in the true state to initiate an advance. Trigger is not 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 524,287).

Notes

If advance condition from last segment to first segment is "advance trigger true" or "Loop done and advance trigger true", the sequence must be run in continuous mode.

The trace limits of each trace taken from each block in the sequence are determined by the trace selected by the TRACe:SELect command previous to selecting the SEQuence Mode.

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.

Linked Sequence Advance Condition:

derived from:	Trigger Count Complete. Loop Complete from any or all segments of a linked sequence. Waveform Complete from an arbitrary waveform or any or all segments of a linked sequence.

Trigger Destinations:

Start Trigger:	Initiates gated or trigger modes and starts sequences.
Advance Trigger:	Conditions advances between segments of a linked 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 524,287

Note

Triggered modes of operation are limited to 10 MHz waveform frequency with 5 digits of frequency resolution.

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
> 2.500 & ≤ 7.500 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 Analog Summing:

The waveform from the 1395 module can be driven onto the VXIbus Backplane SUMBUS. The 1395 can also receive the VXI backplane SUMBUS signal, and sum it with the MAIN OUT output signal. To extend the dynamic range of the SUMBUS signal, the 1395 provides eight input attenuators selectable from the following:

Attenuation, dB:	Division, ratio:
0	1/1
-6	1/2
-12	1/4
-18	1/8
-24	1/16
-30	1/32
-36	1/64
-42	1/128

Note

For SUMBUS Driver/Receiver specifications (Scale Factor, bandwidth, etc.) refer to paragraph 1.2.4.1, SUMBUS Output (driver) and Paragraph 1.2.4.2, SUMBUS Input (receiver).

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 1395'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

Expanded Waveform RAM

Quadruples waveform data storage volatile RAM from 32 K to 128 K points.

1.2.17 AutoCal/Diagnostics

Each 1395 ARB Module contains time and 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.

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 1395 Arb 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 1395 Arbitrary Waveform Generator is a single slot, C size VXIbus module. Using any manufacturer's VXIbus chassis, the Model 1395 Arb 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 1395 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 (8 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
+24 Vdc	250 mA	200 mA
+5 Vdc	2000 mA	100 mA
–2 Vdc	250 mA	20 mA
–5.2 Vdc	2200 mA	100 mA
–24 Vdc	250 mA	200 mA
+12 Vdc	200 mA	50 mA
-12 Vdc	350 mA	50 mA

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.

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. **Call Wavetek.** Call Wavetek's Customer Service representative (619 279-2200) to inform them that the shipment arrived damaged. Please be prepared to provide a detailed damage report.

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'll have to pay to repair any shipping damage as carriers won't pay claims on incorrectly packed equipment.
2. **Call Wavetek Customer Service and ask for a return authorization.** The Wavetek Customer Service representative (619 279-2200) will ask for your name, telephone number, company name, equipment type, model number, serial number, and a description of the problem.
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 1395 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 1395 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 1395 should be stored in a clean, dry environment. In high humidity environments, protect the Model 1395 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 1395's logical address, use the eight position DIP switch (figure 2-1) accessible from the side panel. The Model 1395 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. Insert A in figure 2-1 illustrates a switch set to a logical address of 3.

Wavetek ships the Model 1395 with a logical address of 255 for Dynamic Configuration. Refer to insert B in figure 2-1.

2.3.2 Data Transfer Bus Arbitration

The Model 1395 has VMEbus Mastership capability. This means the Sweep/Function Generator, when enabled, sends Responses and Events as signals to its Commander. The Model 1395 cannot drive the interrupt lines.

The Model 1395 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*). 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

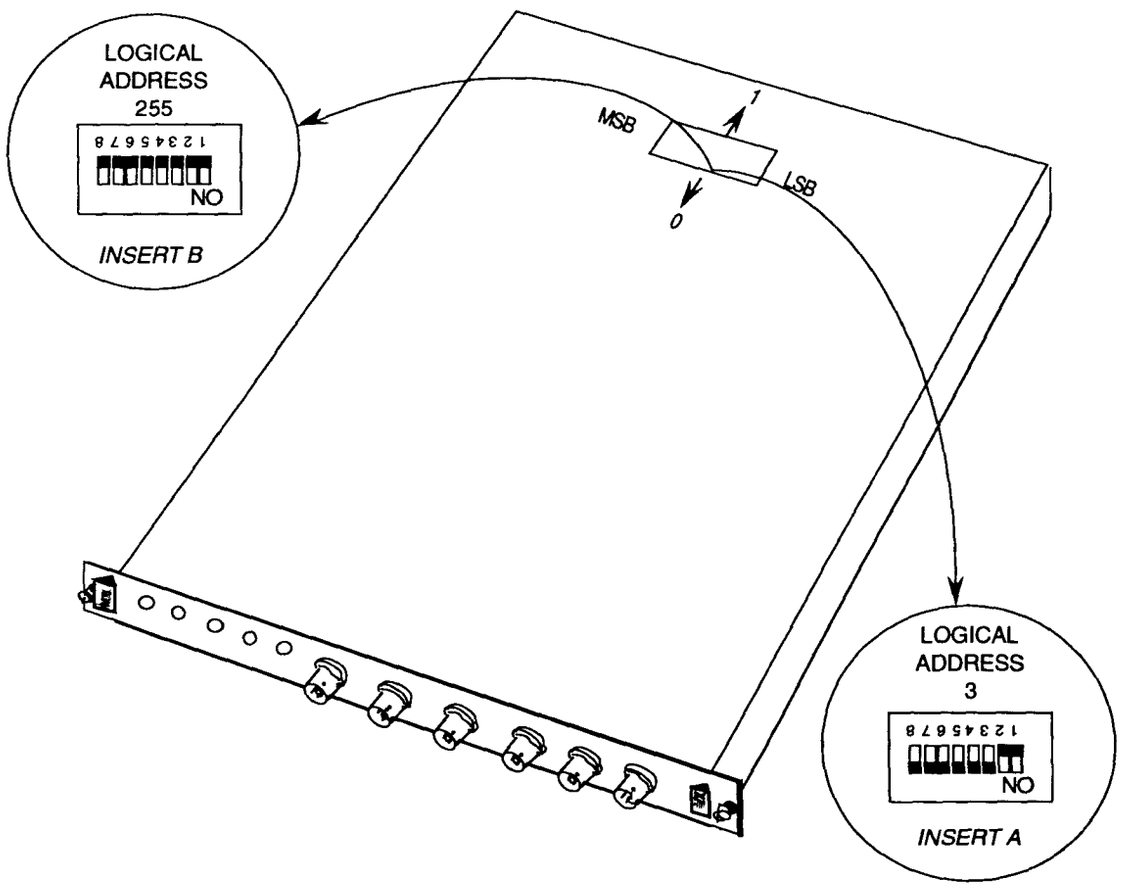


Figure 2-1. Set the Logical Address

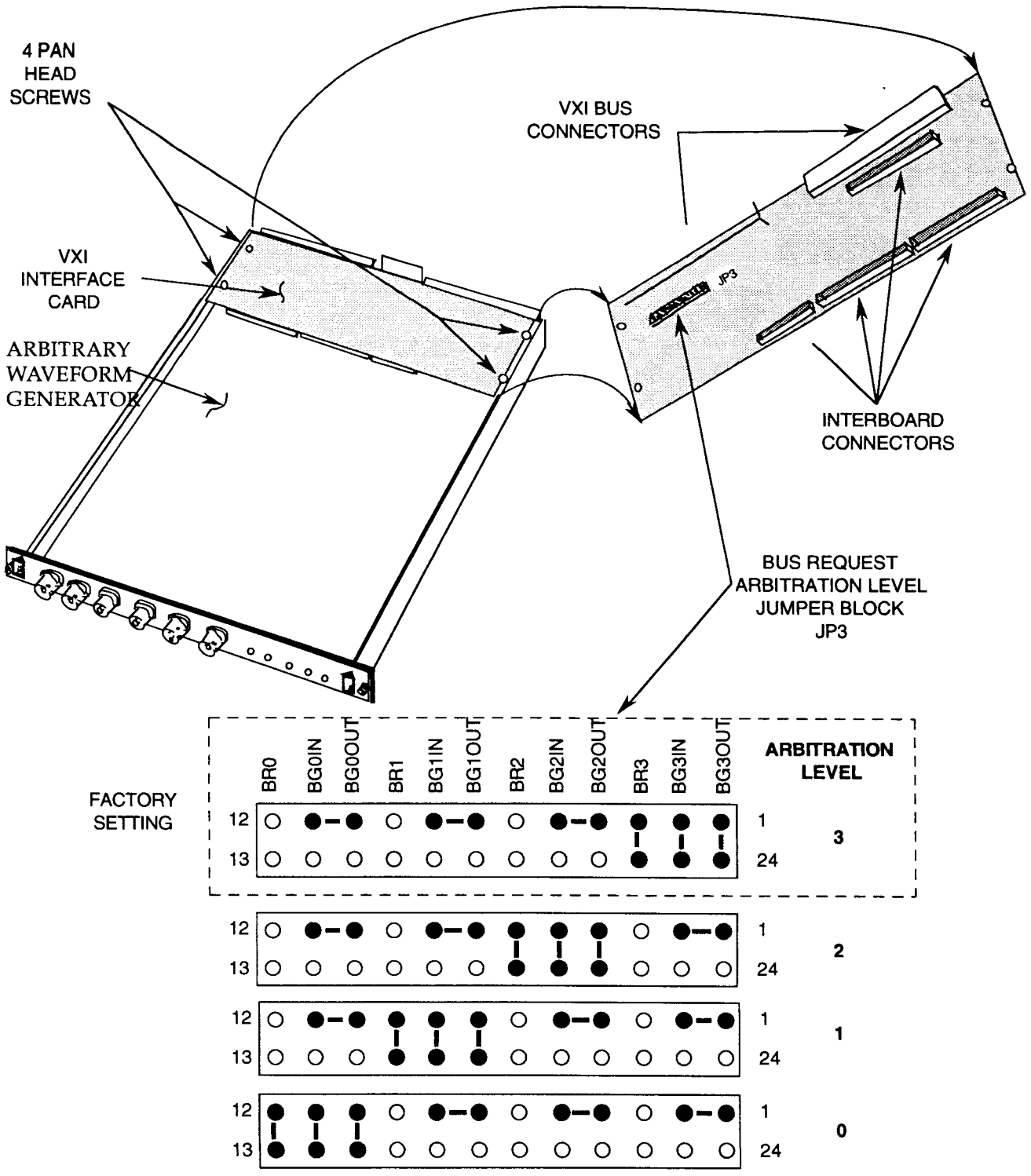


Figure 2-2. Bus Arbitration Level Jumpers

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 1395 to a new level. Refer to the VMEbus specification for more information on *'data transfer bus arbitration'*.

CAUTION

The Sweep/Function Generator 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 1395 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, receiving inspector, etc. with a quick method of verifying the functional operation of the Model 1395. This procedure does not test the unit's specifications. This procedure assumes the Model 1395 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: 1 kHz to 5 MHz Output: TTL
BNC 50Ω Feedthrough (2 ea.)	Accuracy: 0.5% Power: 2W
BNC Coax Cable (2 ea.)	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

LED	Normal Result
Run	On
Fail	On, then Off after a second
MODID	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, call Wavetek Customer Service at 619/279-2200 or 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, 1 kHz 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, 1 kHz square wave from the MARKER SYNC connector.

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

   Verify sets of TTL level pulses at 1 ms intervals.

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

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

   Verify 1Vp, 1 MHz triangle wave.

7) Move cable from 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 1395 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 1395 connectors and LED indicators. Paragraph 3.3 defines the Model 1395 programming messages. Paragraph 3.4 demonstrates how to operate the Model 1395 using the defined messages.

3.2 Connectors and LED Indicators

This paragraph describes the Model 1395 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.

3.3 Model 1395 Programming

The Model 1395 communicates within the SCPI (Standard Commands for Programmable Instruments) and IEEE 488.2 standards. Therefore, the Model 1395 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 1395 and indicates their hierarchical relationships. The IEEE 488.2 Common Commands are listed in a separate table (Table 3-5). The SCPI Command Table is organized as follows:

Keyword	Parameter Form	Notes
[SOURce]		
:FREQuency		
[:CW]	<numeric_value>	
:MODE	<list>	
STARt	<numeric_value>	
STOP	<numeric_value>	

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.

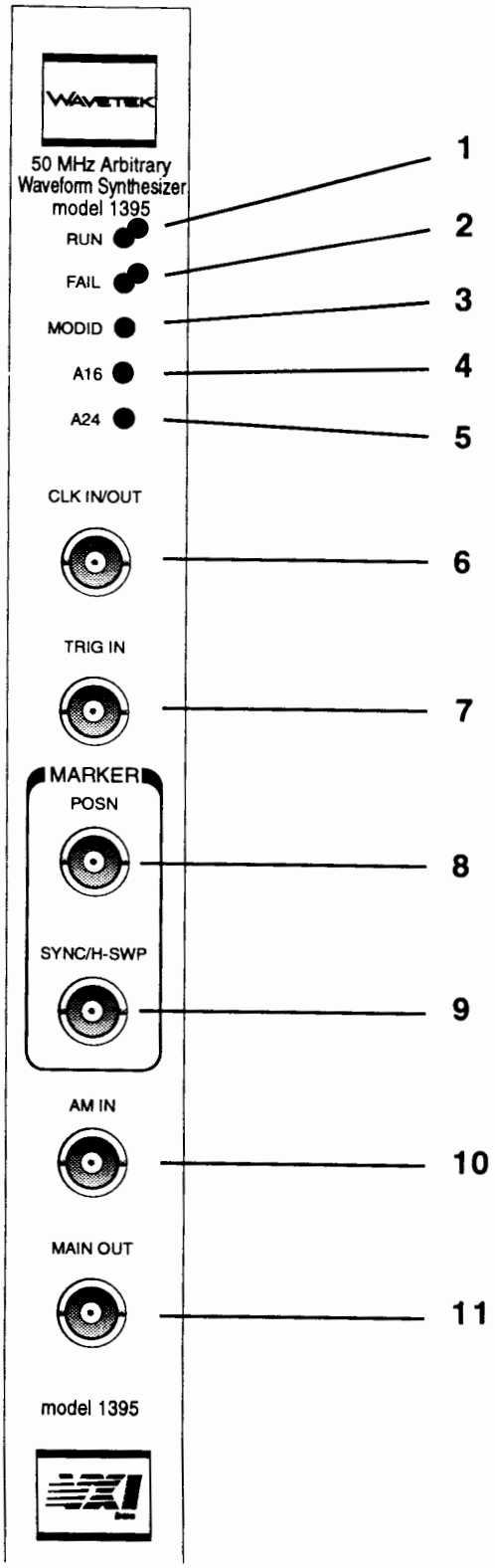


Figure 3-1. Model 1395 Front Panel

Table 3-1. Model 1395 Front Panel

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	MODID LED Indicator	This indicator turns on momentarily indicating the Resource Manager has detected the presence of the Model 1395.
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 in the range of 125 mHz to 100 MHz.
7	TRIG IN Connector	This connector receives the external trigger signal for the Model 1395's triggered, gated, and sequence modes.
8	POSN Connector	This connector outputs TTL pulses which can be set as Position Markers at specific address locations within arbitrary waveforms.
9	SYNC/H-SWP Connector	This connector outputs TTL pulses as waveform Synchronization Markers; and when the generator is sweeping frequency, outputs a horizontal sweep voltage to indicate position within the sweep.
10	AM IN Connector	This connector is the input for external signals to Amplitude Modulate the MAIN OUT.
11	MAIN OUT Connector	This connector supplies the generator's waveform output. Output level is 15 mVpp to 15 Vpp into 50Ω.

The two forms of Program Headers are commonly referred to as "commands" (see paragraph 3.3.2) or as "queries" (see paragraph 3.3.2.6). A Program Header followed by Program Data is defined as a Program Message Unit (paragraph 3.3.2.1).

In the example above, 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 1395 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 parser (included in instrument firmware) to the root node. A leading colon at the root node location is unnecessary (see paragraph 3.3.2.3).

3.3.1.1 Long and Short Form Keywords

The Model 1395 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 1395.

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 1395. For an understanding beyond what is covered in this paragraph, refer to the appropriate SCPI and IEEE 488.2 documents.

Operating the Model 1395 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 1395 will record any unrecognized parts of the command string as an error.

Table 3-2 shows the Model 1395 message structure and message relationships. Refer to this while working within this paragraph.

Note

The Model 1395 records programming errors in its 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 delineated 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 1395 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.

3.3.2.4 Parameter Forms

For the Model 1395, 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.

Table 3-2. Model 1395 Command Summary

KEYWORD	PARAMETER FORM	NOTES
CALibration		
[:**ALL**]?		
:**DATA**	<block>	
:**AFC**orrection	<point>,<frequency>,<gain>	
:**AMPL**itude		
[:**GAIN**]	<numeric_value>	(0.0, **0.0**, 1000)
:**OFFS**et	<numeric_value>	(0.0, **2048**, 4095)
:**AMZ**ero	<numeric_value>	(0.0, **2048**, 4095)
:**OFFS**et		
[:**GAIN**]	<**POS**itive | **NEG**ative>,<numeric_value>	(0.0, **0.0**, 1000)
:**OFFS**et	<**POS**itive | **NEG**ative>,<numeric_value>	(0.0, **2048**, 4095)
:**PAZ**ero	<numeric_value>	(0.0, **2048**, 4095)
:**SCMZ**ero	<numeric_value>	(0.0, **2048**, 4095)
:**STOR**e		
:**STAT**e	<Boolean_data> (**ON**)	
INITiate		
[:**IMM**ediate]		
:**CONT**inuous	<Boolean_data> (**ON**)	
OUTPut		
:**CLOC**k		
:**FREQ**uency	<numeric_value>	(1e-1, **1e3**, 1e8)
:**SOUR**ce	<**RAST**er | **SYNT**hesizer>	
:**ECLTrg**<n>		<n> = 0 to 1 VXIbus
[:**STAT**e]	<Boolean_data> (**OFF**)	ECLTrigger lines
:**FILT**er		
[:**LPAS**s]		
:**SEL**ect	<**BESS**el | **ELL**iptic>	
[:**STAT**e]	<Boolean_data>	
[:**STAT**e]	<Boolean_data> (**OFF**)	
:**SUMB**us		
[:**STAT**e]	<Boolean_data> (**OFF**)	
:**TRIG**ger		
:**MARK**er	<**TRIG**ger | **POS**ition>	
:**SOUR**ce	<**BIT** | **BCOM**plete | **LCOM**plete | **INT**ernal>	
:**END**		
[:**STAT**e]	<Boolean_data> (**OFF**)	
:**TTLTrg**<n>		<n> = 0 to 7 VXIbus
[:**STAT**e]	<Boolean_data> (**OFF**)	TTLTrigger lines
RESet		
[**SOUR**ce]		
:**AM**		
[:**STAT**e]	<Boolean_data> (**OFF**)	
:**MODE**	<**AM**> | <**SCM**>	
:**CLOC**k		
:**CONF**igure	<**INP**ut | **OUTP**ut>	

Table 3-2. Model 1395 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:FREQuency		
[:CW | FIXed]	<numeric_value>	(1e-6, 1e3, 2.5e7)
:MANual	<numeric_value>	(1e-1, 1e3, 2e7)
:MODE	<CW | SWEep | LIST>	
: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> (<SINusoid | TRIangle | SQUare | HFSQuare | DC | NHSine | PHSsine | PRAMp | NRAMp | PRNoise | USER | SMEMory | WTST>)	
:USER	<trace_name>	
:MODE	<FIXed | 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	<ZCRoss | 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	<AUTOmatic | TRIGgered >,<list_index>	(0, 3)
:DWELl	<numeric_value>,<list_index>	(1, 1, 65535),(0,3)
:FUNCtion	<trace_name>,<list_index>	(0, 3)
:LENGth	<numeric_value>	(2, 2, 4)
:TRIGger		
:MODE	<SYNChronous | ASYNchronous>	
:SENSe	<EDGE | LEVel>	
:SUMBus		
[:STATe]	<Boolean_data> (OFF)	
:ATTenuation	<numeric_value>	(0, 0, 42)
:SWEep		
:COUNt	<numeric_value>	(1, 1, 1e6)
:DIRection	<UP | DOWN>	
:SPACing	<LINear | LOGarithmic>	
:TIME	<numeric_value>	(30e-3, 1, 1e3)
:MODE	<CRESet | TRESet | HRESet | CREVerse | TREVerse | HREVerse | MANual>	

Table 3-2. Model 1395 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:VOLTage		
[:LEVel]		
[:IMMediate]		
[:AMPLitude]	<numeric_value>	(0, 1, 7.5)
:OFFSet	<numeric_value>	(-7.5, 0, 7.5)
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>,. . .	
: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>	
:POINt	<trace_name>,<point_index>,<point_value>	
:DELete		
[:NAME]	<trace_name>	
:ALL		
:DIRectory?	<name>,<size>,<limit>,<limit>. . .	
: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)

Table 3-2. Model 1395 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
TRIGger		
:COUNt	<numeric_value>	(1, 1, 1048575)
:GATE		
[:**STAT**e]	<Boolean_data> (**OFF**)	
[:**IMM**ediate]		
:**POL**arity	<**POS**itive | **NEG**ative>	
:**SOUR**ce		
[:**STAR**t]	<**INT**ernal | **EXT**ernal | **CHA**in | **ECH**ain | TTLTrg<n>>	<n> = 0 to 7 VXIbus TTLTrigger lines
:**ADV**ance	<**INT**ernal | **EXT**ernal | **CHA**in | **ECH**ain>	
:**TIM**er	<numeric_value>	(2e-7, 1e-3, 1e4)

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 <numeric_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 1395 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.

3.3.2.5 Program Message Terminators

The Model 1395 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 be delayed until the buffer fills.

3.3.2.6 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 1395 SCPI Commands

This paragraph introduces the operator to the Model 1395 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 1395 (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. Following is the CALibration Subsystem excerpted from the Command Table:

```
CALibration
  [:ALL]?
  :DATA     <block>
   :AFCorrection<point>,
          <frequency>,<gain>
   :AMPLitude
    [:GAIN]      <numeric_value>
    :OFFSet      <numeric_value>
   :AMZero <numeric_value>
   :OFFSet
    [:GAIN]      <POSitive|NEGative>,
          <numeric_value>
    :OFFSet      <POSitive|NEGative>,
          <numeric_value>
   :PAZero <numeric_value>
   :SCMZero      <numeric_value>
   :STORe
  :STATe    <boolean>
```

`CALibration[:ALL]?`

Performs a DC calibration of the output amplitude and offset voltage levels and stores the calibration data in nonvolatile 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

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 rang-

ing 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 gain of the amplitude control DAC. 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 gain of the output offset voltage control DACs. 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 be-

tween 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

```
INITiate
   [:IMMediate]
   :CONTinuous  <Boolean>
```

```
INITiate[:IMMediate]
```

This command is included to support the SCPI specification, but it does not alter the setup of the Model 1395.

```
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.

3.3.3.3 OUTPut Subsystem

```
OUTPut
  :CLOCk
    :FREQuency    <numeric_value>
    :SOURce RASTer | SYNThesizer
  :ECLTrg<n>
    [:STATe]      <Boolean>
  :FILTer
    [:LPASs]
      :SELect     BESSel | ELLiptic
      [:STATe]    <Boolean>
  [:STATe] <Boolean>
  :SUMBus
    [:STATe]      <Boolean>
  :TRIGger
    :MARKer TRIGger | POSition
    :SOURce BIT | BCOMplete | LCOM-
            plete | INTernal
    :END
      [:STATe]    <Boolean>
  :TTLTrg<n>
    [:STATe]      <Boolean>
```

```
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>

```
[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. 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<INTernal
        |EXTernal|ECLTrg0|ECLTrg1>
```

Selects the source of the reference oscillator.

- **INTernal** 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
        <AUTOmatic|TRIGgered>,<index>
```

Selects the conditions under which the sequence advances to the next trace.

- **AUTOmatic** Advances to the next trace in the sequence automatically after the repeat count.
- TRIGgered Waits for a trigger after the repeat count before advancing to the next trace in the sequence.

The index ranges from 0 to 3 as the Sequence Length ranges from 1 to 4. The query form `SEQuence:AD-Vance? <index>` returns `AUTO` or `TRIG` 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 1395 the number of elements in this list is limited to four. The index value is set from 0 to 3 as the `[SOURce:]SEQuence: LENGth` command ranges from 1 to 4. The query form `SEQuence:FUNCtion? <index>` returns the trace name for the segment selected by the index suffix.

```
[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 3 as the Sequence Length ranges from 1 to 4. The query form `SEQuence:DWELl? <index>` returns the numeric dwell value 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 4.

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

```
[SOURce]
 :AM
  :STATe  <Boolean> OFF
  :MODE   <AM|SCM>
 :CLOCk
  :CONFigure   <INPut|OUTPut>
 :FREQuency
  [:CW|FIXed]  <numeric_value>
  :MANual <numeric_value>
  :MODE   <CW|SWEep|LIST>
  :RASTer <numeric_value>
  :STARt  <numeric_value>
  :STOP   <numeric_value>
 :FUNCtion
  [:SHAPe]     <shape_name>
  :USER   <trace_name>
  :MODE   <FIXed|SEQuence>
 :LIST
  :FREQuency   <numeric_value>,
          <list_index>
  :POINts <numeric_value>
 :MARKer
  :POSition
   :AOFF  <trace_name>
   :POINt <trace_name>,
          <point_index>
  :SYNC
   :SOURce      <ZCRoss|BBITs>
   [:STATe]     <Boolean>
  :TRIGger
   [:STATe]     <trace_name>,
          <Boolean>
 :PHASe
  [:ADJust]     <numeric_value>
  :LOCK   <Boolean>
 :ROSCillator
  :SOURce <INTernal|EXTernal|
          ECLTrg<n>>
 :SEQuence
  :ADVance      <AUTOmatic|
          TRIGgered>,<list_index>
  :DWELl  <numeric_value>,
          <list_index>
  :FUNCtion     <trace_name>,
          <list_index>
  :LENGth <numeric_value>
  :TRIGger
   :MODE
          <SYNChronous|ASYNchronous>
   :SENSe <EDGE|LEVel>
 :SUMBus
  [:STATe]      <Boolean>
  :ATTenuation <numeric_value>
 :SWEep
  :COUNt  <numeric_value>
  :DIRection    <UP|DOWN>
  :MODE   <CRESet|TRESet|HRESet
          |CREVerse|TREVerse
          |HREVerse|MANual>
  :SPACing
          <LINear|LOGarithmic>
  :TIME   <numeric_value>
 :VOLTage
  [:LEVel]
   [:IMMediate]
    [:AMPLitude] <numeric_value>
    :OFFSet      <numeric_value>
```

```
[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).

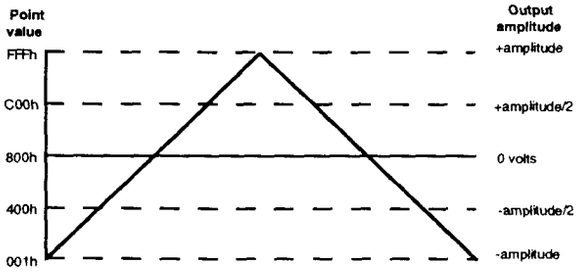
```
[SOURce:]SWEep:TIME
          <numeric_value>(1)
```

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 7.5 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 following diagram.



Trace Point Value vs. Amplitude Nomengraph

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.

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 -7.5 Vdc to +7.5 Vdc.

3.3.3.6 STATus Subsystem

```
STATus
  :OPERation
    :CONDition?
    :ENABle <NRf>
    [:EVENt]?
  :PRESet
  :QUEStionable
    :CONDition?
    :ENABle <NRf>
    [:EVENt]?
```

```
STATus:OPERation:CONDition?
```

Returns the contents of the Operation Condition Register. The Model 1395 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 1395 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 Model 1395 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 Model 1395 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 1395 accepts the command without performing any action.

- 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.
- 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.
- 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.

```
[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[: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.

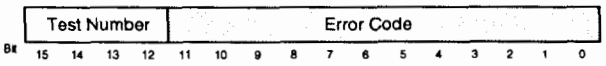
```
[SOURce:]MARKer:SYNC:SOURce
            <ZCRoss|BBITs>
```

Selects the method used to generate the SYNC marker output.

- ZCROss Selects the output of a comparator. 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.

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

3.3.3.9 Trace Subsystem

```
TRACe
  :CATalog?
  :DEFine   <trace_name>,
            (<numeric_value>, |
            <trace_name>)
  [:DATA]   <trace_name>,(<block> |
            <trace_name>)
   :LINE    <trace_name>,
            <point_index1>,
            <point_value1>,
            <point_index2>,
            <point_value2>
   :POINt   <trace_name>,
            <point_index>,
            <point_value>
  :DELete
   [:NAME] <trace_name>
   :ALL
  :DIRectory?
  :FREE?
  :LIMits   <trace_name>,
            <start_index>,
            <stop_index>
  :MODE     <CW | RASTer>
  :POINts   <trace_name>,
            <numeric_value>
```

TRACe:CATalog?

Returns a string containing the names of all defined traces. Trace names are separated by commas.

```
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[: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.

> ***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.

[SOURce:]SEQuence:TRIGger:MODE
<**SYNChronous**|ASYNchronous>

Defines the transition mode from the current trace to the next trace in the sequence after a trigger is received.

- **SYNChronous** After advance conditions are met, waits until end of current trace before selecting next trace in sequence.
- ASYNchronous After advance conditions are met, selects next trace in sequence immediately.

[SOURce:]SEQuence:TRIGger:SENSe
<**EDGE**|LEVel>

Defines the active portion of the trigger signal.

- **EDGE** The rising edge of the trigger signal initiates an action. When all other advance conditions are met, the triggering signal must make a false to true transition to define a trigger event.
- LEVel The level of the trigger signal initiates an action. If the other advance conditions become true, and the triggering signal level is true, then the trigger event occurs.

[SOURce:]SUMBus[:STATe]<ON|**OFF**>

Enables the analog sum input from the backplane.

[SOURce:]SUMBus:ATTenuation
<numeric_value> **(0)**

Controls the level of attenuation in the path of the analog sum input from the backplane. This command accepts an integer value in the range 0 to 42 and rounds the value down to one of the following attenuation levels: 0 dB (÷1), 6 dB (÷2), 12 dB (÷4), 18 dB (÷8), 24 dB (÷16), 30 dB (÷32), 36 dB (÷64), or 42 dB (÷128).

[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:

- **CRESet** Sweeps from the start frequency to the stop frequency and then returns to the start frequency in a continuous loop.
- TRESet Waits for a 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.

- **LINear** 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.

3.3.3.10 TRIGger Subsystem

```
TRIGger
  :COUNt    <numeric_value>
  :GATE
    [:STATe]      <Boolean>
  [:IMMediate]
  :POLarity<POSitive | NEGative>
  :SOURce
    [:STARt]      <INTernal|EXTernal
                |CHAin|ECHain|TTLTrg<n>>
    :ADVance      <INTernal|EXTernal
                |CHAin|ECHain>
  :TIMer    <numeric_value>
```

`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 524,288 for triggered sequences (`FUNCtion: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.
- INTernal 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.

3.3.3.11 High Speed Binary Waveform Transfer

The Model 1395'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 the figure below). This format for block data transfer makes it possible to rapidly move the large amount of data required for arbitrary waveforms.

STATus:QUEStionable:CONDition?

Returns the contents of the Operation Condition Register. The Model 1395 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 1395 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 Model 1395 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 1395 supports this query, but will only return the value "0", indicating operational condition.

3.3.3.7 SYSTem Subsystem

```
SYSTem
  :ERRor?
  :DATE     <year>,<month>,<day>
  :TIME     <hour>,<minute>, <sec-
            ond>
  :VERSion?
```

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>

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]?
  :RAM?
```

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.

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"		

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.

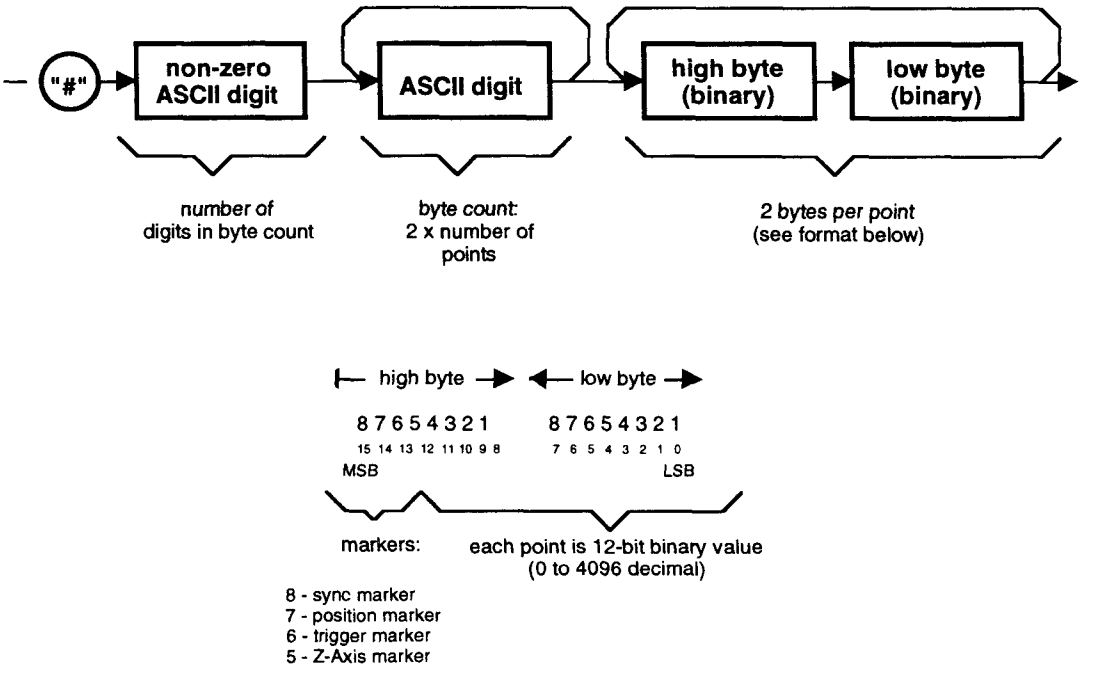
Prior to downloading a Waveform using the `TRACe [:DATA] <trace_name>,<block>` command, the Trace must be first defined. Send the `TRACe:DEFine <trace_name>, <value>` command to define a Trace with a size of `<value>` points under the name `<trace_name>`. This also presets the Trace Limits for this trace 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 1395'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 1395 to generate many error messages.

Definite Length Arbitrary Block Data Format



OUT connector, with R_S representing the Model 1395 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 1395.

Table 3-6 Input and Output Impedances

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

3.4.2 Input/Output Protection

The Model 1395 provides protection for internal circuitry 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 not effected by a reset), the Model 1395 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
FunctionSINusoid
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)
SUMBUS OFF (not input or output)
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 AUTOmatic
Sequence Length 1
Sequence Dwell 1
Sequence Trigger EDGE

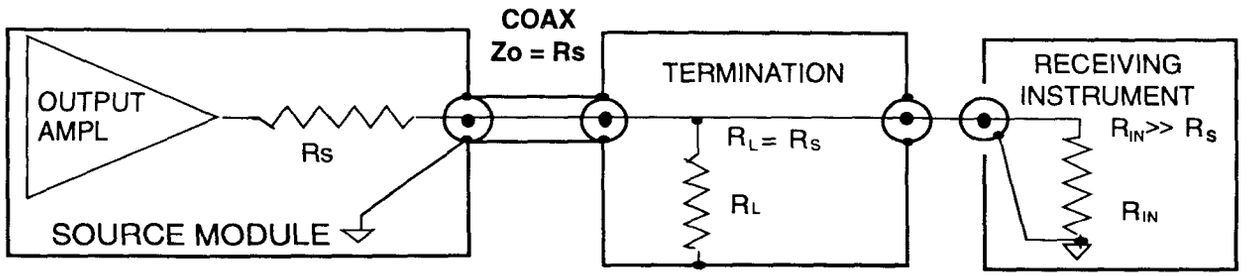


Figure 3-3. Output Termination

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 CALibrate[: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 calibration 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 1395 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 1395.

3.4 Model 1395 Operation

The following paragraphs describe messages for various modes of operation for the Model 1395.

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 3-3 shows proper 50Ω termination for the MAIN

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

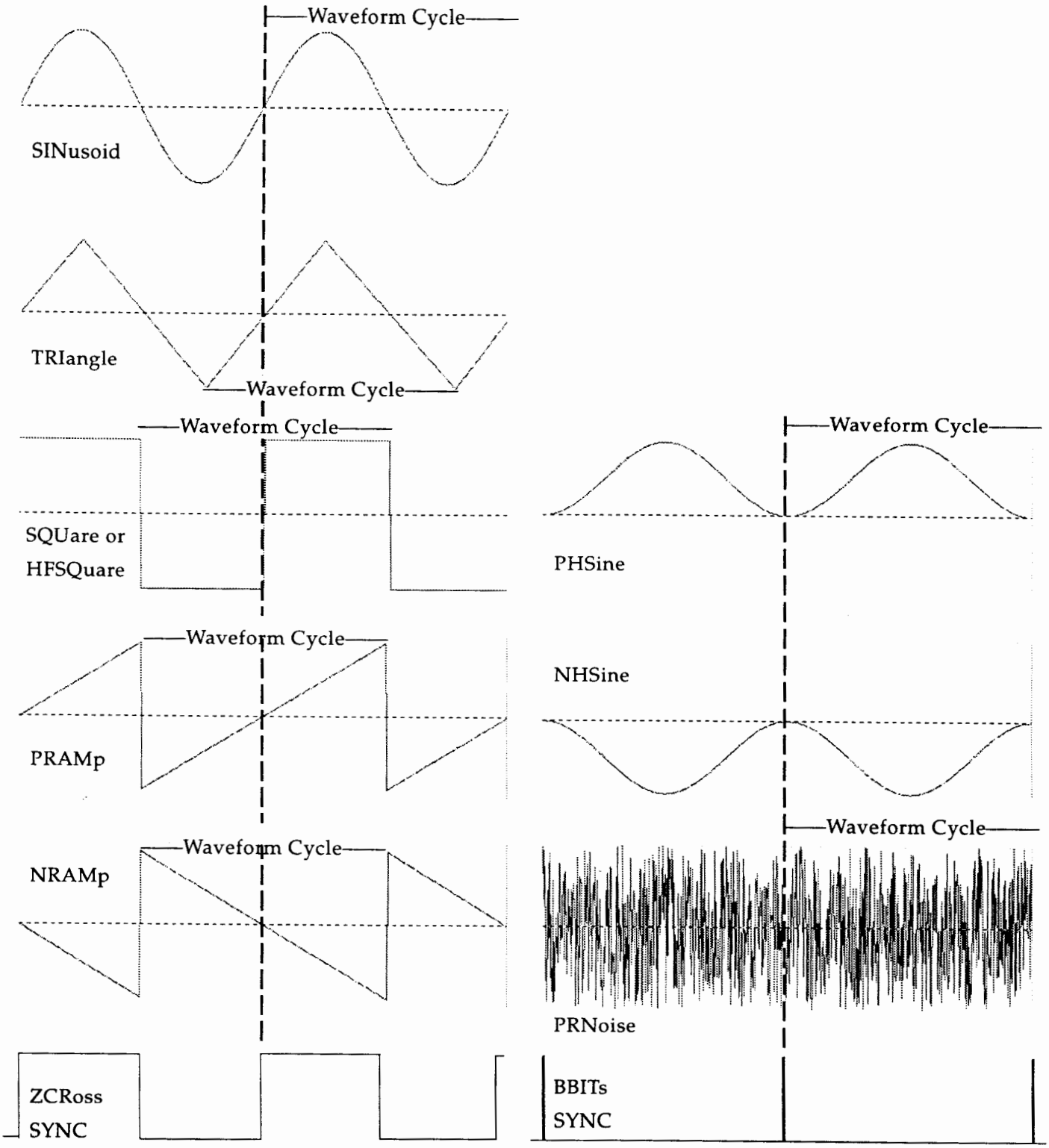


Figure 3-5. Continuous Waveform Characteristics

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 1395's Trace Memory. Trace Memory is volatile, high-speed RAM organized as 32,768 (optionally 131,072) addressable points.

3.4.5.1 Trace Definition

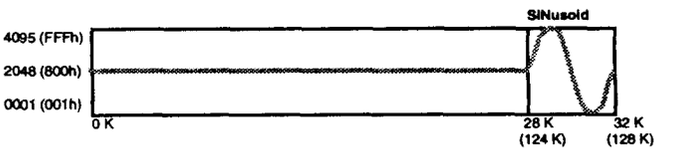
An arbitrary waveform is referred to in the SCPI language as a Trace. A Trace is a block of contiguous points in the Trace Memory which are referenced by a <trace_name>. The "size" of the Trace is the number of points reserved for waveform data. Each data point uses 16 bits, with 12 bits used to describe 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 1395 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 1395's default configuration can be observed by cycling power, setting up the Model 1395 and an oscilloscope per Figure 3-4, 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 1395 firmware manages the start and stop addresses of each Trace so that this simple model is maintained, even as Traces are deleted or re-sized 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 1395 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 Select 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?"
```

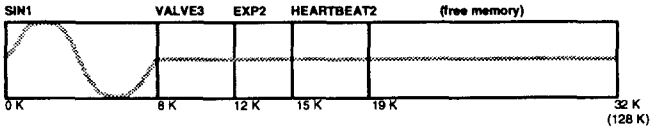
```
dev2: ibrd 20
```

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:

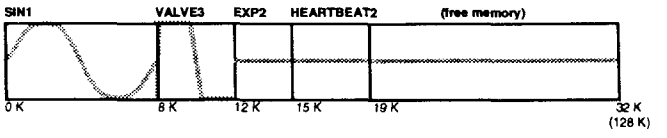
```
dev2: ibwrt "trac:data:poin valve3,
0,1000;poin valve3,1,2000;poin
valve3,2,3000;poin valve3,3,3500;
poin valve3,4,4000;mark:sync:sour
zcr;sour bbit"
```

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

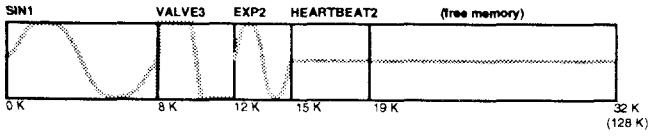
The SIN1 and VALVE3 examples demonstrate Copying a Standard Function to a new Trace. This same process can also Copy an existing USER Function to a new Trace. The destination Trace must be first defined so that it has a name and a size. The destination Trace size may be smaller, equal to, or larger than the source Trace. The Model 1395 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"
```

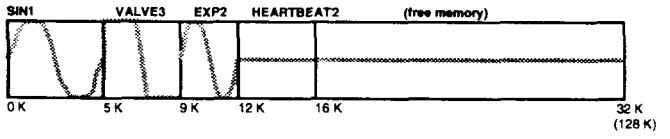
Verify that the EXP2 waveform now appears in Trace Memory per the following drawing:



Waveform Re-sizing 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"
```

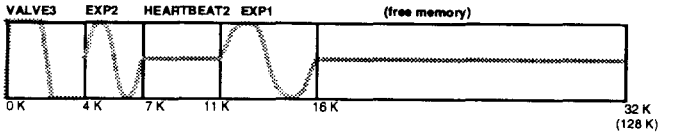
Verify that SIN1 is now approximately five divisions long. The Trace Memory should now be as follows:



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 1395 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 1395'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"
```

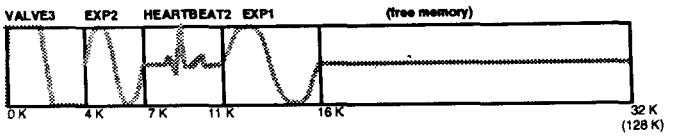
View the entire waveform by resetting the Limits:

```
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 heart-
beat2, 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

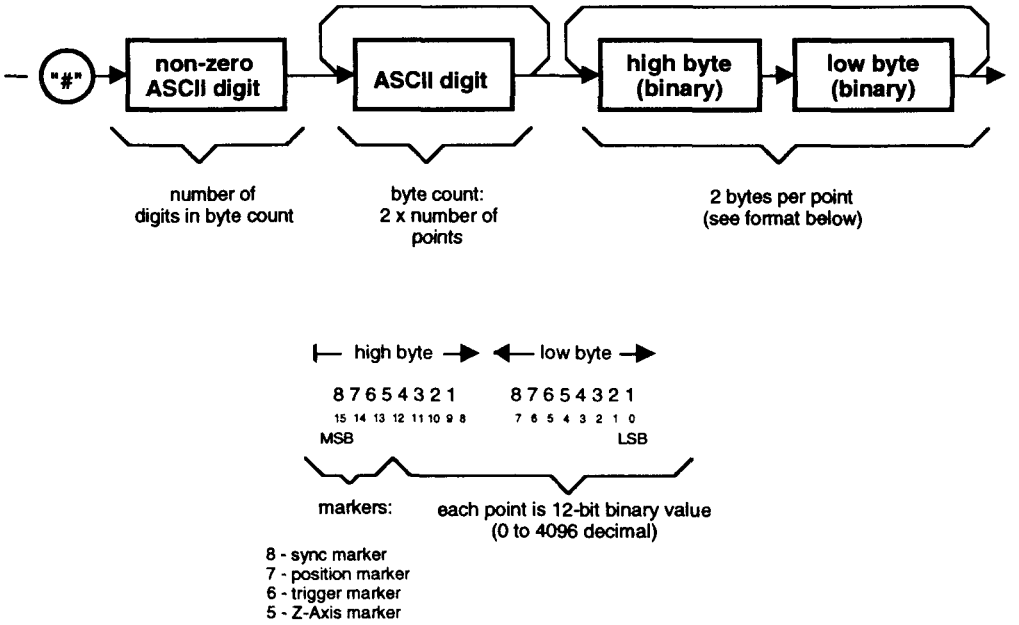


Figure 3-6. Definite Length Arbitrary Block Data Format

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 1395'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 3-6 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 3-6, the four markers have the following definitions:

Z-AXIS Not used in the Model 1395, 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 3.4.14). 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 POSN 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 MARKER SYNC/H-SWP 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 1395 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 1395 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 1395'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, or one half second elapses without receiving any data, 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 1395 to generate many errors.

Several examples of programs written in WaveTest™ BASIC 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 WaveForm DSP™ Download

WaveForm DSP™ is a Wavetek software product running under Windows which creates arbitrary waveform data files and downloads them to Wavetek ARBs using the GPIB or VXIbus interfaces. The WaveForm manual is very extensive, and should be used for proper set up of the GPIB interface and the Model 1395. The first two sections of the WaveForm manual are most important to the first time user. The following topics provide additional information relative to the Model 1395 and VXIbus ARBs in general.

GPIB Addressing

At the time that this manual was written, WaveForm was at revision level 1.11. This revision supports an IBM compatible PC operating as GPIB controller to "stand-alone" GPIB instruments and as "external host" to a VXIbus system.

The host requires a GPIB card and driver software. WaveForm supports GPIB cards using National Instruments drivers (the AT-GPIB and PCII/PCIIA cards) and GPIB cards which are "NEC 7210 compatible" (National's PCII and PCIIB cards, and I/Otech's 488-B and 488-Bplus cards). The "slot 0" or Resource Manager (RM) card inside the VXIbus chassis is not a stand-alone computer, but a GPIB bus to VXIbus translator, and the external host runs the control application. Examples of RM cards include the National Instruments GPIB-VXI and the Wavetek 1320.

In this "external host" configuration (see Figure 3-7), the programmer/operator needs to be aware of three levels of GPIB addressing; the GPIB card, the RM, and the ARB module (Servant).

At the card level, the computer's Base I/O Address, Interrupt (IRQ), and DMA Channel settings have to made for each particular card. Additionally, there can be one or two of the same type cards installed (primary and secondary). Section 1 of the WaveForm manual provides the needed information to set the cards and store the settings in the GPIB information file.

Once this process is completed, the WaveForm user needs only to select primary (GPIB0) or secondary (GPIB1) cards at the time of download.

The remaining GPIB addresses to be set are the RM (primary address) and the ARB module (secondary address). The addresses are set by selecting switch settings on the VXIbus modules (see section 2 of this manual for setting the Model 1395's module address). The primary and secondary addresses are treated together as *device information* (dev<n> where "n" is 1 to 16 for the primary card and 17 through 32 for the secondary card). Each instrument or module to be controlled over the GPIB should have a *device* assigned to it. The programmer/operator goes through the process of setting primary, and if applicable secondary, addresses and storing them for each device. Once completed, the instrument can be addressed simply by calling its device. This process is also detailed in section 1 of the WaveForm manual. Once completed, the WaveForm user needs only to select the correct "dev<n>" at the time of download.

NOTE

The Model 1395 supports both Static (switch settings) and Dynamic Configuration (DC) of the secondary address. To use the 1395 with WaveForm over the GPIB it is best to use Static switch settings so that the 1395's address is fixed and will agree with its dev<n> assignment, even if the modules are moved around inside the chassis.

For the First-Time WaveForm User

Once all of the preparation described in the previous paragraph is completed, the user should go through the following steps to verify communication between the WaveForm application and the Model 1395.

1. Cycle power to the Model 1395. (Alternatively, send the `*CLS` command to flush any errors out of the SYSTem:ERRor queue.) WaveForm checks for errors after each command sent and displays them in a dialog box. The error queue can contain up to 8 error messages which have accumulated since the module was powered on.
2. Start WaveForm. Press "OK" as the start-up dialogs are displayed.
3. Select "Open" from under the "File" menu. Open the file "demo.wfm". A "(sin x)/x" waveform will appear in the top window.
4. Select "Select Download. . ." from under the "Options" menu. A dialog box appears to make control selections to download the waveform to the Model 1395.
5. From under the "Model:" selection box, select "Model 1395-32K", or "Model 1395-128K" if the expanded memory option is installed.

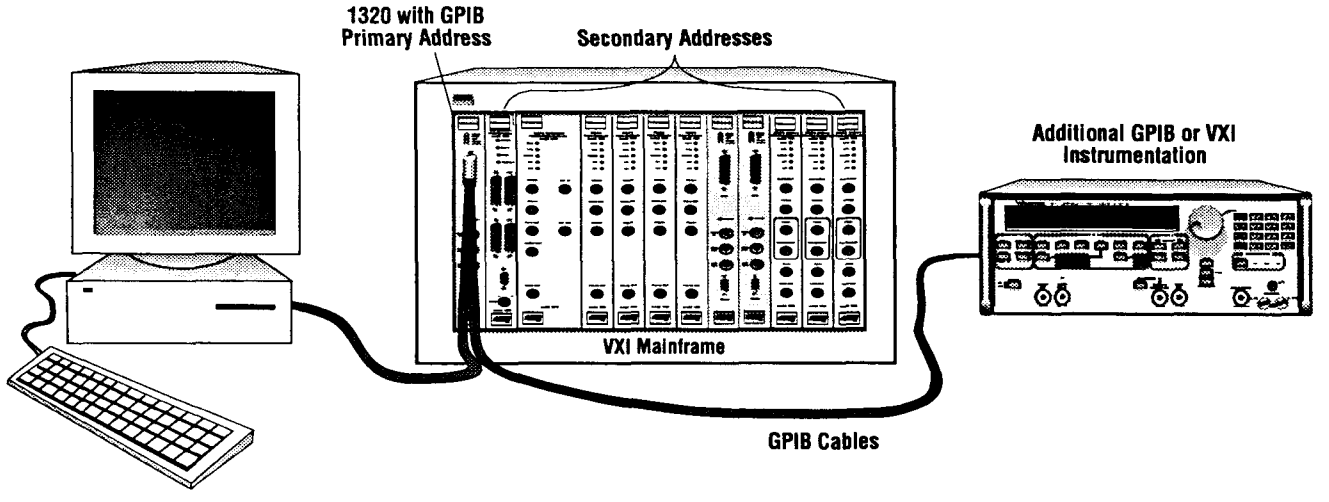


Figure 3-7. VXIbus System Using "External Host" GPIB Controller

6. The "Destination:" selection will default to "GPIB0". This is the most common selection, unless there is a second GPIB card installed and the Model 1395 device number is under the secondary card.
7. In the "Device Address" box, modify the default "DEV9" to "DEVn", where "n" is the device which has been preprogrammed for the RM's primary address and the Model 1395's secondary address.
8. The "Size" setting should be defaulted to "1024", the size of the selected (sin x)/x waveform. This size may be edited to any number of points from 8 to 32K (128K), but if it is changed, the "Mode:" should be changed from "Point to Point" to "Stretch to Fit".

   NOTE

   The Trace created by WaveForm DSP™ for downloading to a SCPI instrument has the name "WAVEFORM". This is not user editable. The "preamble" string sent prior to the binary data first DELetes the Trace with the name WAVEFORM and then DEFines it again with a size equal to the "Size" setting from the "Setup Download. . ." dialog. Therefore, it is best to avoid using the Trace name of WAVEFORM if also using WaveForm DSP™ with a SCPI instrument.

9. Save this download setup by pressing "OK" to accept these settings and then select "Save Setup" from under the "Options" menu.
10. Set up the 1395 and an oscilloscope according to Figure 3-4. Select "Exec Download" from under the "I/O" menu. The waveform should appear on the oscilloscope at WaveForm's default settings of 1 kHz waveform frequency (if changes were not made to the "Setup Waveforms. . ." dialog and if "Size" was not changed from 1024 points) and 1 V peak (into 50Ω) amplitude.

Follow the tutorials in the WaveForm DSP manual to create and download waveforms to the Model 1395 using the control settings to vary the waveform's parameters.

WaveForm DSP Download File Structure

The file structure, although transparent to the user, is an ASCII "preamble" of SCPI commands to set up the Trace parameters, a data transfer in Definite Length Arbitrary Block Data format, and a "postamble" of more SCPI commands if specified by the user. WaveForm uses a data block size of 1024 points, and the Trace Limits command is used to break up long waveforms into 1K blocks.

3.4.6.3 Shared Memory Transfer

The Model 1395'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 1395 does not support the Shared Memory Protocol, its Shared Memory can be used by the Commander and other instruments which do. In this case, the issue of interest to the Model 1395 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 3-6 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 3-6 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 8 to 32K points (128K points with the Option).

A 64 kByte 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 1395's Shared Memory and then transfers it to the 1395'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 1395 to output the selected Trace continuously. Changing this setting to `OFF` sets the Model 1395 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.

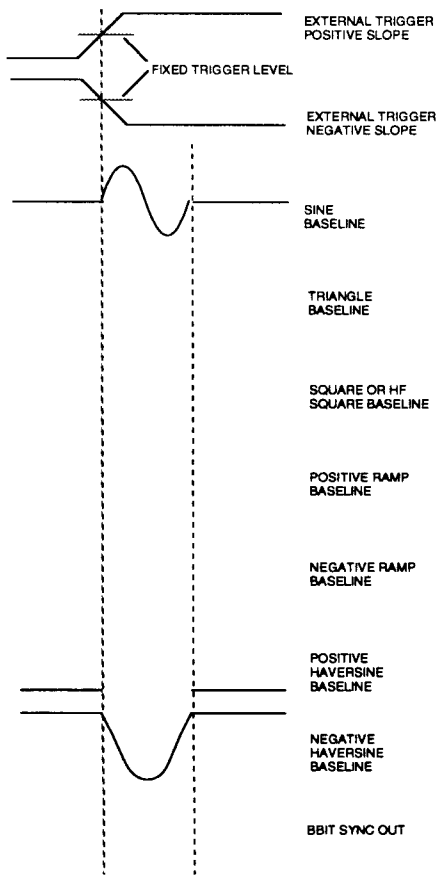


Figure 3-8. Triggered Waveform Characteristics, Count = 1

3.4.7.1 Triggered Operation

In the triggered mode, the Model 1395's output remains quiescent (at a dc level) until triggered by the selected trigger source. All Model 1395 functions may be triggered. When triggered, the Model 1395 produces one complete Trace, 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 3.4.14.1).

To set the Model 1395 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 1395 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 1395 UUT. The third line sets the UUT's output on and turns on the SYNC OUT. The fourth line changes the sine wave's 1 kHz default frequency to 10 kHz 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 1395. 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 5 MHz. Set the external generator for approximately 1 kHz. Connect the external TTL signal to the Model 1395's TRIG IN BNC. Verify that the triggered TRIangle function returns.

VXIbus TTL Trigger Lines Input

The Model 1395 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 TTLTrigger 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 10 kHz sine wave to be internally triggered at a 1 kHz 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 1 kHz 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
```

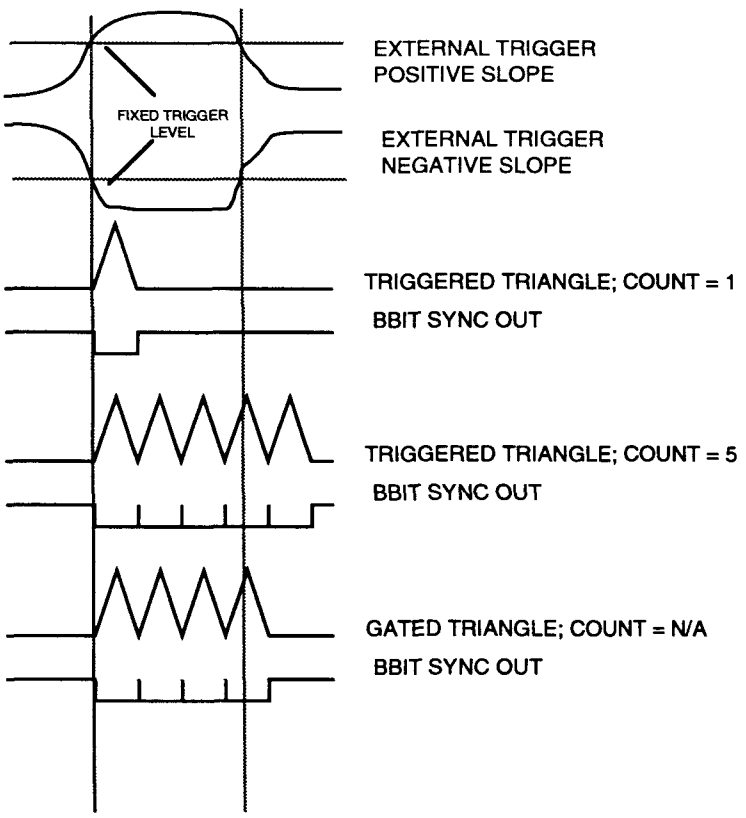


Figure 3-9. Gate/Burst Waveform Characteristics

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 1395 operates as a burst generator. The trigger COUNt is programmable up to 1,048,575 for Traces and up to 524,288 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 1395'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 1395. 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 1395 is in a triggered mode of operation.

Proceed with Example 3 by first removing the external signal from the Model 1395'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"
```

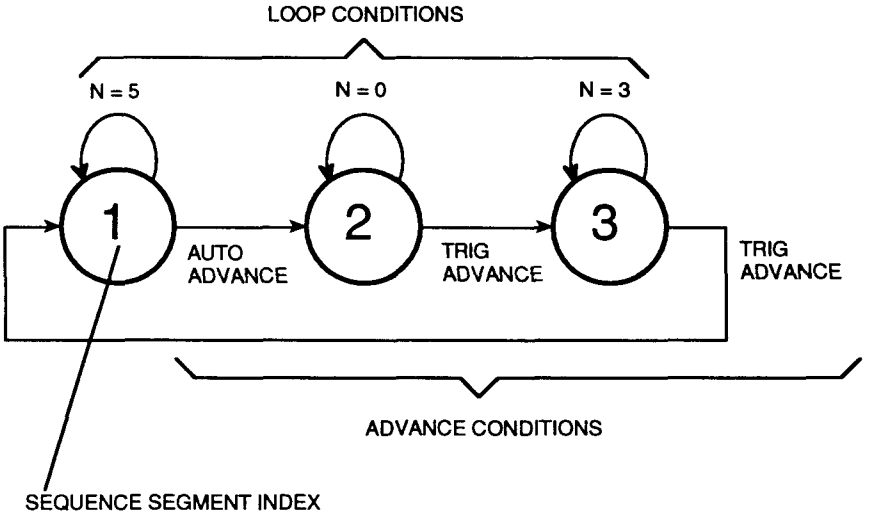


Figure 3-10. CONTinuous Sequence State Diagram

Set the 1395 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 the triggered operation with COUNt = 1.

3.4.7.2 Gated Operation

Gated operation is identical to the triggered operation, except the output from the Model 1395 starts from the quiescent state, produces continuous waveforms for the duration of the trigger signal, then returns to the quiescent state. All waveforms may be gated.

To view gated operation, set up the Model 1395 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 1395 ARB to accomplish even more complex tasks. A Sequence links together from two to four waveforms, as shown in Figure 3-10. A State Diagram best displays the logic involved in setting up a Sequence using the 1395. 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 according to the Advance Condition. Sequences can be run in both the CONTinuous and the TRIGgered Mode. When in CONTinuous, the Sequence is always restarted when the final segment's Loop and Advance Conditions are met.

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 1395 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 seg1"
dev2: ibwrt "func user"
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 4). 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 65,535 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 three Advance Conditions to consider with the Model 1395:

1) The Loop Count is set to some value "n" ranging from "1" to "65,535" and the SEQuence:ADVance selection is AUTOmatic. Under these conditions, the segment will repeat "n" times and then advance to the next segment without requiring a trigger event.

2) The Loop Count is set to "0" (continuous) and the SEQuence:ADVance selection is TRIGgered. Under these conditions, the segment will repeat continuously until the trigger event is true, and then advance to the next segment.

   NOTES

   When the Loop Count is "0" (continuous) the only proper selection for the SEQuence:ADVance selection is "TRIGgered". Selecting "AUTOmatic" will generate an error.

   Branching to the next segment can take place immediately (ASYNchronous) or after the current segment is complete (SYNChronous) depending upon the SE-Quence:TRIGger:MODE setting.

3) The Loop Count is set to some value "n" ranging from "1" to "65,535" and the SEQuence:ADVance selection is TRIGgered. Under these conditions, the segment will repeat "n" times and then remain quiescent at the level of the last point in the segment, and then branch to the next segment when the trigger event is true.

   NOTES

   The SEQuence:TRIGger:MODE setting can be programmed to SYNChronous or ASYNchronous. However, since Loop Count must be complete before responding to the trigger event, this Advance Condition will result in complete segments being created with either setting.

   For a given trigger source, the trigger event may or may not be true depending upon the setting of the SEQuence:TRIG-ger:SENSe command. The "EDGE" setting requires a false to true transition after the Loop Count complete, whereas the "LEVel" setting only requires a true level at the time of Loop Count completion.

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:adv 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 1395'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 3-4 by disconnecting the 1395's TRIG 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.

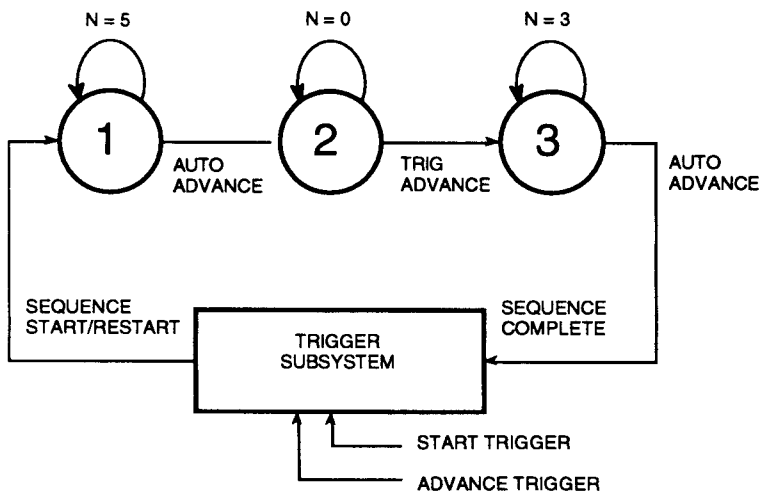


Figure 3-11. TRIGgered Sequence State Diagram

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. Now program the following:

```
dev2: ibwrt "seq:trig:sens lev"
```

Now the sequence restarts immediately after segment 3 ends, because the trigger level is high (true). A transition is not needed. Also note that the advance from segment 2 to segment 3 did not change, because the trigger level was low (false) at the time. Return the TRIGger SENSe to EDGE:

```
dev2: ibwrt "seq:trig:sens edge"
```

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 1395 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:adv auto,2"
```

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

NOTE

In the previous command, the Advance Condition for the last segment in the Sequence was changed from TRIGgered to AUTOmatic. Do not use a TRIGgered Advance Condition to branch to a Triggered Mode restart, or unpredictable operation may occur.

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 1395.

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 1395 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 1395 Unit Under Test (UUT, slot 2) according to Figure 3-4 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 1395 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 1395 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 3-4 by connecting the ARB OUT (MAIN OUT) of the slot 1 module to the AM IN of the 1395 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 MARKER SYNC/H-SWP output connector of the Model 1395 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 1395.

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 1395 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 1395 Unit Under Test (UUT, slot 2) according to Figure 3-4 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/H-SWP output of the UUT and connect it to the POSN output. The oscilloscope will loose sync. Set the oscilloscope to trigger internally from Channel 1. Note that the POSN output is a TTL low.

The Position Marker requires programming to appear at the POSN 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 1395 provides two mechanisms (other than direct programming) for rapid control of the generator's instantaneous frequency. These are covered in the next two paragraphs.

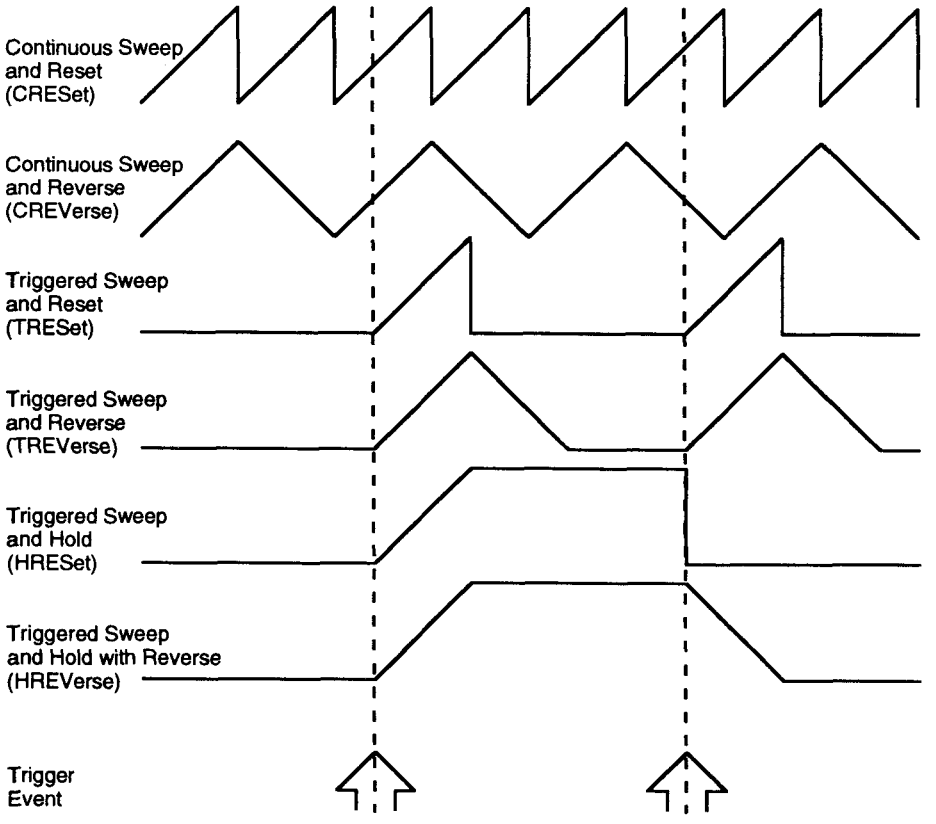


Figure 3-12. Sweep Mode Characteristics.

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 1395'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 1395 in its default states by sending the following:

```
:ibfind gpib0
```

```
gpib0:ibfind dev2
```

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

```
dev2: ibwrt "outp on"
```

Connect the Model 1395 Unit Under Test (UUT, slot 2) according to Figure 3-4 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 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 horizontal drive 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 1395 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/H-SWP 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.

3.4.12 SUMBUS Operation

Operations involving the VXIbus SUMBUS require two modules, one to drive and one to receive, which support the SUMBUS. At the time of this writing, the Wavetek Models 1391 and 1395 support the SUMBUS.

The VXIbus environment was designed to promote the inter-operation of modules within the environment of a chassis with closer coupling of signal timing than is possible between "stand-alone" equipment. Intermodule operation can be generalized as "serial" or "parallel" operation. Serial operation results in a final output being taken from one module. Parallel operation results in an output being taken from each module, with common timing characteristics between modules. SUMBUS operation is serial; paragraph 3.4.14, Intermodule Operation, is concerned with parallel operation.

The VXIbus SUMBUS is intended to operate as an analog summing node. The SUMBUS is a single 50Ω transmission line passing the length of the VXIbus chassis backplane. It is terminated into a 50Ω resistor at each end, resulting in an impedance of 25Ω to ground. Each module looks like a high impedance "stub" from the SUMBUS line. Modules which support SUMBUS will have a Driver and a Receiver located on this "stub". The Driver, when enabled, will source current to and from the SUMBUS line. The SUMBUS signal is a voltage signal resulting from the algebraic sum of all Driver currents with a scale factor of 40 mA per volt (25Ω). A module's Receiver can buffer the SUMBUS signal and apply it as a summing input to its output amplifier. This way, a receiving module can have an output which is the composite of its own programmed output and the SUMBUS signal.

The programmer/operator needs to be aware of several limitations in order to use the SUMBUS effectively. Any number of modules can drive the SUMBUS, and any number of modules can receive from the SUMBUS. However, a given module can drive only or receive only. When one or more modules are set up to drive, the peak instantaneous current at the SUMBUS line must not exceed 40 mA. In order to output a certain level of the SUMBUS signal at the receiving module's output, the programmer/operator must know the receiver module's receive scale factor and then control the SUMBUS signal level accordingly. Additionally, the peak instantaneous value of the scaled and buffered SUMBUS signal and the module's own signal must not exceed the output amplifier's limits, or clipping will occur. The waveform contribution from the SUMBUS will also have reduced bandwidth and wider specifications for amplitude accuracy and waveform dc offset than outputs appearing directly at the MAIN OUT. See Section 1 of this manual for SUMBUS specifications.

The following tutorial, Example 8, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1395's SUMBUS operation. The tutorial assumes a Model 1395 in slot 1 using secondary address 1, and a Model 1395 (the UUT) in slot 2 using secondary address 2. The 1395 in slot 1 will drive the SUMBUS, and the UUT will receive the SUMBUS and provide the composite output at its MAIN OUT.

Example 8

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 1395 in its default states by sending the following:

```
dev2: ibfind dev1
dev1: ibwrt "res"
dev1: ibwrt "outp on;:mark:sync on"
```

Then set up "dev1" to drive a signal to the SUMBUS. This signal is a full amplitude 10 kHz sine wave. Wavetek's SUMBUS instruments have their Driver scale factors set such that a full output level at the main output causes the SUMBUS to be driven at ±40 mA (±1 volt across 25Ω). Similarly, their Receiver scale factors are set such that a ±1 volt signal at the SUMBUS, if unattenuated, will drive the main output to full amplitude. This can be seen by taking the product of the specified Driver and Receiver scale factors from Section 1. If this product, after considering the voltage developed by the Driver into 25Ω, is 1 V/V, then end-to-end scale factor is unity. A unity overall scale factor indicates that whatever signal is at the output of the driving module will be a component of the composite output at the receiving module (with Receiver attenuation set for 1:1).

It is best to set up the driving module to full amplitude in order to drive the SUMBUS to its full ±1 volt amplitude. This produces the best amplitude accuracy and signal-to-noise ratio with the lowest dc offset. Then use the receiver's attenuator to set the level of signal contribution from the SUMBUS. Program "dev1" as follows:

```
dev1: ibwrt "freq 1e-4"
dev1: ibwrt "volt 7.5"
dev1: ibwrt "outp:sumb on"
```

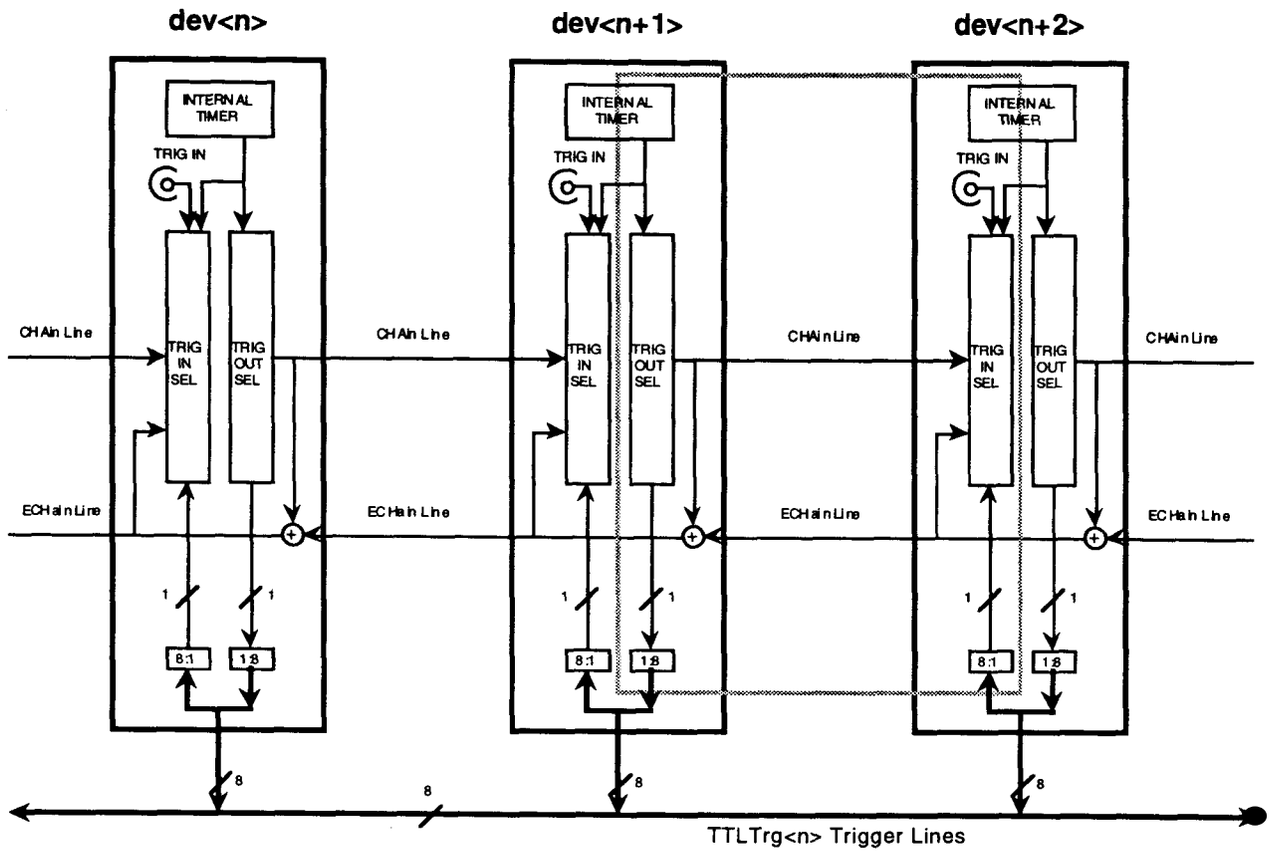


Figure 3-13. Intermodule Triggering Backplane Connections.

Now set up the UUT, "dev2", to receive the SUMBUS signal:

```
dev1: ibfind dev2
dev2: ibwrt "volt 5"
dev2: ibwrt "sumb:att 12;:sumb on"
```

The UUT's MAIN OUT signal will be a 10 Vpp (5 volts peak into 50Ω), 1 kHz sine wave with an approximate 3.75 Vpp, 10 kHz sine wave riding on it, for a total approximate 13.75 Vpp composite signal. This is safely below the 15 Vpp maximum, so no output clipping will occur.

The 3.75 Vpp figure is obtained as follows:

1) "Dev1" is set to full 15 Vpp amplitude. Therefore, it is driving the SUMBUS to its full 1 Vpp amplitude.
2) The full amplitude SUMBUS signal will drive the UUT's output to 15 Vpp full scale if unattenuated.
3) The SUMBUS receiver attenuator was set to "12", which indicates -12 dB attenuation. This is a 1:4 voltage ratio, so the SUMBUS contribution is 15 Vpp divided by 4, or 3.75 Vpp.

3.4.13 Clock Input/Output Operation

The 1395'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.

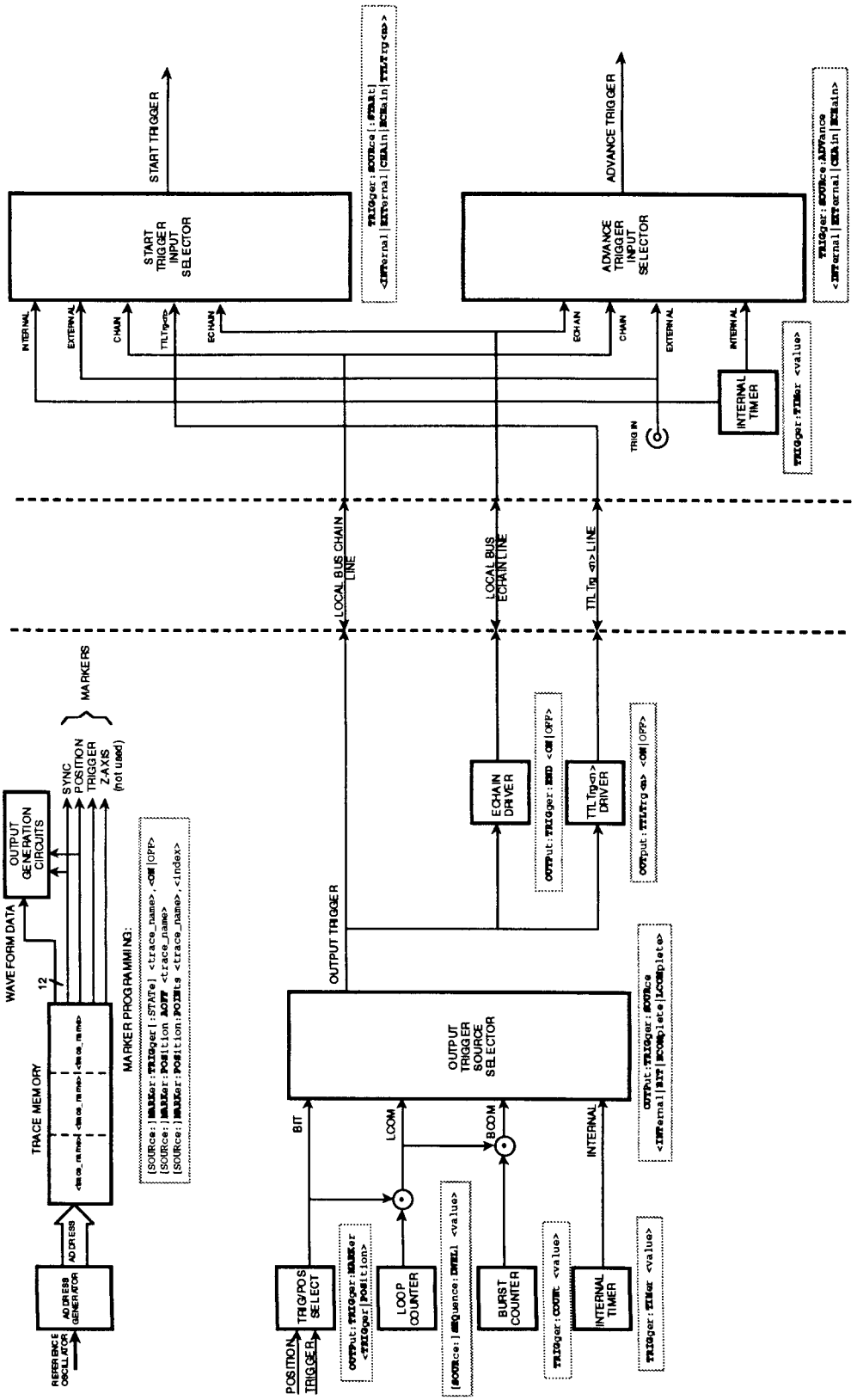


Figure 3-14. Intermodule Triggering Command Reference.

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 1395'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 1395 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 1395 Unit Under Test (UUT, slot 2) according to Figure 3-4 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/H-SWP 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 2 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/H-SWP 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 1395'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 1395. 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.

3.4.14 Intermodule Operations

3.4.14.1 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 1395 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 1395s. 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.

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/H-SWP 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 1395. 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 1395 in the trigger chain. There are two input trigger selectors on a Model 1395, 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 1395's Intermodule Triggering operation. The tutorial assumes a Model 1395 in slot 1 using secondary address 1, and another Model 1395 in slot 2 using secondary address 2. The 1395 in slot 1 will drive and receive the ROSCillator on ECLT1, and will drive the Local Bus CHAin and TTL Trigger lines. The 1395 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 1395 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/H-SWP 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"
```

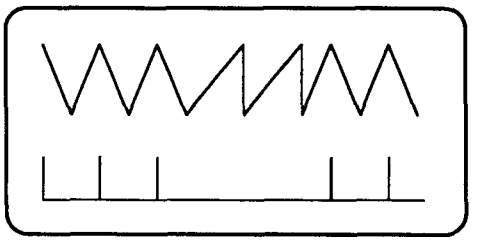
Start the sequence:

```
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/H-SWP connector to the POSN 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"
```

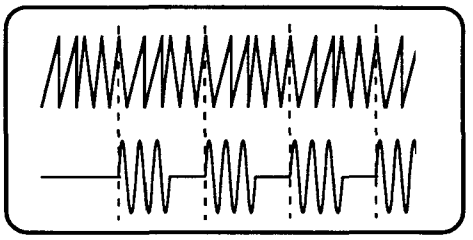
This command selects the Position marker as the BIT trigger signal. The BIT signal (identical to the position marker on the oscilloscope) is then ANDed with the triangle's Loop Complete signal to produce the "LCOM" trigger signal, which occurs at the positive peak (last point in Trace) of the last triangle 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"
```

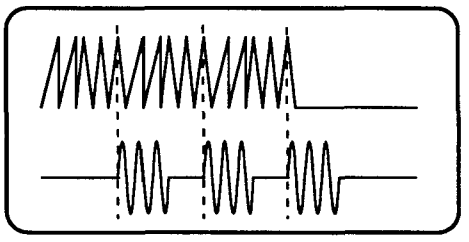
Move the cable from the POSN output of "dev1" to the MAIN OUT of "dev2". Observe the triggered waveforms.



Next modify the Sequence so that it is triggered, as follows:

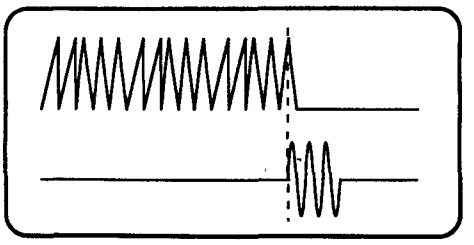
```
dev2: ibfind dev1
dev1: ibwrt "trig:sour:star int;:
trig:tim 3e-4;:init:cont off"
```

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.

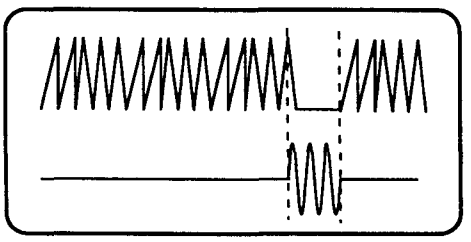
As a final part in this example, "dev2" will be set up to drive the ECHain line back to "dev1", using its trigger marker. Then "dev1" will run the Sequence and trigger "dev2". Following its trigger "dev2" will run its waveform and then trigger "dev1". This way activity can be sustained, alternating back 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.

3.4.14.2 Intermodule Phase Lock

The following tutorial, Example 11, takes the operator through the SCPI programming steps to demonstrate the use of the Model 1395's Phase Lock operation. The tutorial assumes a Model 1395 in slot 1 using secondary address 1, and a Model 1395 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 1395 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 3-4 by connecting the MAIN OUT from the Master to Channel 1 of the oscilloscope. Externally trigger the oscilloscope from the Master's SYNC/H-SWP 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 turn-around time.

4.2 CALIBRATION

Calibration is the process of *Scheduled Maintenance* as described in this section of the Model 1395 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 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	HP 3478A or equivalent.
Oscilloscope	1	Tektronix 2465 or equivalent.
Freq Counter	1	HP 5334A Universal Counter.
Signal Gen	1	Wavetek 1395 or equivalent VXIbus generator, triggerable, ≤5% accuracy from 25 Hz to 50 MHz.
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 STANDARD TEST EQUIPMENT

The following list of test equipment is included in this ATS for reference only. Substitution of any equipment is allowed as long as adequate measurement accuracy is maintained. Test equipment and test methods should always provide at least 5 times better measurement accuracy than the specification of the parameter being tested.

Accessory type equipment such as coax cables and terminations are not listed here. Use of appropriate accessories for the type of test being performed is assumed. If an accessory item is critical to the accuracy of a measurement, its requirements will be defined in the test procedure.

Salient characteristics of the equipment listed below are not specified. If substitution of equipment is desired, a review of the test procedures must be made to ensure that the substitute has all salient characteristics.

Signal Sources:Wavetek Model 91 Synthesized Pulse/Function Generator

Universal Counter: HP 5334B

Digital Multimeter: Datron Model 1062

Oscilloscope: Tektronix Model 2465B

RF Voltmeter: Boonton Model 9200B

4.4.2 STANDARD TEST CONDITIONS:

Temperature: 25°C±10°C

Humidity: 10% to 90%

Altitude: Sea Level

Warm up: 1/2 Hour

4.4.3 TEST SPECIFICATIONS

This section outlines test specifications the module must meet in order to pass the acceptance test. During the testing of the module the frequency reference for the VXIbus chassis and all frequency related instruments in the test system should be derived from the same source.

4.4.5 VXIbus Interface

Operability is inferred since performing the tests specified in this document requires a functional VXIbus interface to control the instrument. Insure that the Service Request and Shared Memory is tested. Query the module for the version of the firmware installed.

4.4.6 Self Test

Purpose: Tests the basic integrity of the 1395 and the interface to the backplane.

Setup: Initiate a self test sequence. Verify that the 1395 returns a result of 0. (Pass/Fail)

Commands:

*tst?

4.4.7 Function Output On/Off

Purpose: Test functionality and performance of the function output on/off control relay.

Spec: >60dB isolation at 1kHz

Setup: Select 1kHz, 5Vpp sine. Terminate Arb Output in 50 Ohms. Verify off amplitude is ≤ 5mV. (Pass/Fail)

Commands:

*rst

Volt 5

4.4.8 Trigger Count

Purpose: Verify the functionality and programmability of the trigger count circuitry.

Setup: Select function sine, 1 MHz, Trigger mode, trigger count as specified, trigger source - manual or bus. Select BBITS as Sync Marker Source. Terminate Sync Marker Out into 50 Ohm. Verify specified number of cycles are completed after receipt of trigger.

Set Value	Spec Min	Spec Max
1,048,575	1,048,575	1,048,575

Commands:

```
*rst
output on
freq 1e6
mark:sync on
mark:sync:sour bbits
init:cont off
trigger:sour ext
trigger:count 1048575
*trg
```

4.4.9 Trigger Source

Purpose: Verify functionality and performance of trigger sources and selection logic.

Setup: On the 1395 define a 5 point sine waveform, set sample rate to 50 MHz, set mode to Triggered with the trigger count set to 1. Select BBITS as Sync Marker Source. Terminate Sync Marker Out into 50 Ohm.

External Trigger

Apply a TTL level (0.8V to 2.1V) trigger signal to external trigger input BNC. Trigger input should be properly terminated to ensure good waveform quality. Set external trigger frequency to 5 MHz and pulse width to 20nS. Select External as trigger source. Verify Sync Marker frequency aprox 5 MHZ. (Pass/Fail)

Internal Trigger

Select Internal as trigger source. Set trigger rate as shown. Verify Sync Marker frequency = programmed trigger rate. Note: For the slow trigger rates, it may be helpful to reduce the waveform sample rate in order to keep the waveform frequency and the trigger frequency within a 10:1 or 100:1 ratio. This will ease the measurements.

Commands:

```
*rst
trace:define mike,5
trace:data mike,sin
func user
mark:sync:sour bbits
mark:sync on
init:cont off
trigger:sour ext      Do External test here
trigger:sour int
trigger:timer 2e-7    Do Internal test here
```

4.4.10 Sine Amplitude Accuracy

Purpose: Verify accuracy and linearity of amplitude control and gain setting components over the range covered by the 0 dB output attenuator. This range is 7.5 Vp to 2.51 Vp.

Spec: ± 1% of setting

Setup: Select function sine, 600 Hz. Measure voltage at Arb Out terminated into 50 Ohm. Termination should be accurate to at least 0.1%.

Programmed Vp)	Programmed (Vrms)	Spec Min (Vrms)	Spec Max (Vrms)
7.50	5.303	5.250	5.356
5.00	3.536	3.500	3.571
2.51	1.775	1.757	1.793

Commands:

```
*rst
output on
freq 600
volt 7.5
volt 5.0
volt 2.51
```

4.4.11 Square Amplitude Accuracy

Purpose: Verify accuracy of square function gain setting components over the range covered by the 0 dB output attenuator. This range is 7.5 Vp to 2.51 Vp.

Spec: ± 1% of setting

Setup: Select function Square (HFSQ), 600 Hz. Measure voltage at Arb Out terminated into 50 Ohm. Termination should be accurate to at least 0.1%.

Programmed (Vp)	Spec Min (Vrms)	Spec Max (Vrms)
7.50	7.425	7.575
5.00	4.950	5.050
2.51	2.485	2.535

Commands:

```
*rst
func hfsq
freq 600
output on
volt 7.5
volt 5.0
volt 2.51
```

4.4.12 Attenuator Accuracy

Purpose: Verify accuracy of output attenuators.

Spec: ±1 % of setting.

Setup: Select function sine, 600 Hz. Measure voltage at Arb Out terminated into 50 Ohm. Termination should be accurate to at least 0.1%.

Set Value	Spec Min	Spec Max
2.000 V	1.400 Vrms	1.428 Vrms
1.000 V	0.7000 Vrms	0.7142 Vrms
0.300 V	0.2100 Vrms	0.2142 Vrms

Commands:

```
*rst
output on
freq 600
volt 2
volt 1
volt .3
```

4.4.13 DC Offset Accuracy

Purpose: Verify the DC accuracy and offset

Spec: ± 1% of setting ± 5 mV

Setup: Select function DC, amplitude 0V. Measure voltage at Arb Out terminated into 50 Ohm. Termination must be accurate to at least 0.1%.

Programmed (Vdc)	Spec Min (Vdc)	Spec Max (Vdc)
7.500	7.420	7.580
0.000	-0.005	0.005
-7.500	-7.580	-7.420

Commands:

```
*rst
output on
func dc
volt:offset 7.5
volt:offset 0
volt:offset -7.5
```

4.4.14 Frequency Response

Purpose: Verify the amplitude verses frequency specification

Spec: ± 2% for frequencies < 5MHz, ± 5% for frequencies ≥ 5MHz

Setup: Select function sine, 1kHz, 15Vpp. Measure voltage at Arb Out terminated into 50 Ohm. Note 1kHz amplitude. Record amplitude variation at programmed frequencies relative to 1kHz amplitude.

Freq	Spec Relative Ampl
10 kHz	± 2%
4 MHz	± 2%
5 MHz	± 5 %
20 MHz	± 5 %

Commands:

```
*rst
output on
volt 7.5
freq 1e3     First measurement
freq 10e3    Next measurement
freq 4e6     Next measurement
freq 5e6     Next measurement
freq 20e6    Last measurement
```

4.4.15 Square Waveform Quality

Purpose: Verify Rise/Fall times and aberrations are within specification.

Spec: Rise/Fall Time ≤ 9.5 nS for amplitudes > 10Vpp, ≤ 9 ns for amplitude ≤ 10 Vpp.Aberrations ≤ (5% ± 5mV) of peak to peak amplitude.

Setup: Select function Square (HFSQ), 10 MHz. Program the following amplitudes and verify rise/fall and aberrations with the Arb Out terminated into 50 Ohm.

Set Value	Spec Rise/Fall	Spec Aberrations
15 Vpp	≤ 9.5 nS	≤ 755 mVpp

Commands:

```
*rst
output on
func hfsq
freq 10e6
volt 7.5
```

4.4.16 Squarewave Duty Cycle

Purpose: Verify squarewave symmetry verses frequency.

Spec: 50 % ± 0.1 % for frequencies < 10 MHz, 50 % ± 2 % for frequencies ≥ 10 MHz

Setup: Select function Square (HFSQ), 10Vpp. Program the following frequencies and verify duty cycle with the Arb Out terminated into 50 Ohm

Set Value	Spec Min	Spec Max
1 kHz	49.0 %	51.0 %

Commands:

```
*rst
output on
func hfsq
volt 5
```

4.4.17 Sync Marker Output

Purpose: Verify performance of Sync Marker Output circuits.

Setup: Select function sine, 10 MHz. Terminate Sync Marker Out into 50 Ohm. Verify a valid TTL Level.

Commands:

```
*rst
freq 10e6
marler:sync on
func user
mark:position:point jack,0
```

4.4.18 Position Marker Output

Purpose: Verify performance of Position Marker Output circuits.

Setup: Define 5 point waveform, 50 MHz sample rate. Set position marker at address 0 of waveform. Terminate Position Marker Out into 50 Ohm. Verify a valid TTL Level.

Commands:

```
*rst
trace:define
func:user
mark:position:point:jack,0
```

4.4.19 Clock Output

Purpose: Verify performance of Clock Output circuits.

Setup: Set sample rate to 25 MHz. Turn on Clock Output BNC with the Sample clock as the source. Terminate Clock Out into 50 Ohm. Verify a valid TTL Level.

Commands:

```
*rst
outp:clock:freq 25e6
```

4.4.20 Clock Input

Purpose: Verify performance of clock input circuits.

Setup: Define 5 point waveform, clock source external. Apply a TTL level (0.8V to 2.1V) clock to Arb Channel Clock Input. Clock Input should be properly terminated to ensure good waveform quality. Set external clock frequency to 5 MHz. Terminate Sync Marker Out into 50 Ohm and verify it's frequency = external clock frequency divided by 5 (1MHz). (Pass/Fail)

Commands:

```
*rst
trace:define bill,5
trace:data bill,hfsq
func user
outp on
rosc:sour ext
```

4.4.21 Frequency Sweep

Purpose: Verify the functionality of the sweep.

Setup: Setup the 1395 for a triggered sweep, 100

sec sweep time, bus or manual trigger source. Terminate Arb Out into 50 Ohm. Monitor frequency of signal on Arb Out BNC.

Verify frequency = start frequency. (Pass/Fail)

Initiate trigger. Measure frequency after 10 sec. Verify frequency > start frequency. (Pass/Fail)

Commands:

```
*rst
output on
freq:mode sweep
sweep:time 10
sweep:mode tres
*trg
```

4.5 ALIGNMENT PROCEDURE

The procedure given in paragraph 4.5.5 requires that the model 1395 be part of a VXI system as described in table 4-1. The model 1395 will be installed on a C-size VXI extender card and controlled by the Resource Manager. 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 1395 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 1395 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.

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	Hewlet Packard	5334B

Overview

The calibration of the Model 1395 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.

3) Set amplitude to 7.5 Vp and verify rise/fall times < 9 ns and aberrations < 455 mV. If out of tolerance, readjust R180 and R198 then verify performance at 5 Vp to the acceptance test limits.

4.5.5.3 SUMBUS Driver Zero

The DC offset of the SUMBUS driver is adjusted with potentiometer R428. This adjustment is used to set the DC offset of the SUMBUS driver to 0. During this adjustment the SUMBUS driver is isolated from the rest of the circuits by grounding its input from the preamplifier and routing its output to the SUMBUS test point across a known 25Ω load. To make this 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:

SUMBUS input	Off
SUMBUS output	Off
Remote Cmds:	OUTP:SUMB OFF
	SUMB OFF

2) While measuring the DC voltage at the SUMBUS test point (TP29) adjust potentiometer R428 until the voltage is less than 1 mV.

4.5.5.4 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 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 Cmds:

   Remote query returns value of zero if self calibration was successful. A non-zero result indicates an self calibration failure.

4.5.5.5 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 1395 and stored in non-volatile calibration memory.

1) Configure the module as follows:

Function	Sine
Frequency	1 MHz
Amplitude	5 Vp
Modulation Mode	SCM
Remote Cmds:	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 Cmds: CAL:DATA:SCMZ?

4) Program a new constant and measure signal level. Iterate until amplitude is < 75mVpp.

   Remote Cmd: 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 Cmd: CAL:DATA:STORE

4.5.5.6 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 1395. The Model 1395 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 Cmds: 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 it's error is > 2 %. If error is > 2% send the appropriate correction to the Model 1395.

   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 Cmds: FREQ 7E6
   CAL:DATA:AMPL:AFC 1,7E6,(correction factor)

4) Measure the amplitude error at 13 MHz relative to 50 kHz. Send the appropriate correction to the Model 1395.

   Remote Cmds: FREQ 13E6
   CAL:DATA:AMPL:AFC 2,13E6,(correction factor)

5) Measure the amplitude error at 17 MHz relative to 50 kHz. Send the appropriate correction to the Model 1395.

   Remote Cmds: FREQ 17E6
   CAL:DATA:AMPL:AFC 3,17E6,(correction factor)

6) Measure the amplitude error at 19 MHz relative to 50 kHz. Send the appropriate correction to the Model 1395.

   Remote Cmds: FREQ 19E6
   CAL:DATA:AMPL:AFC 4,19E6,(correction factor)

7) Measure the amplitude error at 20 MHz relative to 50 kHz. Send the appropriate correction to the Model 1395.

   Remote Cmds: FREQ 20E6
   CAL:DATA:AMPL:AFC

5,20E6,(correction factor)

8) Store calibration constants. Turn calibration state on.

Remote Cmds: CAL:DATA:STORE
CAL:STATE ON

This completes the Calibration Procedure

5.1 DRAWINGS

The following schematics and assembly drawings are in the arrangement shown below.

5.2 ERRATA

Under Wavetek's product improvement program, the latest electronic designs and circuits are incorporated into each Wavetek instrument as quickly as development and testing permit. Because of the time needed to compose and print instruction manuals, it is not always possible to include the most recent changes in the initial printing. Whenever this occurs, errata pages are prepared to summarize the changes made and are inserted inside the shipping carton with this manual. If no such pages exist, the manual is correct as printed.

5.3 ORDERING PARTS

When ordering spare parts, please specify part number, circuit reference, board, serial number of unit and, if applicable, the function performed.

Note

An assembly drawing number is not necessarily the assembly part number. However, the assembly parts list number is the assembly part number.

DRAWING	DRAWING NUMBER
Outline Drawing	0002-00-0699
Instrument Schematic	1004-00-0699
Instrument Assembly Drawing	1001-00-0699
Instrument Parts List (Standard)	1000-00-0699-02
Instrument Parts List (w/option 001)	1000-00-0699-12
50 MHz VXI ARB Main Board Schematic	1104-00-3628
50 MHz VXI ARB Main Board Assembly	1101-00-3628
50 MHz VXI ARB Main Board Parts List	1100-00-3628-02
ARB Engine Board Schematic	1104-00-3674
ARB Engine Board Assembly Drawing	1101-00-3674
ARB Engine Board Parts List (Standard)	1100-00-3674-01
ARB Engine Board Parts List (w/option 001)	1100-00-3674-11
Spares Kit Parts List	1200-00-3629

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R178 R194 R260 R352	RES,MF,1/4W,1%,182 OHM	RN60D1820F	DALE	0587-01820	4
R275 R276 R277 R353	RES,MF,1/4W,5%,13 OHM	CCF070130J	DALE	0595-00130	4
R361	RES,MF,1/4W,5%,51 OHM	CCF070510J	DALE	0595-00510	1
R293 R296	RES,MFLM 115 OHM 1% 1/8W T0	RN55C1150F	UNCEM	110.1150	2
R288 R291	RES,MFLM 169 OHM 1% 1/10W T2	RN55C1690F	UNCEM	110.1691	2
REF	A/D 50 MHZ VXI ARB MAIN	1101-00-3628	WVTK	1101-00-3628	1
REF	SCHEMATIC,ARB MAIN BD	1104-00-3628	WVTK	1104-00-3628	1
R292 R297	RES,MFLM 6.34K 1% 1/10W T2	RN55C6341F	UMCEM	111.6341	2
R203	RES,MFLM 70.27K .1% 1/8W T0	HA55 100PPM	HYMEG	112.7028	1
RN5 RN6	RES,NETWORK, 10K OHM 1% X8	698-3-R10K-F	BECK	128.3103	2
C53 C59 C61	CAP,CER,.001 MF,+-10%,50V,X7R,.1" LS	SR155C102KAA	AVX	1500-01-0207	3
C122 C129 C130 C133 C314 C317 C54 C57	CAP,CER,.01MF,+-10%,50V,X7R,.1"LS	SR155C103KAA	AVX	1500-01-0307	8
C10 C100 C103 C104 C107 C110 C112 C113 C114 C115 C116 C117 C118 C119 C12 C120 C124 C125 C127 C131 C134 C135 C136 C139 C14 C140 C141 C142 C143 C144 C145 C146 C148 C149 C15 C150 C152 C153 C155 C157 C158 C159 C16 C162 C163 C165 C167 C17 C170 C171 C172 C173 C174 C175 C176 C177 C178 C179 C18 C180 C181 C182 C183 C184 C185 C186 C187 C189 C19 C190 C191 C192 C193 C194 C195 C196 C20 C200 C202 C203 C204 C205 C206 C208 C209 C21 C210 C211 C212 C213 C214 C216 C217 C22 C23 C24 C25 C26 C27 C28 C29 C301 C308 C309 C31 C310 C311 C312 C318 C32 C33 C337 C34 C341 C350 C351 C352 C353 C357 C358 C359 C360 C361 C362 C363 C365 C4 C40 C41 C42 C45 C46 C52 C55 C58 C6 C60 C65 C66 C70 C72 C73 C74 C8 C81 C82 C83 C84 C85 C86 C97 C98 C99	CAP,CER,.1 MF,50V,X7R,+-10%,.1" LS	SR205C104KAA	AVX	1500-01-0415	153
C69 C95	CAP,CER,COG, 50V, 5%, 120 PF, .1 LS	SR155A121JAA	AVX	1500-01-2101	2
C199 C94	CAP,CER,COG, 50V, 5%, 150 PF, .1 LS	SR155A151JAA	AVX	1500-01-5101	2
C80	CAP,CER,150PF,1KV	DD-151	CRL	1500-01-5111	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
C156 C315	CAP,CER DISK,1.5PF,1KV,TEMP COMP	NCD1.5C1KVNP0	NIC	1500-01-5507	2
C88 C90	CAP,CER,COG, 50V, 5%, 18 PF, .1 LS	SR155A180JAA	AVX	1500-01-8001	2
C87	CAP,CER,COG, 50V, 5%, 180 PF, .1 LS	SR155A181JAA	AVX	1500-01-8101	1
C35 C44 C89 C92	CAP,CER,COG, 50V, 5%, 220 PF, .1 LS	SR155A221JAA	AVX	1500-02-2101	4
C218	CAP CER MON 2.7PF 50V	CCD2R7DNP0	ARCO	1500-02-7505	1
C96	CAP,CER,COG, 50V, 5%, 390 PF, .1 LS	SR155A391JAA	AVX	1500-03-9102	1
C123 C93	CAP,CER,COG, 50V, 5%, 68PF, .1 LS	SR155A680JAA	AVX	1500-06-8002	2
C316 C91	CAP,CER,COG, 50V, 82PF, .1 LS	SR155A820JAA	AVX	1500-08-2001	2
C11 C13 C3 C5 C63 C7 C9	CAP,ELECT,100MF,35V RADIAL LEAD,SP .20	NRE101M35V10X12.5(OBS)	NIC	1500-31-0102	7
C106 C121 C126 C132 C137 C147 C160 C168 C169 C188 C197 C198 C71	CAP,ELECT,22MF,25V,RADIAL	SRA25VB22RM6X7LL	UNCON	1500-32-2002	13
C62 C64	CAP,ALUM ELECT,470MF,20%,16V,RADAIL .2" SP,10MM DIAx12.5MM HT	KME16VB471M10X12LL	UNCON	1500-34-7114	2
C166 C50	CAP,MET POLY,.001MF,100V	168102K100A	WEST	1500-41-0214	2
C201 C37	CAP,MET POLYS,.1MF,10%,63V,.1 "LS	MKS02-0.1MF-10%-63V	WIMA	1500-41-0418	2
C36 C38 C49	CAP,MET POLYS,.33 MF,+-20%,50V,.1"LS	MKS02-.33MF-20%-50V	WIMA	1500-41-0421	3
C39 C43	CAP,MET POLYS,0.01MF,+-20%,63V,.1IN LS	MKS02-0.01MF,20%-63V	WIMA	1500-41-0422	2
C48 C51	CAP,MET,POLY, .033MF 100V 5% .2 LS	168/.033/J/63/A(OBS)	WEST	1500-43-3304	2
C215 C56	CAP,TANT,22MF,15V	196D226X9015KA1(OBS)	SPRAG	1500-72-2601	2
C319 C322 C325 C327	CAP,TANT, 6.8MF, 20V, 20%, SMD EIA 6032 CASE	PCT6.8/20CM	NEMCO	1500-76-8512	4
C330	CAP, SMD CER NP0 100PF/50V	C1206C101J2GAC	KEMET	1500-80-0001	1
C320 C321 C323 C324 C326 C328 C329 C331 C332 C333 C335 C338 C339 C354 C356 C364	CAP,CER, 0.1MF, 50V,Z5U, +80-20%,SMD 1206	C1206C104Z5UAC	KEMET	1500-81-0401	16
C334	CAP,SMD 5% 100V 15PF	11C1206C0G150J100NB	SPRAG	1500-81-5001	1
C355	CAP,CER, 180 PF, 5%,NP0,SMD 1206 CASE	12061A181JAT050R OR B	AVX	1500-81-8101	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
C336	CAP,CER, 8.2PF, 5%, 50V,NPO,SMD 1206 CASE	C8R2J1206CFLB OR T	PHLIP	1500-88-2901	1
1	PCB,50MHZ VXI ARB MAIN REF: SPEC 0008-00-0455 REV E	1700-00-3628	WVTK	1700-00-3628	1
L5	INDUCTOR,99 NH,5%,Q66	132-07	COILC	1800-00-0050	1
L3 L4	INDUCTOR,FIXED, 345 nH	132-15	COILC	1800-00-0056	2
L2 L6 L7	INDUCTOR,FIXED, 475 nH	132-17	COILC	1800-00-0057	3
L1	INDUCTOR, 117 nH, Q68, AT 25 MHZ FREQ	132-08	COILC	1800-00-0065	1
J10 J11 J12 J13 J8 J9	CONN, BNC(PC)	227161-1	AMP	2100-01-0019	6
J3	HEAD, 26 PIN,DUAL ROW, 0.025 SQ PIN,W/SHRD	TST-113-01-L-D	SAM	2100-02-0308	1
J1 J2	HEAD, 40 PIN,DUAL ROW, 0.025 SQ PIN,W/SHRD	TST-120-01-L-D	SAM	2100-02-0309	2
J4 J5	SOCKET,40 PIN,ST,LOW INSERT	SSW-120-02-G-D	SAM	2100-03-0000	2
J6	SOCKET,20 PIN,ST,LOW INSERT	SSW-110-02-G-D	SAM	2100-03-0001	1
J7	SOCKET,36 PIN,ST,LOW INSERT	SSW-118-02-G-D	SAM	2100-03-0006	1
TP10 TP11 TP15 TP27 TP3 TP30 TP5 TP9	TEST POINT,BLK,PC	TP-104-01-00	COMPO	2100-04-0054	8
TP1 TP16 TP17 TP18 TP19 TP2 TP20 TP21 TP22 TP23 TP24 TP25 TP26 TP29 TP4 TP6 TP7 TP8	TEST POINT,RED,PC	TP-104-01-02	COMPO	2100-04-0055	18
4	TRANSIPAD	503-075	BIVAR	2800-11-0003	16
3	HEATSINK,TO-5 PKG	7-120BA	IERC	2800-11-0042	8
2	HOLE PLUG,BINDER HEAD,NTRAL NYLON	207-120241-03-0101	FASTX	2800-35-0009	4
K1 K10 K11 K16	RELAY,2 FORMC,5V,DIP 14	DF2E-DC5V	AROMT	4500-00-0033	4
K12 K13 K14 K15 K2 K20 K23 K4 K5 K7 K8	RELAY,1 FORMC,5V,.312H,.296W	HD1E-M-DC5V	AROMT	4500-00-0034	11
K17 K18 K21 K22 K3	RELAY, 2 FORM C, 5V, LOW PROFILE	TQ2E-5V	AROMT	4500-00-0038	5
R428 R509	POT,TOP TRIM,20T,1K	68WR1K	BECK	4609-90-0005	2
R180 R198	POT,TOP TRIM,20T,20 OHM	68WR20	BECK	4609-90-0024	2
R205 R206 R209 R210	RES,C,1/2W,5%,10	RC-1/2-10J	STKPL	4700-25-0100	4
R199 R200 R212 R213	RES,C,1/2W,5%,22	RC-1/2-220J	STKPL	4700-25-0220	4
R155 R160	RES,C,1/2W,5%,2.7	RC-1/2-2R7J	STKPL	4700-25-0279	2
R357 R358 R359 R360	RES,C,1/2W,5%,8.2	RC-1/2-8R2J	STKPL	4700-25-0829	4

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R175 R187	RES,C,1/2W,5%,1K	RC-1/2-102J	STKPL	4700-25-1001	2
R252 R253	RES,CC,1/2W,5%,4.7	RC20GF4R7J	AB	4700-99-0088	2
R147 R148	RES,MF,1/8W,.1%,100	RN55E1000BJ	PHLP	4701-02-1000	2
R149 R151 R6	RES,MF,1/8W,.1%,1K	RN55E-1001B	MEPCO	4701-02-1001	3
R152 R153	RES,MFLM,1/8W,.1%,1.1 K	5033RE1101B	MEPCO	4701-02-1101	2
R272	RES,MFLM, 113 OHM, 0.1%, 1/8W, 100 PPM/C	RN55D1130B	BUYOP	4701-02-1130	1
R273	RES,MFLM, 115 OHM, 1/8W, 0.1%, 100 PPM/C	RN55D1150B	BUYOP	4701-02-1150	1
R135 R144	RES, 121 OHM, 0.1%, 1/8W, 100PPM/C	RN55D1210B	BUYOP	4701-02-1210	2
R218 R219	RES,MFLM,1/8W,.1%,301	5033RE3010B	MEPCO	4701-02-3010	2
R267	RES,MFLM, 402 OHM, 0.1%, 1/8W, 100 PPM/C	RN55D4020B	BUYOP	4701-02-4020	1
R136 R143	RES, 464 OHM, 1/8W, 0.1%, 100PPM/C	RN55D4640B	BUYOP	4701-02-4640	2
R204 R4	RES,MF,1/8W,.1%,9K	5023ZE9K00B	MEPCO	4701-02-9001	2
R170 R171 R259 R306 R433 R75 R79 R80	RES,MF,1/8W,1%,100	RN55D-1000F	TRW	4701-03-1000	8
R105 R11 R119 R12 R121 R127 R172 R185 R186 R23 R24 R26 R302 R303 R407 R500 R53 R59	RES,MF,1/8W,1%,1K	RN55D-1001F	TRW	4701-03-1001	18
R1 R132N R166 R2 R271 R274 R7	RES,MF,1/8W,1%,10K	RN55D-1002F	TRW	4701-03-1002	7
R159 R164 R167 R201 R251 R255 R257 R300 R446 R8 R9	RES,MF,1/8W,1%,100K	RN55D-1003F	TRW	4701-03-1003	11
R120 R145 R146 R158 R161 R162 R179 R181 R182 R189 R190 R192 R195 R196 R197 R231 R27 R279 R28 R280 R281 R29 R298 R299 R30 R304 R350 R351 R444 R445 R54 R62 R71 R72	RES,MF,1/8W,1%,10	5043ED10R100F	MEPCO	4701-03-1009	34
R21	RES,MF,1/8W,1%,110K	RN55D-1103F	TRW	4701-03-1103	1
R100 R118 R124 R248 R45 R46 R48 R50 R96	RES,MF,1/8W,1%,121	RN55D-1210F	TRW	4701-03-1210	9
R85 R86	RES,MF,1/8W,1%,124	RN55D-1240F	TRW	4701-03-1240	2
R138	RES,MF,1/8W,1%,13K	RN55D-1302F	TRW	4701-03-1302	1
R17 R221	RES,MF,1/8W,1%,1.5K	RN55D-1501F	TRW	4701-03-1501	2
R128 R129	RES,MF,1/8W,1%,150K	RN55D-1503F	TRW	4701-03-1503	2
R104 R183	RES,MF,1/8W,1%,15	RN55D-15R0F	TRW	4701-03-1509	2
R130 R131 R14	RES,MF,1/8W,1%,15.8K	RN55D-1582F	TRW	4701-03-1582	3
R13	RES,MF,1/8W,1%,16.5K	RN55D-1652F	TRW	4701-03-1652	1
R22	RES,MF,1/8W,1%,169K	RN55D-1693F	TRW	4701-03-1693	1
R19	RES,MF,1/8W,1%,182	RN55D-1820F	TRW	4701-03-1820	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R10 R102 R103 R137 R169 R290 R447 R448	RES,MF,1/8W,1%,200	RN55D-2000F	TRW	4701-03-2000	8
R301 R36 R69 R93	RES,MF,1/8W,1%,2K	RN55D-2001F	TRW	4701-03-2001	4
R15	RES,MF,1/8W,1%,20K	RN55D-2002F	TRW	4701-03-2002	1
R16	RES,MF,1/8W,1%,21.5K	RN55D-2152F	TRW	4701-03-2152	1
R294	RES,MF,1/8W,1%,23.7K	RN55D-2372F	TRW	4701-03-2372	1
R77 R99	RES,MF,1/8W,1%,249	RN55D-2490F	TRW	4701-03-2490	2
R173 R188	RES,MF,1/8W,1%,2.49K	RN55D-2491F	TRW	4701-03-2491	2
R174 R229 R266	RES,MF,1/8W,1%,24.9K	RN55D-2492F	TRW	4701-03-2492	3
R176	RES,MF,1/8W,1%,24.9	RN55D-24R9F	TRW	4701-03-2499	1
R285 R287	RES,MF,1/8W,1%,301	RN55D-3010F	TRW	4701-03-3010	2
R262 R264 R289 R91	RES,MF,1/8W,1%,3.01K	RN55D-3011F	TRW	4701-03-3011	4
R510	RES,MF,1/8W,1%,30.1K	RN55D-3012F	TRW	4701-03-3012	1
R356	RES,MF,1/8W,1%,30.1	RN55D-30R1F	TRW	4701-03-3019	1
R134 R142 R87	RES,MF,1/8W,1%,332	RN55D-3320F	TRW	4701-03-3320	3
R256	RES,MF,1/8W,1%,33.2K	RN55D-3322F	TRW	4701-03-3322	1
R20	RES,MF,1/8W,1%,37.4K	RN55D-3742F	TRW	4701-03-3742	1
R168 R501 R81	RES,MF,1/8W,1%,392	RN55D-3920F	TRW	4701-03-3920	3
R263	RES,MF,1/8W,1%,4.02K	RN55D-4021F	TRW	4701-03-4021	1
R150 R154	RES,MF,1/8W,1%,432	RN55D-4320F	TRW	4701-03-4320	2
R268	RES, 45.3K OHM, 1%, 1/8W, 100 PPM/C	RN55D4532F	BUYOP	4701-03-4532	1
R25 R31 R32	RES,MF,1/8W,1%,464	RN55D-4640F	TRW	4701-03-4640	3
R18	RES,MF,1/8W,1%,4.64K	RN55D-4641F	TRW	4701-03-4641	1
R76 R78	RES,MF,1/8,1%,499	RN55D-4990F	TRW	4701-03-4990	2
R354 R355 R513	RES,MF,1/8W,1%,4.99K	RN55D-4991F	TRW	4701-03-4991	3
R37 R70	RES,MF,1/8W,1%,499K	RN55D-4993F	TRW	4701-03-4993	2
R141 R246 R247 R249	RES,MF,1/8W,1%,49.9	RN55D-49R9F	CORNG	4701-03-4999	4
R98	RES,MF,1/8W,1%,523	RN55D-5230F	TRW	4701-03-5230	1
R90	RES,MF,1/8W,1%,56.2	RN55D-56R2F	TRW	4701-03-5629	1
R82 R89	RES,MF,1/8W,1%,576	RN55D-5760F	TRW	4701-03-5760	2
R511	RES,MF,1/8W,1%,57.6K	RN55D-5762F	TRW	4701-03-5762	1
R5	RES,MF,1/8W,1%,6.04K	RN55D-6041F	MEPCO	4701-03-6041	1
R133 R3	RES,MF,1/8W,1%,7.15K	RN55D-7151F	TRW	4701-03-7151	2
R163 R165	RES,MF,1/8W,1%,7.32K	RN55D-7321F	WVTK	4701-03-7321	2
R258	RES,MF,1/8W,1%,750	RN55D-7500F	TRW	4701-03-7500	1
R265	RES,MF,1/8W,1%,75K	RN55D-7502F	CORNG	4701-03-7502	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R286	RES,MFLM,1/8W,1%,75	5033RD75R0F	MEPCO	4701-03-7509	1
R295	RES,MF,1/8W,1%,825	RN55D-8250F	TRW	4701-03-8250	1
R101 R117 R125 R40 R41 R44 R49 R97	RES,MF,1/8W,1%,82.5	RN55D-82R5F	TRW	4701-03-8259	8
R223 R225	RES,MFLM, 165 OHM, 0.1%, 1/4W, 100 PPM/C	RN60D1650B	BUYOP	4701-04-1650	2
R222 R224	RES,MFLM, 169 OHM, 0.1%, 1/4W, 100 PPM/C	RN60D1690B	BUYOP	4701-04-1690	2
R216 R217	RES, 301 OHM, 0.1%, 1/4W, 100 PPM/C	RN60D3010B	BUYOP	4701-04-3010	2
R214 R215	RES, MFLM, 75 OHM, 0.1%, 1/4W, 100 PPM/C	RN60D75R0B	BUYOP	4701-04-7509	2
R269	RES,MFLM, 113 OHM, 0.1%, 1/2W, 100 PPM/C	RN65D1130B	BUYOP	4701-06-1130	1
R270	RES,MFLM, 115 OHM, 0.1%, 1/2W, 100 PPM/C	RN65D1150B	BUYOP	4701-06-1150	1
R184 R250	RES,MF, 1/4W, 1%, 2K OHM	RN60D2001F	BUYOP	4701-13-2001	2
R106 R107 R109 R122 R123 R108 R42 R43 R60 R61	RES,MF,1/2W,1%,100	RN65D-1000F	TRW	4701-23-1000	10
R207 R208 R220	RES, MF,1W,.25%, T2 100	5053RC100R0C	MEPCO	4701-38-1000	3
R429 R434 R504	RES, 100 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A1000FLR	PHLIP	4703-13-1000	3
R427 R432 R508	1/8W, 1%, 200 PPM, 100 KOHM,SMD 1206 CASE	9C12063A1003FLR	PHCOM	4703-13-1003	3
R442 R443	RES, 200PPM,1/8W, 1%,10 OHM,SMD, 1206 1206 CASE	9C12063A10R0FLR	MEPCO	4703-13-1009	2
R420	RES, 12.4 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A12R4FLR	PHLIP	4703-13-1249	1
R422 R423 R431 R505	RES, 150 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A1500FLR	PHLIP	4703-13-1500	4
R436	RES, 165 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A1650FLR	PHLIP	4703-13-1650	1
R414 R416	RES, 210 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A2100FLR	PHLIP	4703-13-2100	2
R421 R435 R437 R438 R439 R441	RES, 200PPM, 1%, 1/8W, 249 OHM, 1206,SMD	9C12063A2490FLR	MEPCO	4703-13-2490	6
R419	RES, 200 PPM, 1%, 1/8W, 28.7,SMD, 1206	9C12063A28R7FLR	MEPCO	4703-13-2879	1
R417	RES, 324 OHM, 1%, 1/8W, 200 PPM/C, SMD 1206 CASE	9C12063A3240FLR	PHLIP	4703-13-3240	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R502	RES, 3.74K OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A3741FLR	PHLIP	4703-13-3741	1
R411 R430	RES, 200 PPM, 1%, 1/8W, 38.3 OHM,SMD 1206 CASE	9C12063A38R3FLR	MEPCO	4703-13-3839	2
R410 R506 R507	RES,SMD,200PPM,1%,1/8W,49.9ohm	CRCW1206-49.9FT	DALE	4703-13-4999	3
R440	RES, 549 OHM, 1%, 1/8W, 200 PPM/C,SMD 1206 CASE	9C12063A5490FLR	PHLIP	4703-13-5490	1
R503	RES, 750 OHM, 1%, 1/8W, 200 PPM,SMD 1206	9C12063A7500LR	PHLIP	4703-13-7500	1
R512	RES, 200PPM,75 OHM, 1%, 1/8W, 1206	9C12063A75R0FLR	MEPCO	4703-13-7509	1
R412 R413	RES, 200 PPM/C, 1%, 1/8W, 909 OHM,SMD 1206 CASE	9C12063A9090FLR	PHLIP	4703-13-9090	2
R514	RES, 90.9K OHM, 1%, 1/8W, 200 PPM,SMD 1206	9C12063A9092LR	PHLIP	4703-13-9092	1
R426	RES, 121 OHM, 1%, 200 PPM/C, 1/2W, SMD 1812 CASE	MC1812 121 OHM 1% T	RCD	4703-16-1210	1
R424	RES, 150 OHM, 1%, 200 PPM/C, 1W, SMD, 2512 CASE	MC2512 150 OHM 1% T	RCD	4703-18-1500	1
R418 R425	RES, 88.70 OHM, 1%, 200 PPM/C, 2W,SMD 4020 CASE	MC4020 88.7 OHM 1% T	RCD	4703-30-8879	2
RN12 RN4	RES NETWORK 10K 2% 10PIN SIP BUSS	4310R-101-103	BOURN	4770-00-0008	2
RN10 RN11 RN9	RES NETWORK 470 10PIN SIP BUSS	4310R-101-471	BOURN	4770-00-0009	3
RN1	RES NETWORK 2.2K 2% 10PIN SIP BUSS	4310R-101-222	BOURN	4770-00-0011	1
RN2	RES NETWORK 1K 2W 16PIN DIP	4116R-001-102	BOURN	4770-00-0019	1
RN3	RES NETWORK 1K 2% 10PIN SIP BUSS	4310R-101-102	BOURN	4770-00-0022	1
RN13 RN7 RN8	RES NETWORK, SINGLE 10 PIN, 50 OHM	4310R-101-500	BOURN	4770-00-0058	3
CR1 CR72 CR73	DIODE,ZENER,2.7V,.5W, GLASS	1N4371	MOT	4801-01-4371	3
CR24 CR25	DIODE, ZENER 10V, GLASS SILICON, 1W	1N4740A	MOT	4801-01-4740	2
CR63	DIODE,ZENER, 6.8V,MLL34 PKG,SMD	MLL5235B	MOT	4801-01-5235	1
CR12 CR13 CR2 CR20 CR21 CR22 CR26 CR27 CR51 CR52 CR70 CR71	DIODE, HIGH CONDUCTANCE, ULTRA FAST	1N5282	FAIR	4801-01-5282	12

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
CR23 CR28 CR29 CR30	DIODE,1N4002 GEN PURPOSE RECT. 100V,1A	1N4002	FAIR	4801-02-0001	4
CR64	DIODE,ZENER, 3.3V, 5%, 500mW,SMD,LEADLESS	MLL5226B	MOT	4801-02-5226	1
CR67 CR68	DIODE,ZENER, 5.1V, 5%, 500mW, SMD,LEADLESS	MLL5231B	MOT	4801-02-5231	2
CR65 CR66 CR69	DIODE,ZENER, 9.1V, 5%, 500MW, SMD,LEADLESS	MLL5239B	MOT	4801-02-5239	3
CR10 CR15 CR16 CR17 CR18 CR31 CR32 CR33 CR34 CR7 CR8 CR9	DIODE 5082-2811 SCHOTTKY,15V,20MA	5082-2811	HP	4809-02-2811	12
CR3 CR4 CR5 CR6	DIODE,BB909B	BB909B	PHLIP	4899-00-0040	4
DS1	LED,AMBER,RECT BAR	LTL-3251A	LITE	4899-00-0056	1
DS2 DS4	LED,GREEN,PC MT.,INT HOLDER	550-0206	DIALT	4899-00-0066	2
DS5 DS6	LED,YELLOW,PC MOUNT,INT. HOLDER	550-0306	DIALT	4899-00-0067	2
DS3	LED,RED, PC MT., INT. HOLDER	550-0406	DIALT	4899-00-0068	1
Q12 Q4	TRANS 2N2219A NPN GENERAL PURPOSE TO-5	2N2219A	NSC	4901-02-2191	2
Q5 Q6	TRANS,SILICON,PLANAR, EPITAXIAL,NPN,TO-18	2N2369A	MOT	4901-02-3691	2
Q13 Q7 Q8	TRANS, NPN, TO-92	PN3563	NSC	4901-03-5630	3
Q10 Q11 Q17 Q18 Q2 Q3	TRANS	2N3866	MOT	4901-03-8660	6
Q19 Q20	TRANS	2N5160-18(OBS)	MOT	4901-05-1600	2
Q56	TRANS,PNP, HI FREQ,SMD,SO8 PKG	MRF5583	MOT	4901-05-5831	1
Q15 Q9	TRANS 2N5771 PNP SWITCH TO-92	2N5771	NSC	4901-05-7710	2
Q54 Q55	TRANS,NPN,HI FREQ,SMD,SO8 PKG	MRF5943	MOT	4901-05-9431	2
Q1	TRANS,FET P CHANNEL	VP0106N3	SUPER	4902-01-0601	1
Q14	TRANSISTOR,PNP,HI FREQ,MED PWR SELECTED	SRF2573	MOT	4902-02-5730	1
Q16	TRANSISTOR,NPN,HI FREQ, MED PWR SELECTED	4902-02-5740	WVTK	4902-02-5740	1
U62	OP-AMP,DUAL,JFET,SMD, SO-8	TL0-52CD	TI	7000-00-5201	1
U23 U26	OP AMP,ENHANCED JFET,PRECISION QUAD	TL054CN	TI	7000-00-5400	2
VR1 VR4	VOLT REGULATOR,3 TERMINAL ADJUSTABLE,POS,TO39	LM317H	NSC	7000-03-1710	2

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
VR2 VR5	VOLT REGULATOR, 3 TERMINAL ADJUSTABLE, NEGATIVE, T0-39	LM337 H	NATNL	7000-03-3701	2
U46	COMPARATORS, QUAD VOLTAGE	LM339N(REF ONLY)	HARIS	7000-03-3900	1
U63 U64	OP-AMP,HI SLEW RATE, CURNT FDBK,SMD,SO-8	CLC404AJE	COMLR	7000-04-0401	2
U61	OP-AMP,HI SLEW RATE, CURNT FEBK,SMD,SO-8	CLC409AJE	COMLR	7000-04-0901	1
U51	OP AMP,PRECISION,WIDEBAND,8-PIN DIP	OPA621KP	BURR	7000-06-2100	1
U50	MULTIPLIER,ANALOG,WIDEBAND	AD834JN	ANDEV	7000-08-3410	1
U18	VOLTAGE REFERENCE, 10V, 8 PIN DIP	LT1021CCN8-10	LINTE	7000-10-2101	1
U16 U17 U57	DARLINGTON ARRAYS, HIGH VOLTAGE, HIGH CURRENT	ULN2003A	SPRAG	7000-20-0300	3
U54 U55	OP,AMP CURRENT FEEDBACK,50mhz	EL2020CN	ELAN	7000-20-2000	2
U21	ADC, 12 BIT	ICL7109CPL	INTSL	7000-71-0900	1
VR3 VR6	REGULATOR, 3 TERMINAL NEG 5V	LM79L05ACZ	NSC	7000-79-0501	2
U22 U25	DAC, 12 BIT, 4 CHAN,SOIC PKG	MB88351PF	FUJI	7000-88-3511	2
U45	COMPARATOR,HIGH SPEED,ECL	AD96685BQ	AD	7000-96-6850	1
U49	OP AMP,HI SLEW RTE,WIDEBND,JFET,PRIME	MC34081AP	MOT	7003-40-8101	1
U24 U58	OP AMP,HI SLEW RTE,WIDEBND,JFET DUAL	MC34082P	MOT	7003-40-8200	2
U28	OP AMP,HI SLEW RTE,WIDEBND JFET	MC34084P	MOT	7003-40-8400	1
NONE	PLD,GAL16V8, 7 NS, 20 PIN DIP	GAL16V8B-7LP	LATT	8000-16-8007	2
U20	MUX/DEMUX, ANALOG	CD4051BE	RCA	8000-40-5100	1
U5	HEX INVERTER	SN74ALS04BN	TI	8000-74-0402	1
U39	GATE,INVER, SCHMITT TRIG, HEX,ACT-CMOS	MC74ACT14N	MOT	8000-74-1431	1
U11 U2 U3 U4	BUFF,OCTAL,NINV,TTL INP/CMOS OUTPUTS	MC74HCT541N	MOT	8000-74-5411	4
U12 U13 U14 U15 U19 U56 U59	SHFT REG,8-BIT,CMOS,OUTPUT CLR	SN74HC595N	TI	8000-74-5950	7
U33 U38 U53 U69	MECL 10KH HIGH-SPEED ECL	MC10H102P	MOT	8001-01-0201	4
U31 U34 U37 U47 U48	RECEIVER-TRIP LN ECL	MC10H116P	MOT	8001-01-1601	5

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
U35	TRANSLATOR,QUAD TTL-TO-MECL	MC10H124P	MOT	8001-01-2401	1
U36 U68	TRANSLATOR,QUAD TTL-TO-MECL	MC10H125P	MOT	8001-01-2501	2
U32	FLIP-FLOP, DUAL D, MAS/SL, ECL	MC10H131P	MOT	8001-01-3101	1
U60	MUXMECL, 10KH	MC10H164-P	MOT	8001-01-6400	1
U30	PLL, ECL	MC12018P	MOT	8001-20-1800	1
U10	PROM,EE, 2Kx8,CMOS, 200NS	XL28C16AP-200	EXEL	8002-81-6000	1
U66 U9	DECODER/DEMUX,3 TO 8,TTL	SN74F138N	TI	8007-41-3820	2
U65	MUX 8-INPUT	74F151PC	FAIR	8007-41-5120	1
U7 U8	TRANSCEIVER, OCTAL BUS W/3 STATE	SN74ALS645A-1N	TI	8007-46-4500	2
U27	PLL SERIES INPUT FREQ SYNTHESIZER	MC145158P(OBS)	MOT	8014-51-5800	1
NONE	GAL,90MA,30NS,24PIN DIP	GAL6001-30	LATT	8060-01-3000	1
U67	BUFFER,OCTAL,OPEN COLTR,64MA	N74F760N	SIG	8074-76-0001	1
U29	OSC EMT CUPLD, ECL	MC1648P	MOT	8100-16-4810	1
U42	GATE ARRAY,COUNTER,SYNCH,MULTI-MODE,32-BIT	CLA53047/BW/DP	PLESS	8700-00-0005	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R1 R12 R14 R2 R5 R6	RES,200 PPM,1/8W,5%,82 OHM,SMD,1206	9C12063A82R0JLR	MEPCO	4703-20-8205	6
RN1 RN2 RN4 RN5 RN7	RES NETWORK 470 10PIN SIP BUSS	4310R-101-471	BOURN	4770-00-0009	5
RN3 RN6	NTWK,RES, 47 OHM, 7 RES,ISOL,SMD	SOMC-1403-470G	DALE	4770-00-0070	2
U19	OP-AMP,HI SLEW RATE, CURNT FEBK,SMD,SO-8	CLC409AJE	COMLR	7000-04-0901	1
U17	RECEIVER,ECL,3 LINE,PLCC	MC10H116FN	MOT	8000-10-0116	1
U18	D/A,12 BIT,50MSPS,ECL INP,PLCC PKG	TDC1112R3C2	TRW	8000-11-1201	1
U4 U5	XCVR,OCTAL,HCTMOS,TTL I/O,SOIC PKG	MC74HCT245ADW	MOT	8000-74-2451	2
U3	GATE,QUAD,2-INP,SOIC PKG	SN74ALS32D	TI	8000-74-3251	1
U10 U11 U12 U8 U9	XLATOR,QUAD,TTL-TO-ME CL,PLCC PKG	MC10H124FN	MOT	8001-01-2402	5
U1 U16	XLATOR,QUAD,TTL-TO-EC L,PLCC PKG	MC10H125FN	MOT	8001-01-2502	2
U13 U14 U15	FLIP-FLOP,D-TYPE,HEX, ECL,PLCC PKG	MC10H176FN	MOT	8001-01-7602	3
U20 U21	SRAM,CMOS, 32K x 8, 15 NS, SOJ PKG	TC55B328J-15	TOSH	8007-12-5616	2
NONE	PLD,EECMOS,GAL16V8, 10 INP, 8 I/O 7.5 NS, 20 PIN PLCC	GAL16V8B-7LJ	LATT	8016-08-0750	1
U2	ASIC,ADDRESS GENE. AND SEQUENCER,AGS4804	8700-00-0007	WVTK	8700-00-0007	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
REF	A/D, ARB ENGINE BOARD	1101-00-3674	WVTK	1101-00-3674	1
REF	SCHEMATIC, ARB ENGINE BOARD	1104-00-3674	WVTK	1104-00-3674	1
C1 C3 C41 C43 C53	CAP,TANT,22 MF,16V,20%,SMD	TAJD226M016R	STANT	1500-72-2001	5
C10 C11 C12 C13 C14 C16 C17 C18 C19 C2 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 C33 C34 C35 C37 C38 C39 C4 C40 C42 C44 C45 C46 C47 C48 C49 C5 C50 C51 C52 C6 C7 C8 C9	CAP,CER, 0.1MF, 50V,Z5U, +80-20%,SMD 1206	C1206C104Z5UAC	KEMET	1500-81-0401	46
1	PCB, ARB ENGINE REF: SPEC 0008-00-0455 REV E	1700-00-3523-01	WVTK	1700-00-3523-01	1
DL1	DELAY LINE, 3.0 NS, SINGLE-IN-LINE PKG	EP123107	PCA	1800-00-0062	1
P1A P1B P2A P2B	CONN,HEADER,SNGL ROW,LW PROF,22 PIN	929870-01-22-10	3M	2100-02-0305	4
P3A P3B	SOCKET,BOARD MNT,LOW PROFILE,15 CONTACT	929870-01-15-10	3M	2100-03-0100	2
2	SKT,20PIN,PLCC SM	213-020-601	METHD	2100-03-0103	1
3	SKT, 84 PIN, PLCC, SMD	PLCC-84-T-A	SAM	2100-03-0115	1
TP2 TP6	TEST POINT,BLK,PC	TP-104-01-00	COMPO	2100-04-0054	2
TP1 TP3 TP4 TP5	TEST POINT,RED,PC	TP-104-01-02	COMPO	2100-04-0055	4
4	STANDOFF, SWAGE,6-32X1/8X1/4DIA	B-25000-632-105-22	GLOBE	2800-06-0071	3
FB1 FB2	BALUN CORE,FERRITE,82 OHMS	2743015111	FRITE	3100-00-0018	2
R9	RES,200PPM,1/8W,1%,1.5K OHM,SMD,1206	9C12063A1501FLR	MEPCO	4703-13-1501	1
R10	RES,200 PPM,1/8W,1%,15.8K,SMD ,1206	9C12063A1582FLR	MEPCO	4703-13-1582	1
R11	RES,200 PPM,1/8 W,1%,1.65K,SMD,1206	9C12063A1651FLR	MEPCO	4703-13-1651	1
R23	RES, 200PPM, 1%, 1/8W, 249 OHM, 1206,SMD	9C12063A2490FLR	MEPCO	4703-13-2490	1
R18 R19	RES,200 PPM,1/8W,1%,24.9 OHM,SMD,1206	9C12063A24R9FLR	MEPCO	4703-13-2499	2
R16 R17 R20 R21	RES,200 PPM, 1/8W,1%, 453 OHM,SMD,1206	9C12063A4530FLR	MEPCO	4703-13-4530	4
R22	RES,SMD,200PPM,1%,1/8 W,49.9ohm	CRCW1206-49.9FT	DALE	4703-13-4999	1
R13 R15 R3 R4 R7 R8	RES, SMD GP 1/8 LONG 200PPM 5% 120 OHM	MCR18EZH121JW5%	ROHM	4703-20-1215	6
R24	RES, SMD GP 1/8 LONG 200PPM 5% 160 OHM	MCR18EZHJW161	ROHM	4703-20-1615	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
R1 R12 R14 R2 R5 R6	RES,200 PPM,1/8W,5%,82 OHM,SMD,1206	9C12063A82R0JLR	MEPCO	4703-20-8205	6
RN1 RN2 RN4 RN5 RN7	RES NETWORK 470 10PIN SIP BUSS	4310R-101-471	BOURN	4770-00-0009	5
RN3 RN6	NTWK,RES, 47 OHM, 7 RES,ISOL,SMD	SOMC-1403-470G	DALE	4770-00-0070	2
U19	OP-AMP,HI SLEW RATE, CURNT FEBK,SMD,SO-8	CLC409AJE	COMLR	7000-04-0901	1
U17	RECEIVER,ECL,3 LINE,PLCC	MC10H116FN	MOT	8000-10-0116	1
U18	D/A,12 BIT,50MSPS,ECL INP,PLCC PKG	TDC1112R3C2	TRW	8000-11-1201	1
U4 U5	XCVR,OCTAL,HCTMOS,TTL I/O,SOIC PKG	MC74HCT245ADW	MOT	8000-74-2451	2
U3	GATE,QUAD,2-INP,SOIC PKG	SN74ALS32D	TI	8000-74-3251	1
U10 U11 U12 U8 U9	XLATOR,QUAD,TTL-TO-ME CL,PLCC PKG	MC10H124FN	MOT	8001-01-2402	5
U1 U16	XLATOR,QUAD,TTL-TO-EC L,PLCC PKG	MC10H125FN	MOT	8001-01-2502	2
U13 U14 U15	FLIP-FLOP,D-TYPE,HEX, ECL,PLCC PKG	MC10H176FN	MOT	8001-01-7602	3
U20 U21 U22 U23 U24 U25 U26 U27	SRAM,CMOS, 32K x 8, 15 NS, SOJ PKG	TC55B328J-15	TOSH	8007-12-5616	8
NONE	PLD,EECMOS,GAL16V8, 10 INP, 8 I/O 7.5 NS, 20 PIN PLCC	GAL16V8B-7LJ	LATT	8016-08-0750	1
U2	ASIC,ADDRESS GENE. AND SEQUENCER,AGS4804	8700-00-0007	WVTK	8700-00-0007	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
NONE	CAP,CER,.001 MF,+-10%,50V,X7R,.1" LS	SR155C102KAA	AVX	1500-01-0207	1
NONE	CAP,CER,COG, 50V, 5%, 220 PF, .1 LS	SR155A221JAA	AVX	1500-02-2101	1
NONE	CAP,ELECT,100MF,35V RADIAL LEAD,SP .20	NRE101M35V10X12.5(OBS)	NIC	1500-31-0102	1
NONE	CAP,ALUM ELECT,470MF,20%,16V,RADAIL .2" SP,10MM DIAx12.5MM HT	KME16VB471M10X12LL	UNCON	1500-34-7114	1
NONE	CAP,TANT,22MF,15V	196D226X9015KA1(OBS)	SPRAG	1500-72-2601	1
NONE	CONN, BNC(PC)	227161-1	AMP	2100-01-0019	2
NONE	RELAY,2 FORMC,5V,DIP 14	DF2E-DC5V	AROMT	4500-00-0033	2
NONE	RELAY,1 FORMC,5V,.312H,.296W	HD1E-M-DC5V	AROMT	4500-00-0034	2
NONE	RELAY, 2 FORM C, 5V, LOW PROFILE	TQ2E-5V	AROMT	4500-00-0038	1
NONE	RES,C,1/2W,5%,10	RC-1/2-10J	STKPL	4700-25-0100	2
NONE	RES,C,1/2W,5%,22	RC-1/2-220J	STKPL	4700-25-0220	2
NONE	RES,C,1/2W,5%,8.2	RC-1/2-8R2J	STKPL	4700-25-0829	2
NONE	RES,MF,1/2W,1%,100	RN65D-1000F	TRW	4701-23-1000	2
NONE	RES, MF,1W,.25%, T2 100	5053RC100R0C	MEPCO	4701-38-1000	2
NONE	RES NETWORK 470 10PIN SIP BUSS	4310R-101-471	BOURN	4770-00-0009	1
NONE	DIODE, HIGH CONDUCTANCE, ULTRA FAST	1N5282	FAIR	4801-01-5282	5
NONE	DIODE,1N4002 GEN PURPOSE RECT. 100V,1A	1N4002	FAIR	4801-02-0001	2
NONE	DIODE, DUAL SWITCHING, COMMON CATHODE,SOT-23	MMBD6100L(OBS)	MOT	4807-02-6100	1
NONE	DIODE 5082-2811 SCHOTTKY,15V,20MA	5082-2811	HP	4809-02-2811	2
NONE	DIODE,BB909B	BB909B	PHLIP	4899-00-0040	2
NONE	LED,AMBER,RECT BAR	LTL-3251A	LITE	4899-00-0056	1
NONE	LED,GREEN,PC MT.,INT HOLDER	550-0206	DIALT	4899-00-0066	1
NONE	LED,YELLOW,PC MOUNT,INT. HOLDER	550-0306	DIALT	4899-00-0067	1
NONE	LED,RED, PC MT., INT. HOLDER	550-0406	DIALT	4899-00-0068	1
NONE	TRANS, NPN, TO-92	PN3563	NSC	4901-03-5630	1
NONE	TRANS	2N3866	MOT	4901-03-8660	2

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
NONE	TRANS 2N5771 PNP SWITCH TO-92	2N5771	NSC	4901-05-7710	2
NONE	TRANS,FET P CHANNEL	VP0106N3	SUPER	4902-01-0601	1
NONE	TRANSISTOR,PNP,HI FREQ,MED PWR SELECTED	SRF2573	MOT	4902-02-5730	1
NONE	TRANSISTOR,NPN,HI FREQ, MED PWR SELECTED	4902-02-5740	WVTK	4902-02-5740	1
NONE	SWITCH,DIP,8 POS,PC MOUNT,SPST	LD08-S	C&K	5199-00-0013	1
NONE	VOLT REGULATOR,3 TERMINAL ADJUSTABLE,POS,TO39	LM317H	NSC	7000-03-1710	1
NONE	VOLT REGULATOR, 3 TERMINAL ADJUSTABLE, NEGATIVE, T0-39	LM337 H	NATNL	7000-03-3701	1
NONE	DARLINGTON ARRAYS, HIGH VOLTAGE, HIGH CURRENT	ULN2003A	SPRAG	7000-20-0300	2
NONE	OP,AMP CURRENT FEEDBACK,50mhz	EL2020CN	ELAN	7000-20-2000	1
NONE	OP AMP,HI SLEW RTE,WIDEBND,JFET,PRIME	MC34081AP	MOT	7003-40-8101	1
NONE	SRAM,32KX8,120NS,CMOS	HM62256LFP-12T	HTACH	8000-43-0256	1
NONE	MICROPROCESSOR,68000, CPU,CMOS,PLCC	MC68HC000FN8	MOT	8000-68-0008	1
NONE	MULTIVIBRATOR,RETRIG, MONOSTABLE,SOIC	SN74LS123D	TI	8000-74-0123	1
NONE	DECODER/DEMULTIPLEXER ,3-TO-8 LINE SOIC	SN74ALS138D	TI	8000-74-0138	1
NONE	GATE,NOR,DUAL,5 INPUT,SOIC	N74LS260D	SIG	8000-74-0260	1
NONE	COUNTER,DUAL,4 BIT,SOIC	SN74LS393D	TI	8000-74-0393	1
NONE	NON-INVERT BUFFER,HEX,TTL,SOIC	SN74ALS1035D	TI	8000-74-1035	1
NONE	GATE,INVER, SCHMITT TRIG, HEX,ACT-CMOS	MC74ACT14N	MOT	8000-74-1431	1
NONE	XCVR,OCTAL REGISTERED,3-STATE OUTPUTS,CMOS	IDT74FCT543SO	IDT	8000-74-5430	1
NONE	FLIP FLOP,OCTAL D,W/3 STATE OUTPUTS,CMOS	IDT74FCT574SO	IDT	8000-74-5740	1
NONE	SHFT REG,8-BIT,CMOS,OUTPUT CLR	SN74HC595N	TI	8000-74-5950	2
NONE	XCVR,OCTAL BUS,3-STATE OUTPUTS,CMOS	IDT74FCT645SO	IDT	8000-74-6450	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT
NONE	MECL 10KH HIGH-SPEED ECL	MC10H102P	MOT	8001-01-0201	2
NONE	RECEIVER-TRIP LN ECL	MC10H116P	MOT	8001-01-1601	1
NONE	XLATOR,QUAD,TTL-TO-MECL,PLCC PKG	MC10H124FN	MOT	8001-01-2402	1
NONE	FLIP-FLOP,D-TYPE,HEX,ECL,PLCC PKG	MC10H176FN	MOT	8001-01-7602	1
NONE	PROM,EE, 2Kx8,CMOS, 200NS	XL28C16AP-200	EXEL	8002-81-6000	1
NONE	SRAM,CMOS,32K*8,15NS, .3IN DIP	MCM6206CNP15	MOT	8007-12-5615	1
NONE	REAL TIME CLOCK W/RAM,CMOS	HD146818FP	HTACH	8014-68-1800	1
NONE	ENCODER,8-TO-3 PRIORITY	SN74LS148D	TI	8074-14-8000	1
NONE	LINE DRIVER,OCTAL BUFFER,CMOS	IDT74FCT541SO	IDT	8074-54-1000	1
NONE	PGA,VXI ASIC CHIP	8700-00-0006	AMI	8700-00-0006	1

REFERENCE DESIGNATORS	PART DESCRIPTION	ORIG-MFGR-PART-NO	MFGR	WAVETEK NO.	QTY/PT

WAVETEK PARTS LIST	TITLE ASSY, RECOMMENDED SPARE PARTS KIT FOR MODEL 1395	ASSEMBLY NO. 1200-00-3629 PAGE 4	REV A

Self Calibration Appendix A

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 `CALibrate[:ALL]?` query. This information supplements paragraphs 3.3.3.1 (`CALibrate`) and 3.3.6 (`*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 response as the `CALibrate[: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 1395 is set up with the following configuration:

Output off

Filter off

Sync Marker Off

Sumbus output off

Master clock output off

Output Attenuator /1 range

Preamplifier 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 a 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. For more information on the cause of the Trace Memory failure refer to Section 6 of this manual.

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 minimum 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 6 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 midrange 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 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 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 measurement not settled after 10 measurements
1	Voltage measurement 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 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 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 measurement not settled after 10 measurements
1	Voltage measurement 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 measurement not settled after 10 measurements
1	Voltage measurement 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 measurement not settled after 10 measurements
1	Voltage measurement 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' midrange 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 measurement not settled after 10 measurements
1	Voltage measurement over range
2	Amplitude offset correction value diverging
3	Amplitude 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 3.3.3.7 (`TEST`) and 3.3.6 (`*TST?`).

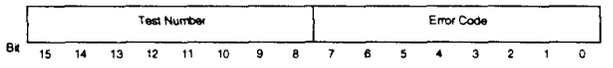
B.2 TEST Query Response

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 response 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 1395'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 controllability 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

Sumbus output off

Master clock output off

Output Attenuator /1 range

Preamplifier 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 a 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. For more information on the cause of the Trace Memory failure refer to Section 6 of this manual.

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 indicates 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:

$$V_{expected} = 7.5 * (6.0/7.5) * (24.9/124.9)$$

The offset voltage is programmed to its positive maximum. The voltage is measured and its magnitude 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 offset voltage is programmed to its negative maximum. The voltage is measured and its magnitude is verified to be greater than $V_{expected}$ and less than $V_{expected}$ + 20%. If the voltage is less than $V_{expected}$, then bit 4 of the Error Code is set to 1. If the voltage is greater than $V_{expected}$ + 20%, 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 is 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 0.0 V ± 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 AGS4804 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:

$$V_{expected} = 7.5 * 2.0 * (1.0/5.0)$$

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 underrange
3	Peak voltage spread overrange
4	Offset voltage positive overrange
5	Offset voltage negative overrange

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 underrange
3	Peak voltage spread overrange
4	Offset voltage positive overrange
5	Offset voltage negative overrange

Test 8 Filters

This test checks DC operation of the Bessel and Elliptic filters by monitoring voltage levels at the output of the Preamplifier 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:

$$V_{expected} = (7.5/2.0) * 2.0 * (9.0/79.27)$$

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 verified. 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 verified. 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 verified. 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 verified. 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 verified. 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 12 SUMBUS Driver

This test checks the operation of the SUBMUS driver circuit by monitoring voltage levels at the preamplifier and SUMBUS outputs.

First the gain of the SUMBUS Driver is tested.

The Amplitude Control DAC is programmed for maximum amplitude. The Waveform DAC data inputs are set to 0xFFF to set the amplitude voltage to its positive peak, and the voltage at the SUMBUS and PREAMP test points is measured. The Waveform DAC data inputs are set to 0x001 to set the amplitude voltage to its negative peak, and the voltage at the SUMBUS and PREAMP test points is measured again. The spread between the positive and negative peak voltages is calculated for the PREAMP and SUMBUS test points. A 3 volt peak-to-peak spread at the PREAMP test point should correspond to a 2 volt peak-to-peak spread at the SUMBUS test point. If the SUMBUS voltage is less than 2/3 of the PREAMP voltage - 5%, then bit 2 of the Error Code is set to 1. If the SUMBUS voltage is greater than 2/3 of the PREAMP voltage + 5%, then bit 3 of the Error Code is set to 1.

Next the offset of the SUMBUS driver is tested. This test checks that the peak-to-peak voltage swing as measured at the SUMBUS output is centered at the same place as the measured offset voltage.

The Waveform DAC data inputs are set to 0x800 to set the amplitude voltage to its midpoint, and the voltage at the SUMBUS test point is measured. This voltage is compared to the midpoint of the peak-to-peak spread measured in the first part of this test. If the difference between the measured offset and the calculated offset is less than 5% of -Vpp at the SUMBUS test point, then bit 4 is set to 1. If the difference between the measured offset and the calculated offset is greater than 5% of +Vpp at the SUMBUS test point, then bit 5 is set to 1.

If any portion of this test fails the Test Number is set to 12.

Error Code Bit	Meaning
0	Voltage not settled after 20 measurements
1	DVM over range
2	Peak voltage spread overrange
3	Peak voltage spread underrange
4	Offset voltage positive overrange
5	Offset voltage negative overrange

Test 13 SUMBUS Receiver

This test checks the operation of the SUMBUS Receiver circuit by monitoring voltage levels at the PWRAMPHI test point. The SUMBUS Driver and Receiver are simultaneously enabled, allowing voltage from the Preamp to reach the Output Amplifier through the SUMBUS circuitry.

The amplitude is programmed to one fourth of maximum. The SUMBUS and OUTPUT attenuators are set to /1. The peak-to-peak voltage at the PWRAMPHI test point is measured with the SUMBUS Driver and Receiver disabled. This voltage is stored as V_{off_vdiff}. The peak-to-peak voltage is measured again with the SUMBUS Driver and Receiver enabled. This voltage is stored as V_{on_vdiff}.

The following relationship is checked:

$$V_{on\_vdiff}/V_{off\_vdiff} = 2 \pm 5\%$$

If V_{on_vdiff} is higher than expected then bit 2 is set to 1. If V_{on_vdiff} is lower than expected then bit 3 is set to 1.

The center points of the peak-to-peak voltage spreads are calculated and stored as V_{on_voffs} and V_{off_voffs}. The voltage measurements are compensated for the scaling resistors at the PWRAMPHI test point. The offset at the SUMBUS output is measured by setting the input to the Waveform DAC to 0x800. This voltage is stored as V_{sum_voffs}. The following relationship is checked:

$$(V_{on\_voffs} - V_{off\_voffs})/15 - V_{sum\_voffs} = 0$$

If this equation results in a voltage greater than +10 mV, then bit 4 is set to 1. If this equation results in a voltage less than -10 mV, then bit 5 is set to 1.

If any portion of this test fails the Test Number is set to 13.

Error Code Bit	Meaning
0	Voltage not settled after 20 measurements
1	DVM over range
2	Peak voltage spread overrange
3	Peak voltage spread underrange
4	Offset voltage positive overrange
5	Offset voltage negative overrange

Test 14 SUMBUS Attenuator

This test checks the operation of the SUMBUS Attenuator circuit by monitoring voltage levels at the PWRAMPHI test point. The SUMBUS Driver and Receiver are simultaneously enabled, allowing voltage from the Preamp to reach the Output Amplifier through the SUMBUS circuitry.

The amplitude is programmed to one fourth of maximum. The SUMBUS and OUTPUT attenuators are set to /1. The peak-to-peak voltage at the PWRAMPHI test point is measured with the SUMBUS Driver and Receiver disabled. This voltage is stored as V_{off_vdiff}.

The peak-to-peak voltage is measured again with the SUMBUS Driver and Receiver enabled. The difference between the two peak-to-peak voltages is calculated and stored as V_{ex}.

The peak-to-peak voltage is measured with the 6 dB attenuator enabled. This voltage is stored as V_{on_vdiff}. The following relationship is checked:

$$V_{on\_vdiff} - V_{off\_vdiff} = V_{ex}/2 \pm 5\%$$

If V_{on_vdiff} is higher than expected then bit 2 is set to 1. If V_{on_vdiff} is lower than expected then bit 3 is set to 1.

The peak-to-peak voltage is measured with the 12 dB attenuator enabled. This voltage is stored as V_{on_vdiff}. The following relationship is checked:

$$V_{on\_vdiff} - V_{off\_vdiff} = V_{ex}/4 \pm 5\%$$

If V_{on_vdiff} is higher than expected then bit 4 is set to 1. If V_{on_vdiff} is lower than expected then bit 5 is set to 1.

The peak-to-peak voltage is measured with the 24 dB attenuator enabled. This voltage is stored as V_{on_vdiff}. The following relationship is checked:

$$V_{on\_vdiff} - V_{off\_vdiff} = V_{ex}/16 \pm 5\%$$

If V_{on_vdiff} is higher than expected then bit 6 is set to 1. If V_{on_vdiff} is lower than expected then bit 7 is set to 1.

If any portion of this test fails the Test Number is set to 14.

Error Code Bit	Meaning
0	Voltage not settled after 20 measurements
1	DVM over range
2	06 dB attenuator voltage too high
3	06 dB attenuator voltage too low
4	12 dB attenuator voltage too high
5	12 dB attenuator voltage too low
6	24 dB attenuator voltage too high
7	24 dB attenuator voltage too low

TEST:RAM?

This query checks the integrity of Trace Memory by performing 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 chips is 32K bytes or 128K bytes. Thus the total size of Trace Memory is 32K words or 128K words.

The RAMs are electrically isolated from the CPU bus by the AGS4804 and two 74HCT245 transceivers. The address lines are buffered by the AGS4804 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 struck 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 assumed 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 accessible, 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

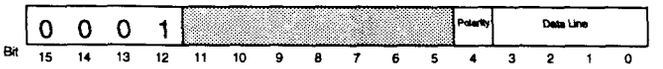
This test checks for shorts or opens on the data line connections.

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.

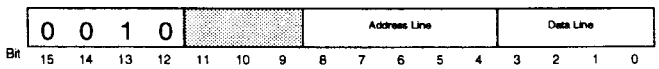
Data Line	CPU Signal	Polarity	Meaning
0	D00		
1	D01	0	Stuck low
2	D02	1	Stuck high
3	D03		
4	D04		
5	D05		
6	D06		
7	D07		
8	D08		
9	D09		
10	D10		
11	D11		
12	D12		
13	D13		
14	D14		
15	D15		

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 through 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

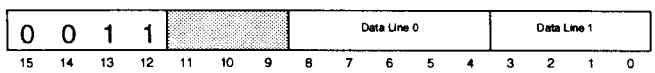
If a failure is detected in this subtest, the code 2 is placed in the Test Number field. The encoding of the Data Line field is the same as used for the Bit Position field in subtest 1. It identifies the shorted data line. The encoding of the Address Line field is as shown below. It identifies which 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.

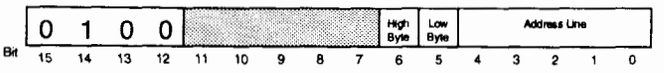
Subtest 4

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 1395'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 1395 SCPI Command Trees

Appendix D provides the programmer/operator with a complete set of SCPI Command trees for the model 1395. These trees are an alternate method (to the Command Table) of constructing syntactically correct SCPI commands.

Model 1395 SCPI Command Table

Table 3-2, Command Summary, and Table C-2, Model SCPI Command Syntax, provide the programmer/operator with a complete set of SCPI Commands for the model 1395. 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 1395 SCPI version

The Model 1395 VXIbus Arbitrary Waveform Synthesizer has been designed to comply with SCPI Version 1992.0, dated February 1992.

C.4 Model 1395 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-2 for the complete syntax of Model 1395 SCPI commands. Model 1395 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-2 for the complete syntax of Model 1395 SCPI commands. Model 1395 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 are not yet listed in the SCPI specification. Refer to Table C-2 for the complete syntax of Model 1395 SCPI commands. Any Model 1395 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-2) would use footnotes to indicate partial conformance to the SCPI Specification.

Table C-2. Model 1395 Command Summary

KEYWORD	PARAMETER FORM	NOTES
CALibration		Confirmed
[:**ALL**]?		Confirmed
:**DATA**	<block>	Confirmed
:**AFC**orrection	<point>,<frequency>,<gain>	Not SCPI Approved
:**AMPL**itude		Not SCPI Approved
[:**GAIN**]	<numeric_value> (0.0, **0.0**, 1000)	Not SCPI Approved
:**OFFS**et	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:**AMZ**ero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:**OFFS**et		Not SCPI Approved
[:**GAIN**]	<**POS**itive | **NEG**ative>,<numeric_value>	Not SCPI Approved
:**OFFS**et	<**POS**itive | **NEG**ative>,<numeric_value>	Not SCPI Approved
:**PAZ**ero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:**SCMZ**ero	<numeric_value> (0.0, **2048**, 4095)	Not SCPI Approved
:**STOR**e		Not SCPI Approved
:**STAT**e	<Boolean_data> (**ON**)	Confirmed
INITiate		Confirmed
[:**IMM**ediate]		Confirmed
:**CONT**inuous	<Boolean_data> (**ON**)	Confirmed
OUTPut		Confirmed
:**CLOC**k		Not SCPI Approved
:**FREQ**uency	<numeric_value> (1e-1, **1e3**, 1e8)	Not SCPI Approved
:**SOUR**ce	<**RAST**er | **SYNT**hesizer>	Not SCPI Approved
:**ECLTrg<n>**		Confirmed
[:**STAT**e]	<Boolean_data> (**OFF**)	Confirmed
:**FILT**er		Confirmed
[:**LPAS**s]		Confirmed
:**SEL**ect	<**BESS**el | **ELL**iptic>	Not SCPI Approved
[:**STAT**e]	<Boolean_data>	Confirmed
[:**STAT**e]	<Boolean_data> (**OFF**)	Confirmed
:**SUMB**us		Not SCPI Approved
[:**STAT**e]	<Boolean_data> (**OFF**)	Not SCPI Approved
:**TRIG**ger		Not SCPI Approved
:**MARK**er	<**TRIG**ger | **POS**ition>	Not SCPI Approved
:**SOUR**ce	<**BIT**|**BCOM**plete|**LCOM**plete|INTernal>	Not SCPI Approved
:**END**		Not SCPI Approved
[:**STAT**e]	<Boolean_data> (**OFF**)	Not SCPI Approved
:**TTLTrg<n>**		Confirmed
[:**STAT**e]	<Boolean_data> (**OFF**)	Confirmed
RESet		Confirmed
[**SOUR**ce]		Confirmed
:**AM**		Confirmed
[:**STAT**e]	<Boolean_data> (**OFF**)	Confirmed
:**MODE**	<**AM**> | <**SCM**>	Not SCPI Approved
:**CLOC**k		Not SCPI Approved
:**CONF**igure	<**INP**ut | **OUTP**ut>	Not SCPI Approved

Table C-2. Model 1395 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
:USER	<trace_name>	Not SCPI Approved
:MODE	<FIXed | SEQuence>	Confirmed[1]
:LIST		Confirmed
:FREQuency	<value>,<list_index> (1e-1, 1e3, 2e7),(0,1023)	Confirmed
: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	<ZCRoss | 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	<INTernal | EXTernal | ECLTrg<n>>	Confirmed[1]
:SEQuence		Not SCPI Approved
:ADVance	<AUTOmatic | TRIGgered>,<list_index> (0, 3)	Not SCPI Approved
:DWELl	<numeric_value>,<list_index> (1, 1, 65535),(0,3)	Not SCPI Approved
:FUNCtion	<trace_name>,<list_index> (0, 3)	Not SCPI Approved
:LENGth	<numeric_value> (2, 2, 4)	Not SCPI Approved
:TRIGger		Not SCPI Approved
:MODE	<SYNChronous | ASYNchronous>	Not SCPI Approved
:SENSe	<EDGE | LEVel>	Not SCPI Approved
:SUMBus		Not SCPI Approved
[:STATe]	<Boolean_data> (OFF)	Not SCPI Approved
:ATTenuation	<numeric_value> (0, 0, 42)	Not SCPI Approved
:SWEep		Confirmed
:COUNt	<numeric_value> (1, 1, 1e6)	Confirmed
:DIRection	<UP | DOWN>	Confirmed
:SPACing	<LINear | LOGarithmic>	Confirmed
:TIME	<numeric_value> (30e-3, 1, 1e3)	Confirmed
:MODE	<CRESet | TRESet | HRESet | CREVerse | TREVerse | HREVerse | MANual>	Confirmed[1]

Table C-2. Model 1395 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
[SOURce] (continued)		
:VOLTage		Confirmed
[:LEVel]		Confirmed
[:IMMediate]		Confirmed
[:AMPLitude]	<numeric_value> (0, 1, 7.5)	Confirmed
:OFFSet	<numeric_value> (-7.5, 0, 7.5)	Confirmed
STATus		Confirmed[3]
:OPERation		Confirmed[3]
:CONDition		Confirmed[3]
:ENABle		Confirmed[3]
:ENABle?	<NRf>	Confirmed[3]
[:EVENt]		Confirmed[3]
:QUEStionable		Confirmed[3]
:CONDition		Confirmed[3]
:ENABle		Confirmed[3]
:ENABle?	<NRf>	Confirmed[3]
[:EVENt]		Confirmed[3]
:PRESet		Confirmed[3]
SYSTem		Confirmed
:ERRor?		Confirmed
:DATE	<year>,<month>,<day>	Confirmed
:TIME	<hour>,<minute>,<second>	Confirmed
:VERSion?		Confirmed
TEST		Confirmed
[:ALL]?		Confirmed[1]
:RAM?		Confirmed[1]
TRACe		Confirmed
:CATalog?	<trace_name>,<trace_name>,. . .	Confirmed
:DEFine	<trace_name>,<numeric_value>|<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
: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-2. Model 1395 Command Summary (Continued)

KEYWORD	PARAMETER FORM	NOTES
TRIGger		Confirmed
:**COUN**t	<numeric_value> (1, **1**, 1048575)	Confirmed
:**GATE**		Not SCPI Approved
[:**STAT**e]	<Boolean_data> (**OFF**)	Not SCPI Approved
[:**IMM**ediate]		Confirmed
:**POL**arity	<**POS**itive | **NEG**ative>	Not SCPI Approved
:**SOUR**ce		Confirmed[1]
[:**STAR**t]	<**INT**ernal | **EXT**ernal | **CHA**in | ECHain | TTLTrg<n>>	Not SCPI Approved
:**ADV**ance	<**INT**ernal | **EXT**ernal | **CHA**in | ECHain>	Not SCPI Approved
:**TIM**er	<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.

SCPI Command Tree

Appendix D

D.1 Command Trees

SCPI Command Trees provide a visual alternative to Command Tables as an aid in building syntatically correct commands to operate the Model 1395. 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.

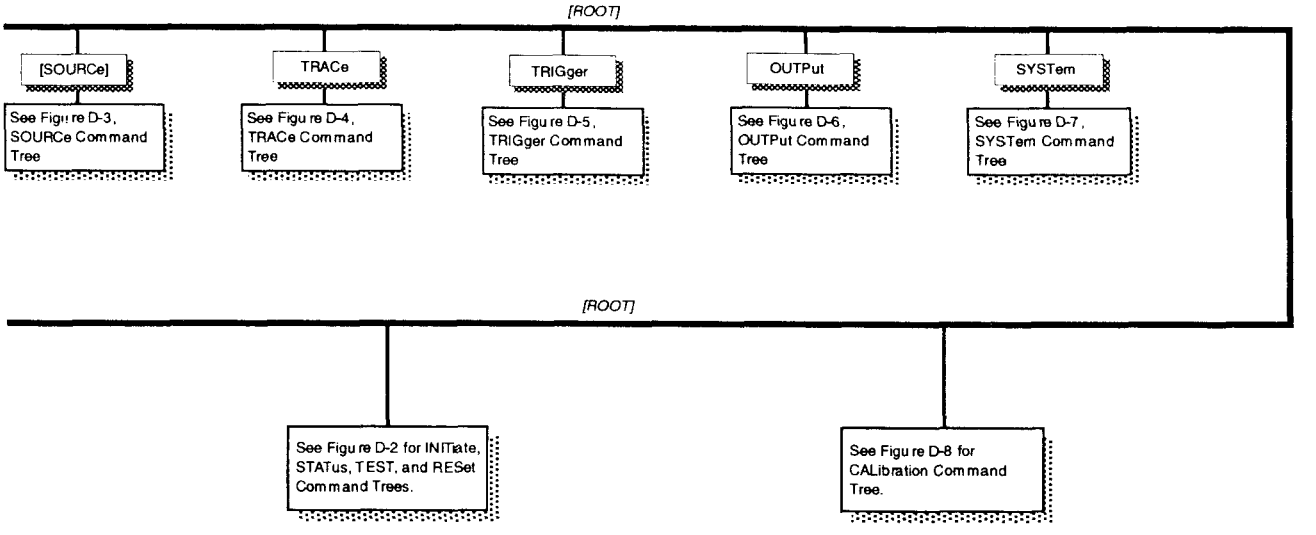


Figure D-1. Subsystems (Root Node).

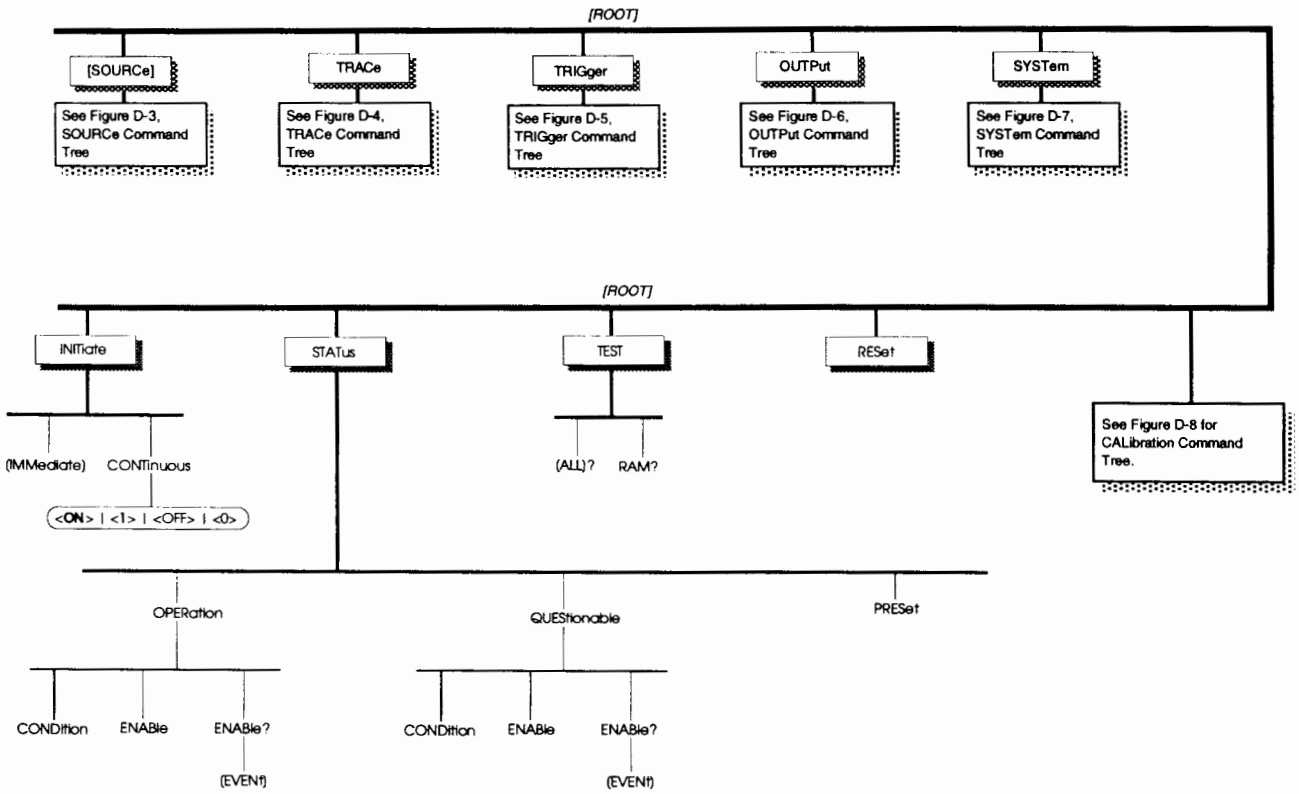


Figure D-2. INITiate, STATus, TEST, and RESet Subsystems.

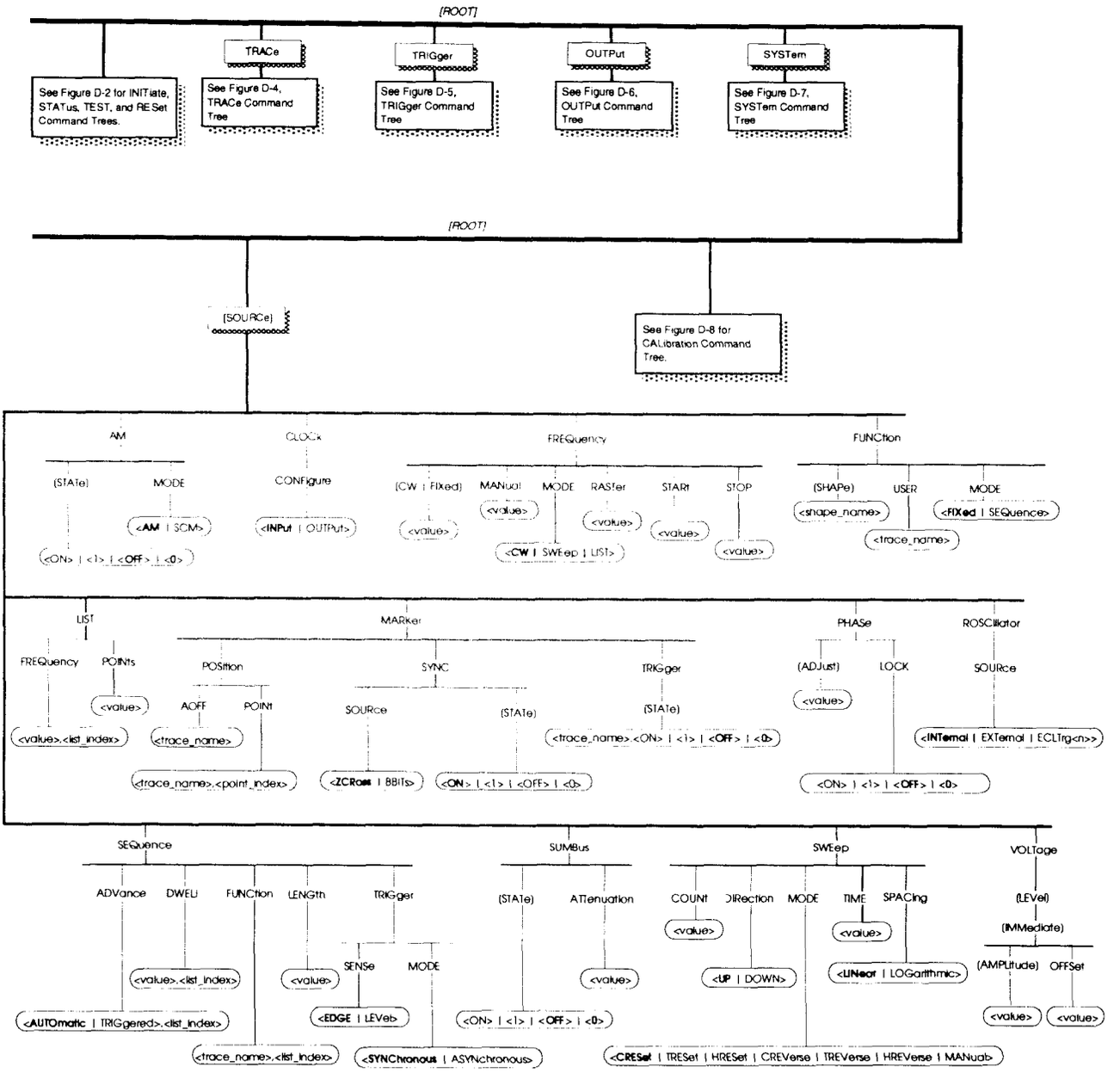


Figure D-3. SOURce Subsystem.

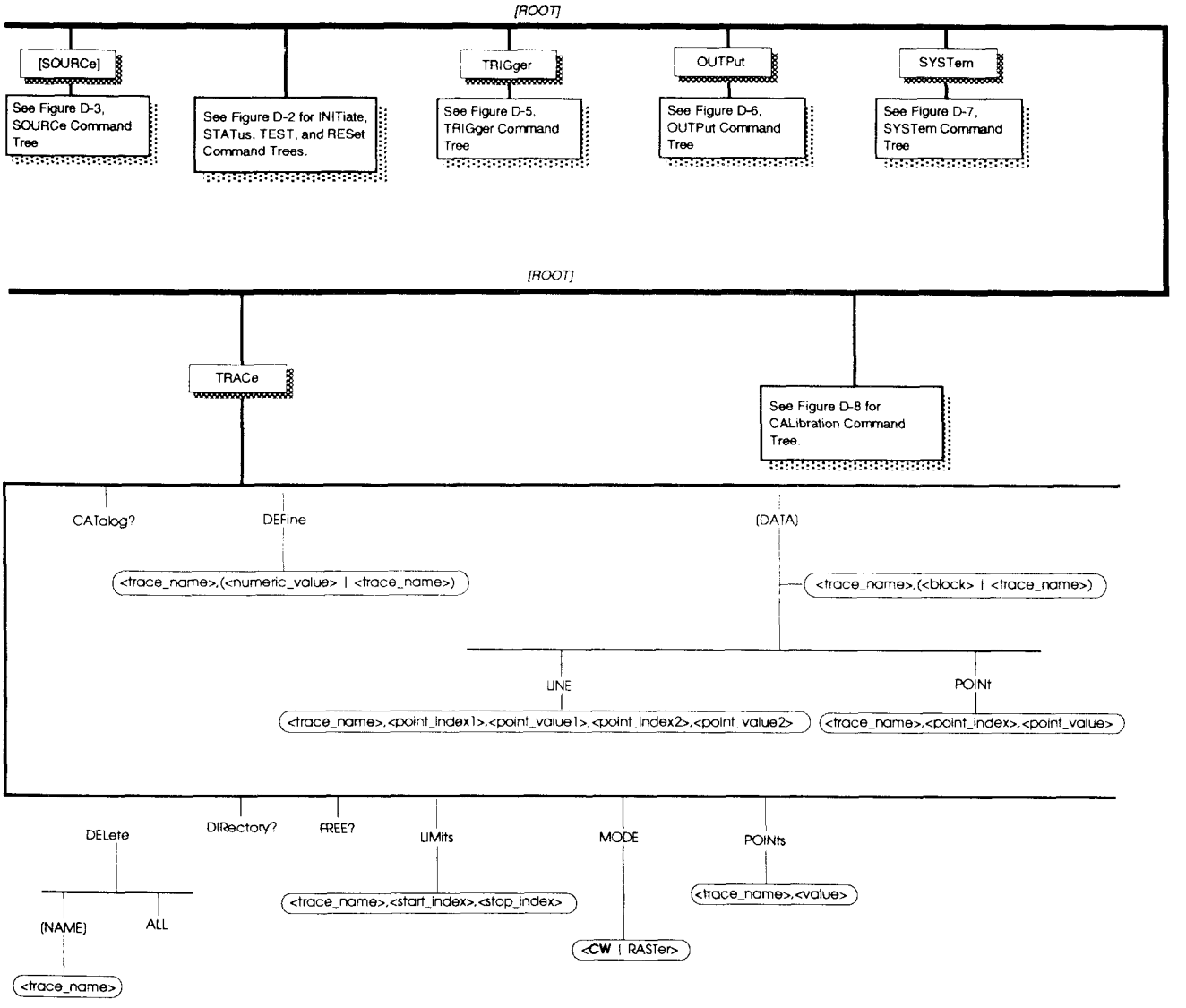


Figure D-4. TRACe Subsystem.

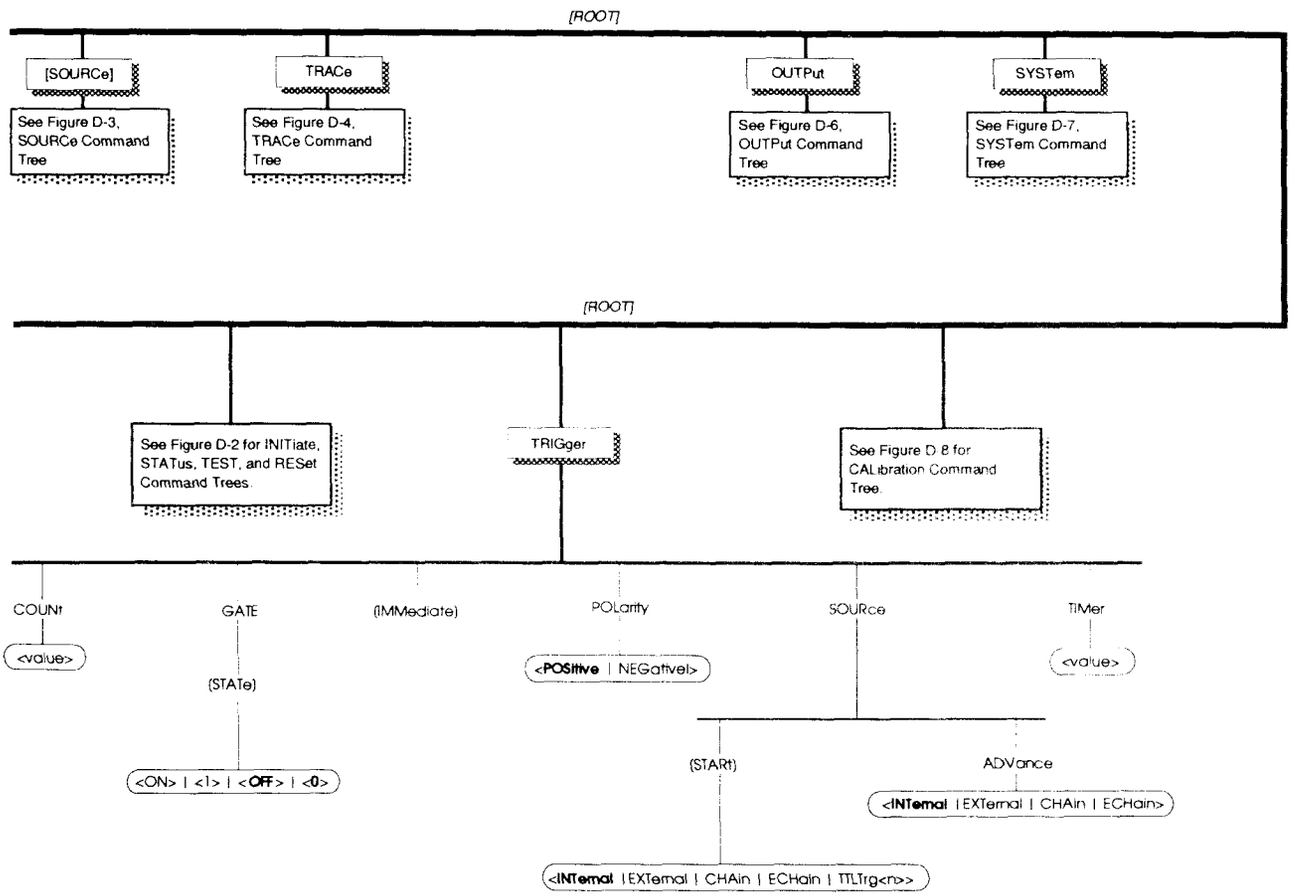


Figure D-5. TRIGger Subsystem.

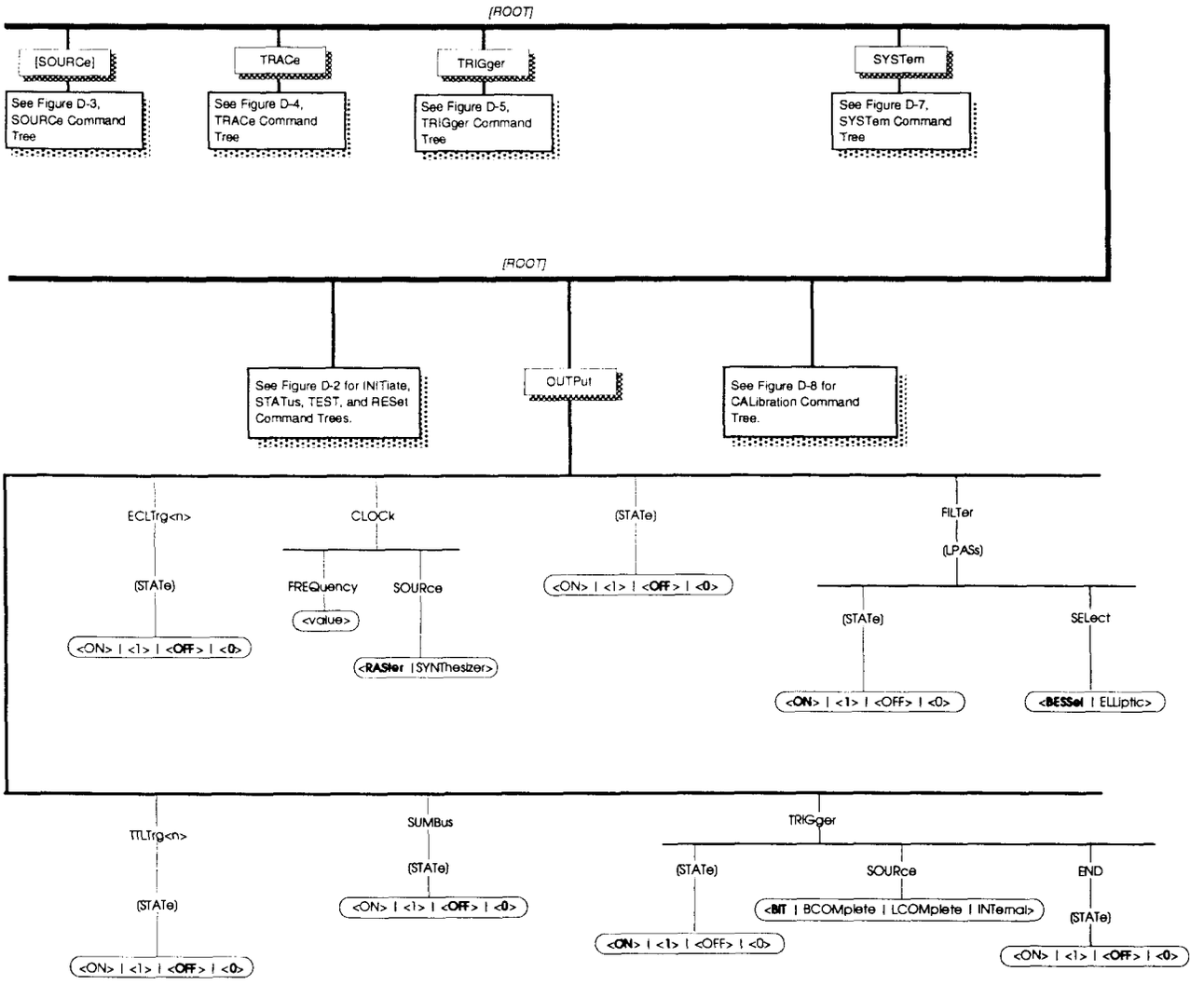


Figure D-6. OUTPut Subsystem.

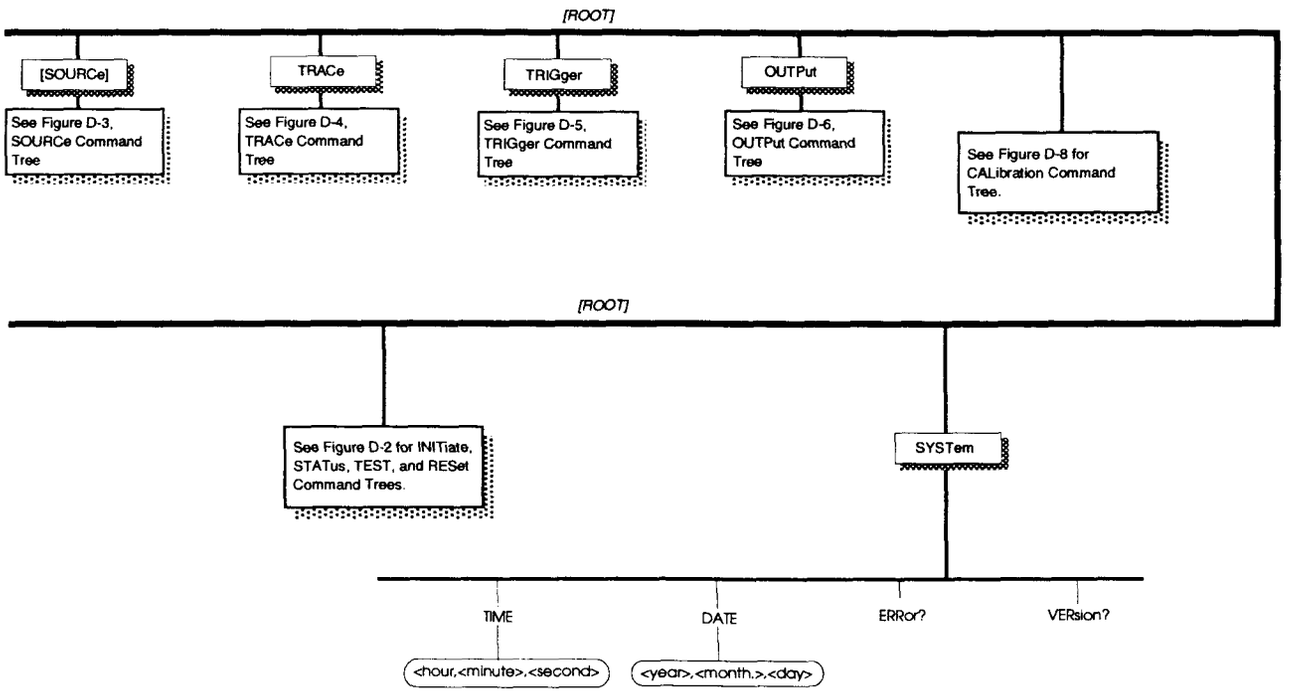


Figure D-7. SYSTem Subsystem.

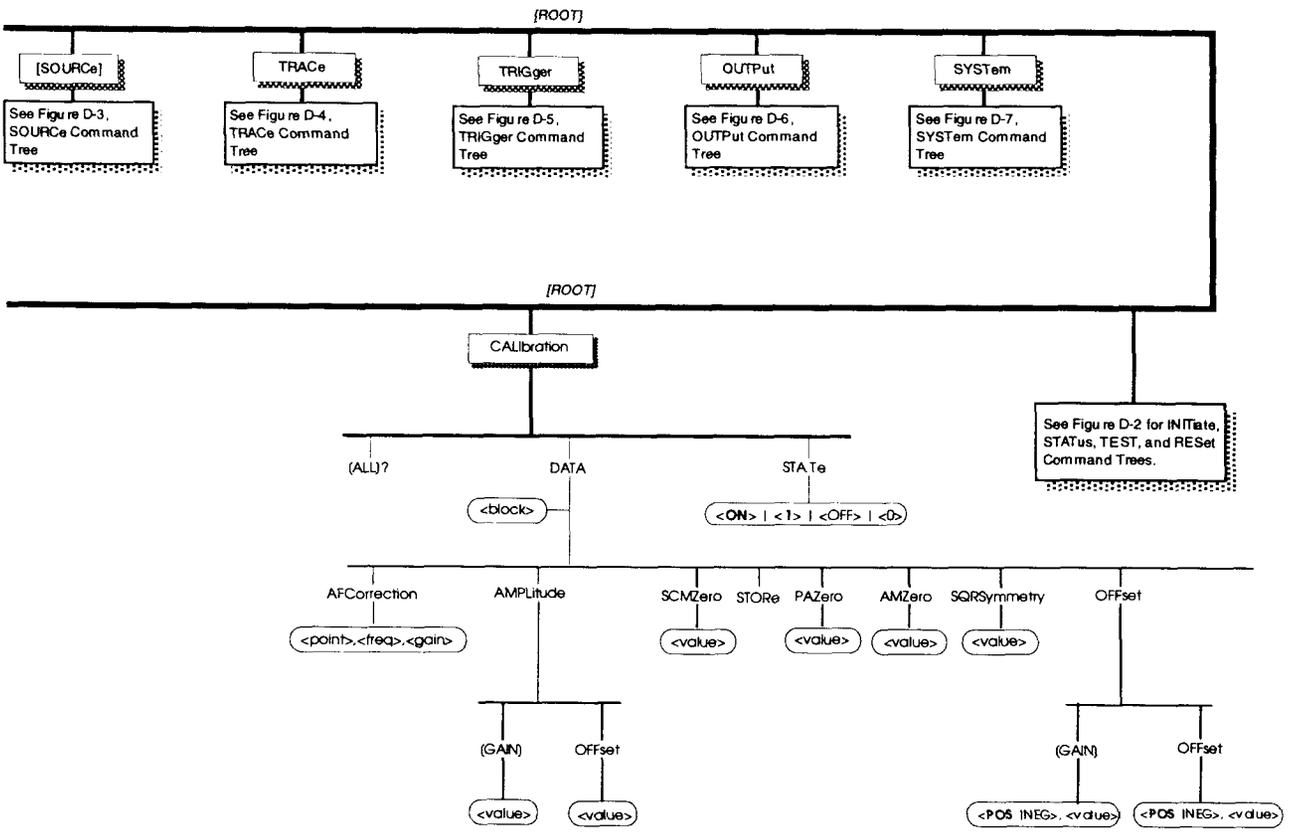


Figure D-8. CALibration Subsystem.

Sample Programs

Appendix E

E.1 Introduction

This appendix provides the Operator/Programmer with programming examples to create arbitrary waveforms on the Model 1395.

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 predefined waveforms to supply data to the trace. Complete SCPI keywords are used, with the approved abbreviation in capitals.

The final program in this appendix illustrates a "C" program running on a VXIbus embedded controller.

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:FREQ-
    uency: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 1395 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 1395 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 subcommand 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, 2 Vpp Internal Trigger 1 kHz

```
10   !This creates WAVE2, a 200
     point, 20 kHz sine at 2 Vpp;
     triggered at 1 kHz
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:CONT-
     inuous 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 transferred 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 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 1395 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 1395 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 1395 found
by the Resource Manager during system power on is added to the devices[] array.

*****************************************************************************/
smem_download(devices)
DEVICE   *devices;
{
int      error;

long i,                          /* 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 1395       */
    offset=devices[0].offset;  /* Shared Memory offset of first 1395    */
    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 1395, 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);
        }

    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);
}
```

MATE Syntax Reference

Appendix F

F.1 Introduction

This appendix provides the Operator/Programmer with additional information needed to interface the Model 1395 with external instruments using the IEEE-488 (GPIB) bus and command language. This information supplements the information contained in paragraphs 3.3.3 (Model 1395 SCPI Commands), 3.3.4 (IEEE-488.2 Command Commands), and Appendices C and D.

F.2 GAL Command

The Model 1395 MATE is initialized into the MATE/CIIL language at power-on. The GAL command needs to be sent in order to put the 1395 into standard SCPI operation. When the SCPI command SYST:LANG:CIIL is sent the 1395 reverts back to MATE syntax. Sending a DCL will also revert back to MATE.

F.3 ARB Generator Documentation

Opcode

FNC	Function opcode
SET/SRN/SRX	Used to set modifier value
CLS/OPN	Close or open output/input for module
RST	Resets module to power-up condition
CNF/IST	Initiates selftest for module
STA	Requests status of module

Nouns

ACS	Function generator sinewave
DCS	Function generator DC
RPS	Function generator ramp waveform
SQW	Function generator square waveform
TRI	Function generator triangle waveform
WAV	Arbitrary waveform (user defined)

Modifier	Limits	How used
BURS	1 to 1048575	Burst count
CLFR	1e-1 to 5e7	Sets Raster clock frequency (WAV only) (CLSC INT only).
CLSC	INT/EXT	Selects WAV clock source.
DCOF	± 7.5	DC offset of selected waveform.
FREQ	1e-6 to 25e6	Frequency of selected waveform.
GALV	+ 1.0	External gate level (TTL levels only).
GASC	INT/EXT	Selects gate source.
GSTA	POS/NEG	Selects external gate start slope. Must be opposite GSTO.
GSTO	POS/NEG	Selects external gate stop slope. Must be opposite GSTA.
PERI	4e-8 to 1e6	Sets period of waveform. If CLSC EXT selected the output frequency will be external clock frequency divided by (programmed period times 50e6).
STIM	± 1.0	WAV data points (min 8 and max 32768 without VLTR or VLTS, else min of 2 and max of 2048).
TIMP	50	50 ohm impedance only.
TRLV	+ 1.0	Trigger level, source external only (TTL levels only).
TRSC	INT/EXT	Selects trigger source.
TRSL	POS/NEG	Selects external trigger slope.
VLTR	————	Sets voltage ramp.
VLTS	————	Sets voltage step.
VLPK	0 to 7.5	Sets amplitude peak value.
VLPP	0 to 15.0	Sets amplitude peak to peak value.
VOLT	± 7.5	Sets DC output to value.
VRMS	0 to 5.3	Sets sine amplitude RMS value.
	0 to 7.5	Sets square amplitude RMS value.

F.4 Arb Generator Error Messages

Below is a list of the possible error messages that might occur during operation. Each is listed inside quotes with explanation of what it means. Each message will be preceded by one of the following two ASCII strings: "F07ARB00 (MOD):" or "F07ARB00 (DEV):"

Where <ASCII> is seen in string means any continuous group of ASCII characters. These may be a valid noun, modifier, or data as well as misspelled or garbage strings. If the group of characters are valid syntax then they are out of order or other reason for not being accepted.

1. *"<ASCII> not allowed"*
   This ASCII group is not allowed as used.

2. *"<ASCII> missing"*
   A mandatory modifier is missing in command string. Could be VLPP for instance, which is necessary for ACS function commands.

3. *"<ASCII>"*
   The ASCII string here could be any of the module dependent error messages listed in the operator's manual.

4. *"GSTA/GSTO slope error"*
   An attempt was made to program gate start and stop slope to same value.

5. *"Channel number error"*
   Only channel 0 is allowed on arb generator since it only has one output to be programmed.

6. *"Settings missing from FNC command"*
   The command string was good up to where the SET modifier portion should be. There are some mandatory SET modifiers needed.

7. *"INT/EXT not allowed with <ASCII>"*
   The ASCII string here is a valid modifier but does not take on the values of INT or EXT.

8. *"POS/NEG not allowed with <ASCII>"*
   The ASCII string here is a valid modifier but does not take on the values of POS or NEG.

9. *"Numeric value not allowed with <ASCII>"*
   The modifier sent does not accept numeric value. TRSC and GSTA are some examples of modifiers that do not take numeric value.

10. *"Value needed with <ASCII>"*
    The modifier noted by ASCII string needs a value sent with it.

11. *"Command terminated before complete"*
    "FNC ACS<cr><lf>" will cause this error message. Up to the terminating characters the command is good, it is just not complete.

12. *"Max points is 2048 with VLTR/VLTS else 32768"*
    With the STIM command, there has to be a minimum of 8 points and a maximum of 32768 points when not using VLTR or VLTS.

13. *"STIM values between +1 and -1 only"*
    The values associated with STIM have maximum limits of +/-1. The arb will scale these values up to maximum programmed amplitude.

14. *"Cannot allocate memory for data"*
    The memory allocation for temporary storage of these STIM values failed. This error should not occur under normal conditions.

15. *"Cannot free allocated memory for data"*
    The free memory for temporary storage of these STIM values failed. This error should not occur under normal conditions.

16. *"TRSL/TRLV only allowed with TRSC EXT"*
    Trigger slope and level only valid for external trigger.

17. *GSTA/GSTO/GALV only allowed with GASC EXT"*
    Gate slope and level only valid for gate source external.

18. *"BURS not allowed with GASC"*
    Burst mode uses TRSC.

19. *"50.0 ohm impedance only"*
    Fixed impedance setting.

20. *"CLFR not allowed with CLSC EXT"*
    Clock frequency not valid unless clock source internal.

21. *"TRFR not allowed with external source"*
    Trigger frequency not allowed unless TRSC or GASC has Internal source.

22. *"TRFR mandatory with internal source"*
    Trigger frequency mandatory when TRSC or GASC has Internal source.

23. *"Need minimum of 8 points without VLTR/ VLTS"*
    With the STIM command, there has to be a minimum of 8 points and a maximum of 32768 points when not using VLTR or VLTS.

24. *"Need minimum of 2 points with VLTR/VLTS"*
    With the STIM command, there has to be a minimum of 2 points and a maximum of 2048 points when using VLTR or VLTS.

F.5 Misc Arb Generator Documentation

When using WAV and STIM, the following apply:

1. If VLTR is used, the minimum number of points is 2 and the maximum is 2048. There are 15 points interpolated between each pair of given points in STIM list. The total size of the waveform will be (n-1)*16+1 where n is the number of points sent in STIM list.

2. If VLTR is used, the minimum number of points is 2 and the maximum is 2048. There are 16 points drawn for each given point in STIM list. The total size of the waveform will be (n*16) where n is the number of points sent in STIM list.

3. If VLTR or VLTS is not used, the minimum number of points is 8 and maximum of 32768. Each point in STIM the list is drawn individually. This gives the user the capability of creating very complex waveforms. At lower frequencies, this will look like the step waveform format.

4. When CLFR is used in internal CLSC, the Arb is put into Raster scan mode. The resultant output frequency is CLFR value divided by the number of points in waveform. Determine number of points from 1, 2, or 3 above.

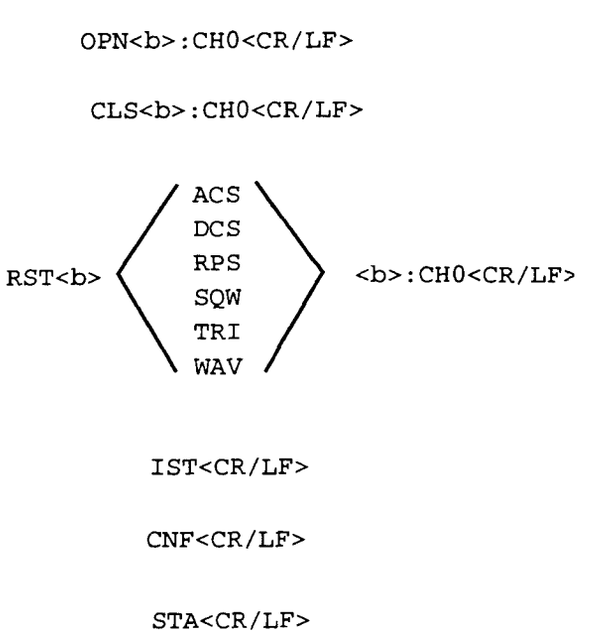
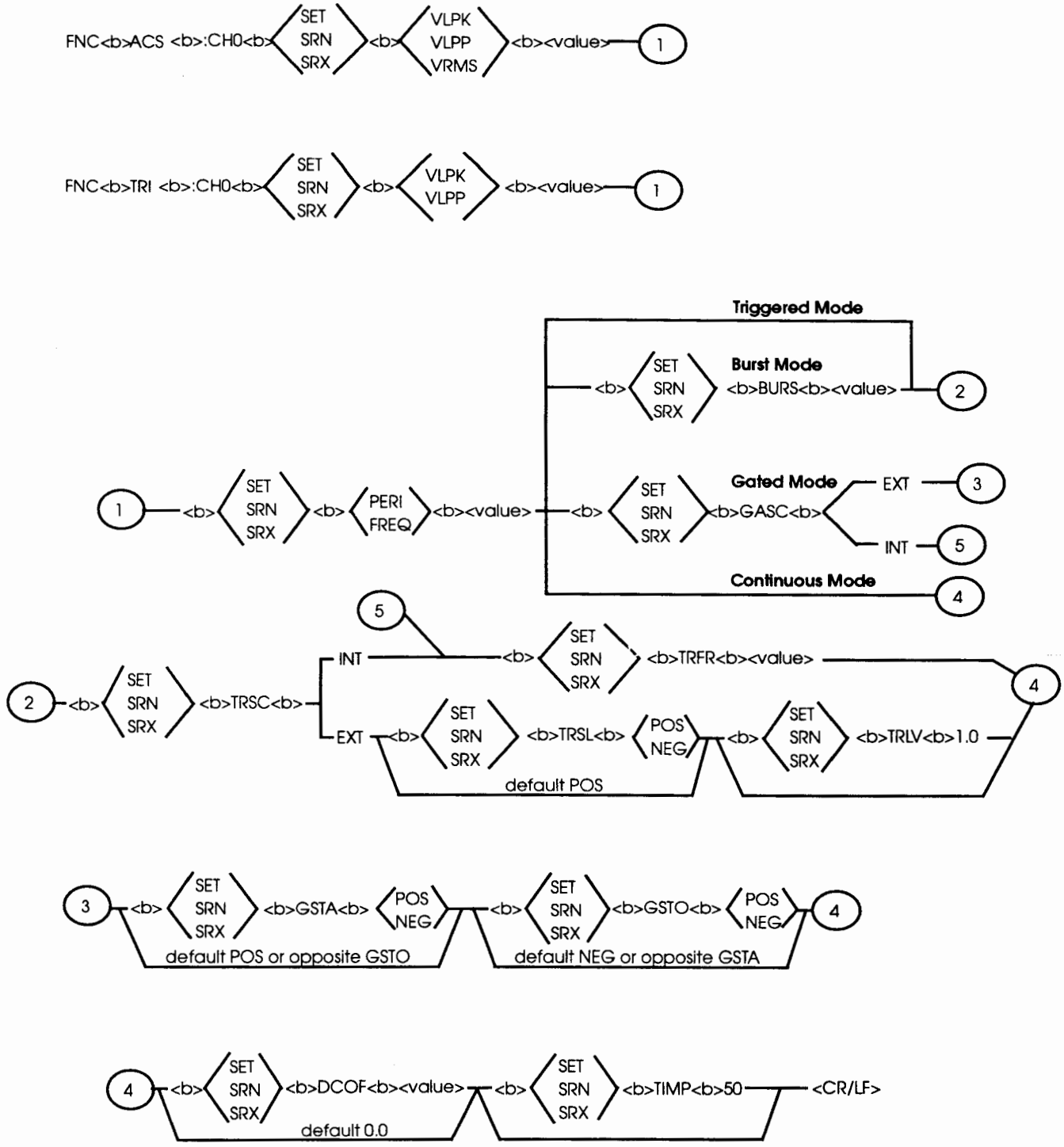
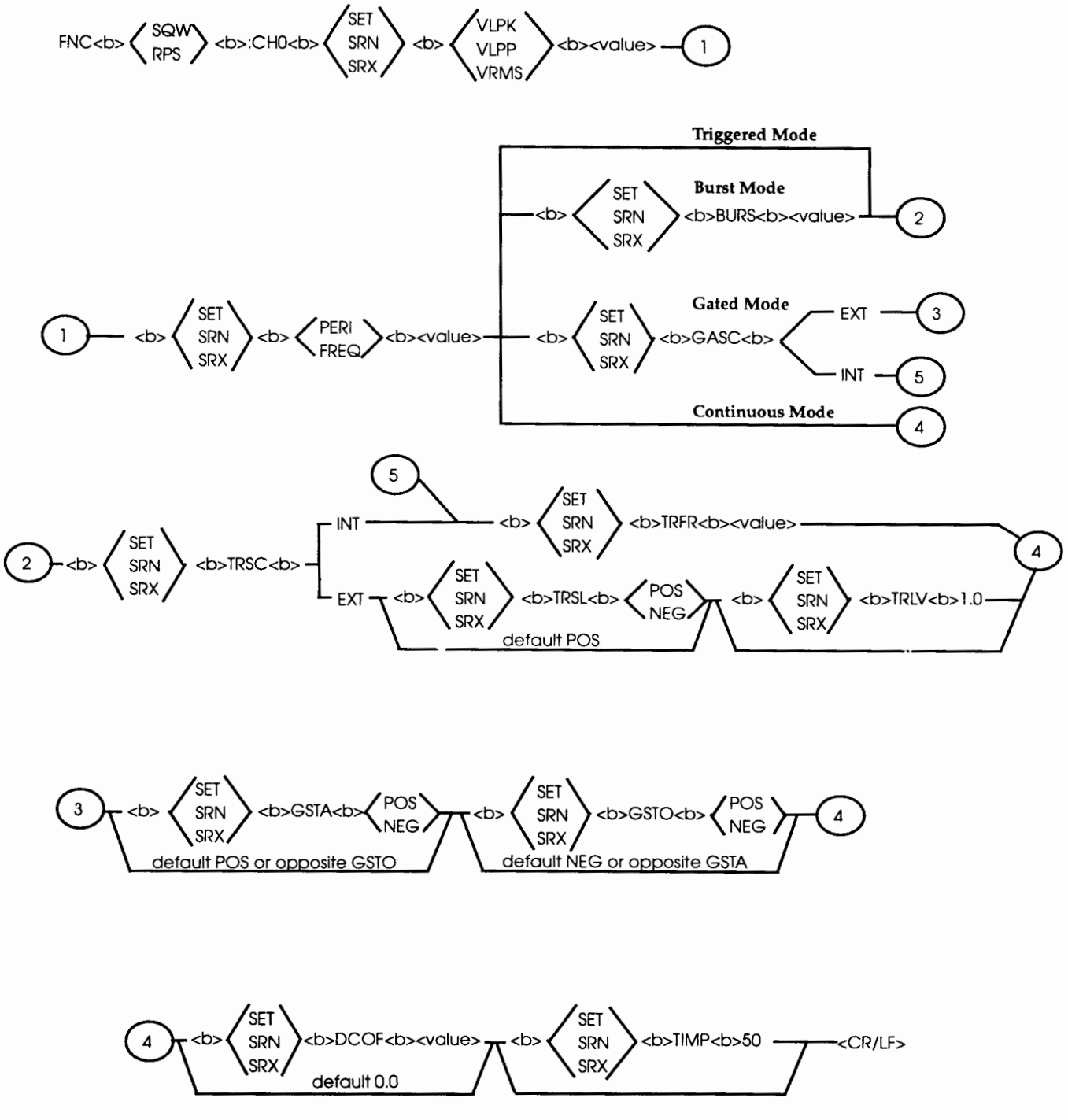


Figure F-1 Common Command Format



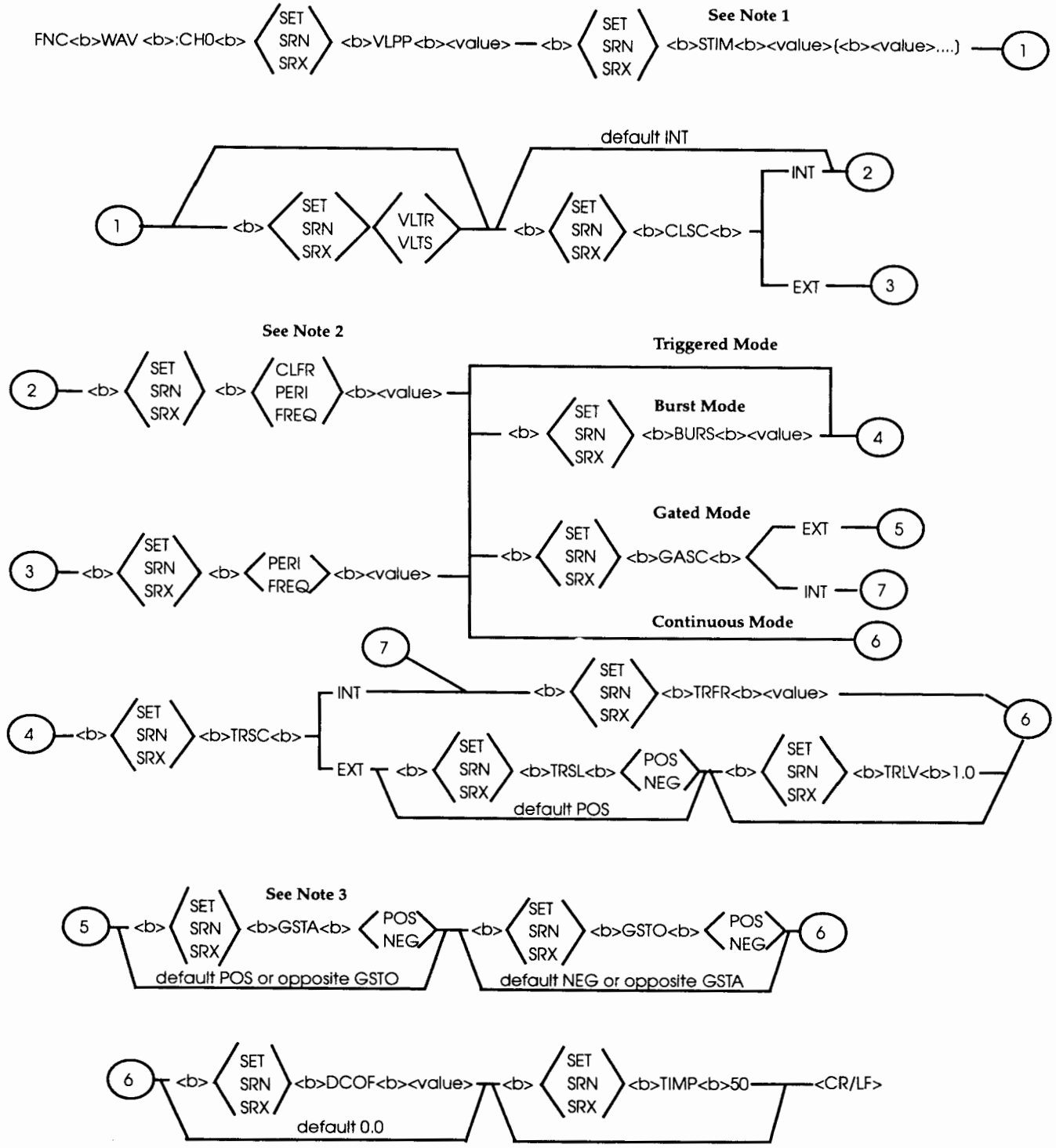
Note: *GSTA and GSTO cannot be set to same slope*

Figure F-2 Sine and Triangle Format



Note: *GSTA and GSTO cannot be set to same slope*

Figure F-3 Square and Ramp Syntax Format



Note 1 *STIM needs a minimum of 8 points and maximum of 32768.*
Note 2 *CLFR will put arb in raster scan mode.*
Note 3 *GSTA and GSTO cannot be set to same slope*

Figure F-4 User Defined Waveform Format

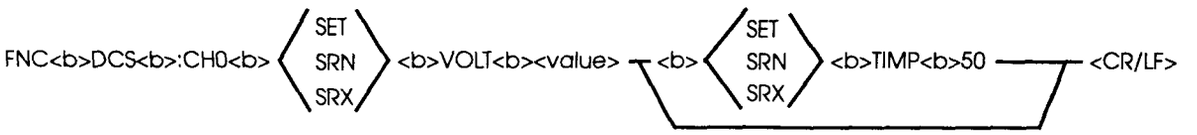


Figure F-5 DC Function Format