# MODEL 2414A

## ARBITRARY WAVEFORM GENERATOR

OPERATION MANUAL

PART NUMBER – 810008-CD
REV. C

TEGAM, INC.
TEN TEGAM WAY
GENEVA, OHIO 44O41
TEL (440) 466-6100
FAX (440) 466-6110
EMAIL sales@tegam.com

NOTE: This user's manual was as current as possible when this product was manufactured. However, products are constantly being updated and improved. Because of this, some differences may occur between the description in this manual and the product received.

# TABLE OF CONTENTS


**SECTION 1 INTRODUCTION**

1.1 PRODUCT SUMMARY ..... 1-1
1.2 KEY FEATURES ..... 1-1
1.3 SPECIFICATIONS ..... 1-2

**SECTION 2 CONNECTING POWER**

2.1 GROUND CONNECTION ..... 2-1
2.2 SELECTING LINE (MAINS) VOLTAGE ..... 2-1
2.2.1 Line Switch ..... 2-1
2.2.2 HI/LO Switch ..... 2-1
2.2.3 Fuses ..... 2-1

**SECTION 3 QUICK START**

3.1 INTRODUCTION ..... 3-1
3.2 CONNECTIONS ..... 3-1
3.3 DEFAULT SETTINGS ..... 3-2
3.4 HOW TO CHANGE DEFAULT PARAMETERS ..... 3-2
3.4.1 Turning Output On ..... 3-2
3.4.2 Selecting Standard Waveforms ..... 3-2
3.4.3 Amplitude and Offset ..... 3-4
3.4.4 Sample Clock and Output Frequency ..... 3-4
3.4.5 Modes ..... 3-4

**SECTION 4 GENERATOR CONCEPT & CONTROL**

4.1 INTRODUCTION ..... 4-1
4.2 DIGITAL SYNTHESIS ..... 4-1
4.3 CLOCKING SYSTEM ..... 4-2
4.4 HOW WAVEFORM IS PLAYED BACK ..... 4-3

**SECTION 5 MEMORY ORGANIZATION**

5.1 INTRODUCTION ..... 5-1
5.2 DEFAULT PARTITIONING ..... 5-1
5.3 WAVEFORM NUMBERING ..... 5-1
5.4 CHANGING WAVEFORM BLOCK LENGTHS ..... 5-2
5.4.1 Standard Wave ..... 5-2
5.4.2 Waveform Number Block Lengths ..... 5-2
5.5 DELETING WAVEFORMS ..... 5-2
5.6 INSERTING NEW WAVEFORM NUMBERS ..... 5-2

## SECTION 6 CREATING AND EDITING WAVESHAPES


6.1 INTRODUCTION ..... 6-1
6.2 LINE MODE ..... 6-1
6.2.1 Editing From Start Point ..... 6-2
6.2.2 Editing From Left Anchor ..... 6-2
6.2.3 Creating Line Segments ..... 6-4
6.3 VERTEX MODE ..... 6-4
6.3.1 Selecting Left and Right Anchor Points ..... 6-4
6.3.2 Selecting Vertex Point ..... 6-4
6.3.3 Scaling ..... 6-6
6.3.4 Smoothing ..... 6-6
6.3.5 Inserting Standard Functions ..... 6-6
6.3.6 Summing Standard Functions ..... 6-7
6.3.7 Dump Function ..... 6-7
6.3.8 Move ..... 6-8
6.4 POINT MODE ..... 6-8
6.4.1 Selecting Left and Right Anchor Points ..... 6-8
6.4.2 Entering Point Values ..... 6-9
6.5 MATH OPERATIONS ..... 6-9
6.5.1 Selecting Math Function ..... 6-9
6.5.2 Selecting Waveform Numbers ..... 6-10
6.6 EXAMPLES ..... 6-10
6.6.1 Insert and Sum Functions ..... 6-10
6.6.2 Math Function ..... 6-12


## SECTION 7 SEQUENCE GENERATOR


7.1 INTRODUCTION ..... 7-1
7.2 PROGRAMMING A SEQUENCE ..... 7-1
7.3 DELETING A SEQUENCE ..... 7-4
7.4 ADDING A STEP TO AN EXISTING SEQUENCE ..... 7-4
7.5 DELETING A STEP FROM AN EXISTING SEQUENCE ..... 7-4
7.6 MODIFYING A STEP WITHIN AN EXISTING SEQUENCE 7-4


## SECTION 8 MULTIPLE UNITS


8.1 INTRODUCTION ..... 8-1
8.2 PARALLEL OPERATION ..... 8-1
8.2.1 Clock Connections ..... 8-1
8.2.2 Trigger Connections ..... 8-1
8.3 SERIES OPERATION ..... 8-2
8.3.1 Clock Connection ..... 8-2
8.3.2 Trigger Connection ..... 8-2

## SECTION 9 OTHER FEATURES


| | | |
|---|---|---|
| 9.1 | VIEW FUNCTION | 9-1 |
| 9.2 | SYNC OUTPUTS | 9-1 |
| 9.2.1 | End Pulse | 9-2 |
| 9.2.2 | Run | 9-2 |
| 9.2.3 | End Block | 9-2 |
| 9.3 | Z-AXIS LEVEL | 9-2 |
| 9.4 | OUTPUT FILTER | 9-2 |
| 9.5 | INTERNAL TRIGGER GENERATOR | 9-3 |
| 9.6 | RTS | 9-3 |
| 9.7 | HOLD | 9-3 |
| 9.8 | MONITOR BURST COUNT | 9-3 |
| 9.9 | DSOLINK™ | 9-4 |
| 9.9.1 | RS-232 Waveform Generator Setup | 9-4 |
| 9.9.2 | GPIB Waveform Generator Setup | 9-4 |
| 9.9.3 | DSO Setup | 9-5 |
| 9.10 | WaveWorks Pro™SOFTWARE | 9-5 |


## SECTION 10 RS-232C & GPIB


| | | |
|---|---|---|
| 10.1 | INTRODUCTION | 10-1 |
| 10.2 | QUICK REFERENCE | 10-1 |
| 10.3 | RS-232C OVERVIEW | 10-3 |
| 10.3.1 | Introduction | 10-3 |
| 10.3.2 | Interface Requirements | 10-3 |
| 10.3.3 | 2414A Setup | 10-5 |
| 10.3.4 | Verify Communication | 10-6 |
| 10.3.5 | Command Syntax | 10-6 |
| 10.3.5.1 | Common Commands | 10-6 |
| 10.3.5.2 | Event Register and Status and Error Reporting | 10-7 |
| 10.3.5.3 | Functional Syntax Elements | 10-7 |
| 10.3.5.4 | Data Formats | 10-9 |
| 10.3.6 | Sample Program | 10-10 |
| 10.4 | GPIB (IEEE-488.2) OVERVIEW | 10-12 |
| 10.4.1 | Introduction | 10-12 |
| 10.4.2 | Common Commands | 10-12 |
| 10.4.3 | Status and Event Registers | 10-13 |
| 10.4.4 | Functional Elements | 10-14 |
| 10.4.5 | Data Formats | 10-16 |
| 10.4.6 | Error Reporting | 10-18 |
| 10.5 | REMOTE COMMAND SET | 10-19 |
| 10.5.1 | Introduction | 10-19 |
| 10.5.2 | Command Set Hierarchy | 10-20 |
| 10.5.3 | Stacked Queries | 10-20 |
| 10.5.4 | Command Set | 10-21 |
| 10.5.5 | Sequence Generator Application Notes | 10-36 |
| 10.5.6 | Programming Example | 10-38 |
| 10.6 | RESET AND FACTORY DEFAULTS | 10-40 |
| 10.7 | WAVEFORM EDITING PRINCIPLES | 10-42 |
| 10.8 | WAVEFORM MEMORY FORMATS | 10-44 |
| 10.8.1 | Decimal Waveform Download | 10-44 |
| 10.8.2 | Binary Waveform Download | 10-45 |

**APPENDIX**

MENUS
BLOCK DIAGRAM
PROGRAMMING WORKSHEET
WAVEFORM DESIGN SHEET
DICTIONARY OF TERMS
GPIB AND RS-232 CONNECTOR PINOUTS
STANDARD WAVESHAPE EQUATIONS
ERROR CODES
WAVEFORM MEMORY ORGANIZATION
WAVEFORM MEMORY REMOTE COMMANDS
OSCILLOSCOPE SETUPS FOR DSOLINK™
WARRANTY
WaveWorks Pro™ WAVEFORM CREATION SOFTWARE
CALIBRATION VERIFICATION PROCEDURE

**INDEX**

# SECTION 1
# INTRODUCTION

## 1.1 PRODUCT SUMMARY

The 2414A provides an unlimited variety of signal source waveshapes and sequences. Twenty standard waveshapes are pre-programmed for instant recall. Arbitrary waveshapes can be downloaded from a computer or digital storage oscilloscope or created locally using line, vertex or point editing. All standard and custom waveshapes are digitally synthesized, with 12 bits (4095 points) of amplitude resolution and 160k of waveform memory (128k active and 32k buffer). An accurate internal clock up to 20 MHz provides an extremely wide range of output frequencies. The instrument is easy to operate, with an intuitive front panel and a menu-driven, easy-to-read 40-character backlit display. Parameter changes may be done conveniently using either a numeric keypad or a rotary edit knob. Two previously stored waveforms can be added, subtracted or multiplied together for special applications such as an amplitude-modulated signal. Diverse waveform sequences can easily be created by using a Sequence Generator option which permits different waveform segments to be repeated and/or linked in any order. All waveforms, as well as instrument setups, are completely non-volatile. An RS-232 remote interface is standard and GPIB (IEEE 488.2) is available as an option. The Model 2414A is housed in an extremely rugged extruded aluminum case sized small enough to fit comfortably on any bench. High-quality, state-of-the-art components assure the utmost reliability and performance.

## 1.2 KEY FEATURES

- 20 Standard Waveshapes
- 128K Active Waveform Memory
- 32k Buffer Waveform Memory
- 20 MHz Sample Clock
- 12 Bits Amplitude Resolution
- Waveform Creation and Editing (Mouse and Pad Included)
- Add, Subtract and Multiply Waveforms
- Loop and Link Waveforms with Sequence Generator (Optional)
- Standard RS-232 Interface
- GPIB Interface (Optional)
- Up to 1000 Waveforms
- Data Transfer from DSO
- ARBLink Software & RS-232 Cable Included

## 1.3 SPECIFICATIONS

**OUTPUT WAVEFORMS**
Up to 1000 Custom Waveforms, Sine, Triangle, Square, ±Ramp, DC, ±Pulse, ±Exponential, AM, SCM, FM, Lin/Log Sweep, Noise, Sin x/x, Gaussian, Haversine, Circle.

**SEQUENCE GENERATOR** (Optional)
**Waveform:** Loop and Link
**Repetitions:** Loop 1,000,000 times; Link1000 waveforms
**Program:** 1000 Steps

**WAVEFORM MEMORY**
**Total:** 160k (163,808 points)
**Active:** 128k (131,040 points)
**Buffer:** 32k (32,768 points) Buffer accessible by RS-232C or GPIB only.
**Vertical Resolution:** 12 bits; 4096 points (+2047, -2048).

**MEMORY PARTITION**
**Active:** 1000 waveforms
**Buffer:** 100 waveforms

**WAVEFORM SAMPLING RATE**
**Range:** 0.1Hz to 20MHz (10s to 50ns).
**Resolution:** 4 1/2 digits
**Accuracy:** ±10ppm

**WAVEFORM RISE/FALL TIME**
Less than 20ns, tested with square wave, filter off, 10Vp-p, 50Ω termination.

**SPECTRAL PURITY**
**THD + Noise:** Typically below 65dB in 80kHz measurement bandwidth. Tested at 20MHz clock; sinewave; 1,000 points (20 kHz); filter on; full amplitude; 50Ω termination.

**AMPLITUDE AND OFFSET**

| Range | Resolution | Accuracy |
|---|---|---|
| ±1.00 to 10V | 10mV | 1% of setting +20mV |
| ±100mV to 999mV | 1mV | 3% of setting +5mV |
| ±10mV to 99.9mV | 100µV | 5% of setting +1mV |

Note: 50Ω source impedance, measured at open circuit, tested with 1kHz sinewave plus DC offset.

**ANALOG FILTER**
User-selectable, 7MHz, 7th-order low-pass filter.

**OPERATIONAL MODES**
**Continuous:** Output runs continuously between selected memory address locations.
**Triggered:** Output at start point until triggered, then runs once.
**Gated:** As triggered except output is continuous until gate signal ends.
**Toggled:** Alternate triggers gate the output waveform.
**Burst:** Each trigger outputs a pre-programmed number of waveforms from 1 to 1,048,575.
**Hold:** Front-panel button or external signal stops waveform at present memory location while applied.
**RTS:** Front-panel button or external signal interrupts the output waveform and returns the output level back to the start level.

**OUTPUTS**
**Output:** Front-panel main waveform output. 50Ω impedance.
**Sync Output:** Front-panel TTL sync output. Programmable address and width. 50Ω impedance.
**Clock In/Out:** Rear-panel ARB waveform sample clock input or output (TTL).
**Reference In/Out:** Rear-panel internal 10-MHz reference output (TTL).
**Sync Trigger Out:** Rear-panel TTL sync for triggering additional units in parallel or series.
**Z Axis Out:** Rear-panel Z Axis output in edit mode.
**Sync 3/Run Out:** Rear-panel TTL output. High when output signal on.
**Sync 4/End Block Out:** Rear-panel TTL output. Single pulse at end of each memory block in continuous and triggered modes; single pulse after each group of cycles in gated and burst modes.

**INPUTS**
**Sum In:** Front-panel input allows external signal to be added to output. Gain = -2 open circuit and -1 with 50Ω output termination and 50Ω input impedance.
**Trigger Input:** Rear-panel TTL trigger input for triggered, gated, toggled, and burst modes.
**Clock In:** Rear-panel waveform sample clock input (TTL ≤20MHz).
**Reference In:** Rear-panel 10-MHz reference input. The internal crystal-controlled oscillator will phase-lock to the input.
**RTS In:** Rear panel TTL input to initiate RTS mode.
**Hold In:** Rear panel TTL input to stop waveform at present level.

**TRIGGER SOURCES**
**External Trigger Input**
**Manual Trigger**
**Internal Trigger Generator:** 0.02 to 10 seconds.

**CREATION TOOLS**
**Waveform Editing:** Point Mode, Line Mode, Vertex Mode; Insert Function, Sum Function, Dump Function, Digital Amplitude/Offset, Smooth, Copy/Paste, Waveform Math (A+B, A-B, AxB).
**Pointing Device:** Mouse or front-panel numeric keypad or edit knob.

**DSO DATA TRANSFER**
**DSOLink:** Direct data download from DSO to waveform memory via RS-232C or GPIB.

**STORED DATA**
**Waveforms:** 1000 Active, 100 Buffer
**Setups:** 30 settings
**Sequence:** 100 files

**REMOTE INTERFACE**
**RS-232C:** 19.2kBaud, max.
**GPIB:** IEEE STD. 488.2-1987 *(Optional)*

**OPTIONS**
**Sequence Generator**
**GPIB Remote Interface**
**WaveWorks Pro™ Waveform Creation Software**

**GENERAL**
**Temperature Range:** +23°C ±3°C for specified operation. Operates 0°C to+50°C. Storage -20°C to+60°C.
**Dimensions:** 11.5cm (4.53 in.) H; 25.8 cm (10.14 in.) W; 30cm (11.81 in.) D.
**Weight:** 5.0kg (11 lbs)
**Power:** 55VA; 45W (max) 100/120/220/240 VAC, +5%, -10%; 48 to 63 Hz.

*Weight and dimensions are approximate.*
*Errors and omissions excepted.*

*Specifications subject to change without notice.*

*TEGAM*, Vertex Formatting, DSOLink and WaveWorks Pro are trademarks of TEGAM, Inc.

This page intentionally left blank.

# SECTION 2
# CONNECTING POWER

## 2.1 GROUND CONNECTION

**WARNING**

To prevent death or injury from electrical shock, be sure the Model 2414A is connected to earth ground through an approved and inspected three-wire power cord.

## 2.2 SELECTING LINE (MAINS) VOLTAGE CAUTION

Severe damage to the Model 2414A can occur if the rear-panel power switches are set to incorrect positions. Be sure to check these settings during initial installation.

### 2.2.1 Line Switch

The Line Switch (see Figure 2-1) selects one of two coarse power ranges. Set the switch to the 120V position for line (mains) voltages from 90 to 126 volts. Set the switch to the 240V position for voltages from 198 to 252 volts.

### 2.2.2 HI/LO Switch

The HI/LO Switch (see Figure 2-1) selects the fine power ranges. Set the switch to the HI position for line (mains) voltages from 108 to 126 or from 216 to 252 volts. Set the switch to the LO position for voltages from 90 to 105 or from 198 to 231 volts.

### 2.2.3 Fuses

A line (mains) fuse is located in the power cord connector module (see Figure 2-1). Replacement fuses must be 5 x 20mm, type GDC slow blow. Use a 0.8A fuse for the nominal 120V range or a 0.4A fuse for the nominal 240V range.

**CAUTION**

To provide proper instrument protection, be sure the correct size fuse is installed.

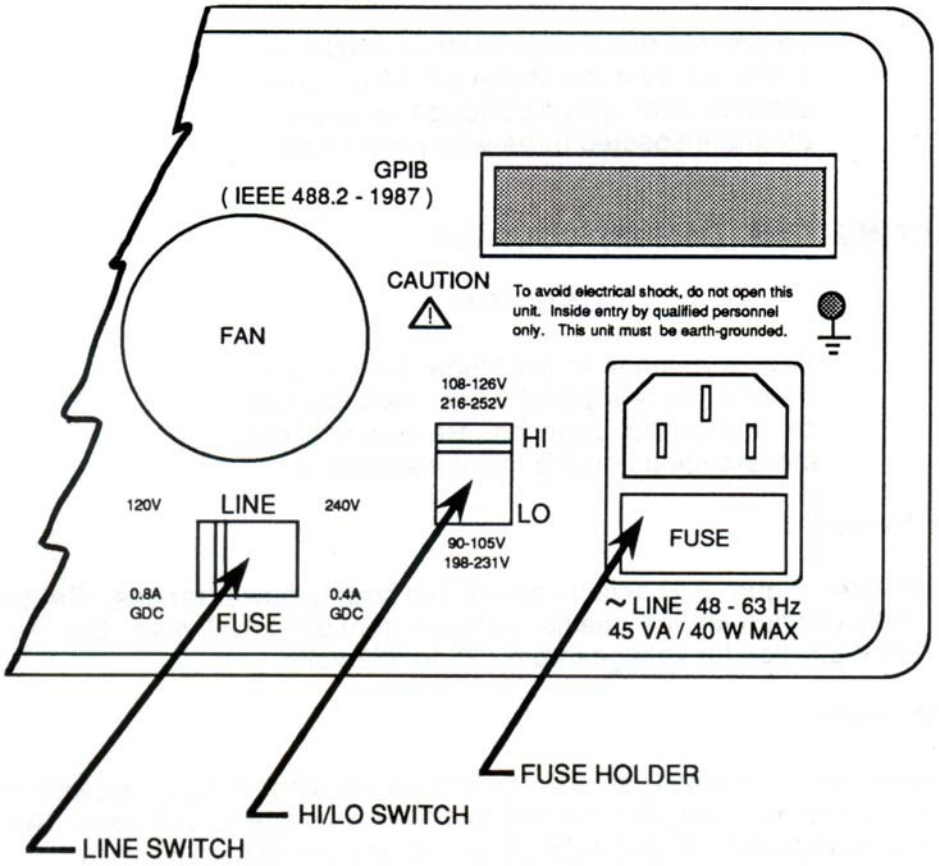



Figure 2-1. Power Switch and Fuse Locations

# SECTION 3
# QUICK START

## 3.1 INTRODUCTION

This section explains how to connect the Model 2414A and how to quickly obtain standard waveforms. Also included are instructions on how to set amplitude, offset, and frequency. The six basic operational modes are explained. More detailed operating procedures, including construction of arbitrary waveforms, are contained in the following sections of this manual. See Section 6 for arbitrary waveforms.

## 3.2 CONNECTIONS

All waveforms are obtained from the OUTPUT BNC connector on the front panel. A TTL sync pulse is available from the SYNC OUT connector on the front panel. Proper operation of the Model 2414A can be verified by connecting an oscilloscope as shown in Figure 3-1.

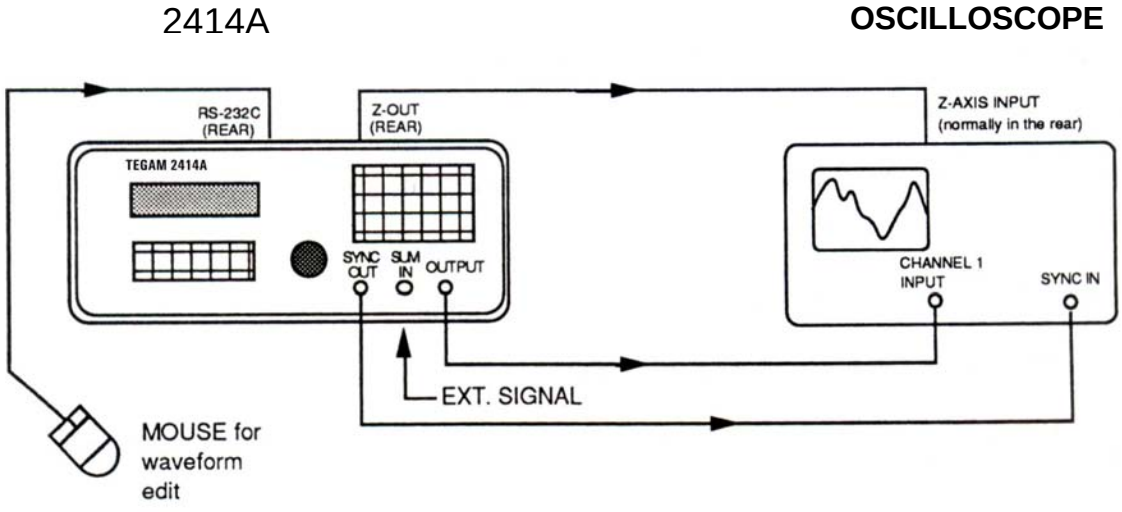



Figure 3-1. Output and Sync Connectors.

## 3.3 DEFAULT SETTINGS

The Model 2414A automatically provides a continuous sine wave signal after the instrument is reset. To reset the 2414A:

1) Press the SHIFT key (green LED will come on).
2) Press RESET (9) key.
3) Press CURR or ALL softkey. (CURR resets all the current settings to the values listed below. ALL additionally resets all waveform and sequence programming and stored settings.)
4) Press OK softkey to reset or CANC softkey to cancel.
5) Press SHIFT key again to restore normal key functions (green LED will go out).

**Reset Default Values**

| | |
|---|---|
| Function: | Sine |
| Mode: | Continuous |
| Amplitude: | 5.000 Volts |
| Offset: | 0.000 V |
| Clock: | 10MHz |
| Output: | Off |
| GPIB Address: | 16 (If Installed) |

## 3.4 HOW TO CHANGE DEFAULT PARAMETERS

The following paragraphs explain how to select waveforms and modes and set different values for amplitude, offset and frequency.

### 3.4.1 Turning Output On

Resetting turns the signal output off. To turn the output on:

1) Press the OUTPUT key.
2) Press the ON softkey.
3) Press the ENTER key. Output LED illuminates.

### 3.4.2 Selecting Standard Waveforms

All waveforms are digitally synthesized. Twenty standard waveforms can be readily recalled from stored algorithms:

1) Press FUNC key.
2) Press STDW softkey.
3) Scroll through selections with arrow softkeys.
4) Select desired waveform with softkey (F1 through F4). Selected item is all capital letters.
5) Press ENTER key.

Several standard waveforms have ancillary functions; such as phase, number of cycles, duty cycle, rise and fall times, etc. These are listed in Table 3-1.

| | ANCILLARY | RANGE & | DEFAULT |
|---|---|---|---|
| FUNCTION | FUNTION | RESOLUTION | VALUE |
| | | | |
| SINE | | | |
| | Phase | 0.000 to 360.000° | 0.000° |
| | Number | 0.01 to 1000.00 | 1.00 |
| SQUARE | | | |
| | Number | 1 to 1000 | 1 |
| | Duty Cycle | 1 to 100% | 50% |
| TRIANGLE | | | |
| | Number | 1 to 1000 | 1 |
| SAWTOOTH | | | |
| | Number | 1 to 1000 | 1 |
| | Duty Cycle | 1 to 100% | 100% |
| DC | | | |
| | Digital Offset | -2048 to +2047 | 0 |
| EXPONENTIAL | | | |
| | Time Constant | 0.01 to 20.00 | 5.00 |
| AM | | | |
| | Carrier Frequency | x1 to x10,000 | x20 |
| | Modulation Frequency | x1 to x10,000 | x1 |
| | Modulation Index | 0 to 200% | 100% |
| | Modulation Phase | 0 to 360° | 0° |
| | Carrier Phase | 0 to 360° | 0° |
| SCM | | | |
| | Carrier Frequency | x1 to x10,000 | x20 |
| | Modulation Frequency | x1 to x10,000 | x1 |
| | Modulation Phase | 0 to 360° | 0° |
| | Carrier Phase | 0 to 360° | 0° |
| FM | | | |
| | Carrier Frequency | x1 to x10,000 | x20 |
| | Modulation Frequency | x1 to x10,000 | x1 |
| | Modulation Index | 0.01 to 100.00 | 10.00 |
| | Modulation Phase | 0 to 360° | 0° |
| | Carrier Phase | 0 to 360° | 0° |
| HAVERSINE | | | |
| | Number | 0.01 to 1000.00 | 1.00 |
| LINEAR SWEEP | | | |
| | Begin | x1 to x1000 | x1 |
| | End | x1 to x1000 | x10 |
| LOG SWEEP | | | |
| | Begin | x1 to x1000 | x1 |
| | End | x1 to x1000 | x10 |
| PULSE | | | |
| | Delay | 0 to 100% | 0% |
| | Rise Time | 0 to 100% | 10% |
| | High Time | 0 to 100% | 30% |
| | Fall Time | 0 to 100% | 10% |
| | Number | 0 to 1000 | 1 |
| GAUSSIAN | | | |
| | Exponent Power | 0.01 to 20.00 | 2.00 |
| SINE X/X | | | |
| | Number | 4.00 to 1000.00 | 5.50 |
| CIRCLE | | | |
| | Number | 0.01 to 1000.00 | 1.00 |
| | Phase | 0.01 to 360.00° | 0.00 |

(NOTE: Typically, a minimum of 3 to 10 samples are required to represent the tabulated functions. Therefore, the length of the waveform must be taken into consideration when selecting range and resolution values.)

Table 3-1. Model 2414A Ancillary Functions

### 3.4.3 Amplitude and Offset

Output signal amplitude can be changed from the default value of 5 volts peak (open circuit) to any value between 10mV and 10.20V (within limits of resolution, see Specifications on page 1-2). Either the edit knob or the numeric keypad may be used to obtain the new value. (For maximum resolution, use the keypad.) Press the AMPL/OFST key to display the value.

**NOTE**

The edit knob and keypad control only the parameter displayed on the right-hand side of the top row of the LCD. Use the double-arrow key to reverse the positions if two parameters are displayed.

DC offset can be changed from the default value of zero to any value between ±10.2 V.

**CAUTION**

To prevent waveform clipping, the combined amplitude and offset must not exceed ±10.2V.

### 3.4.4 Sample Clock and Output Frequency

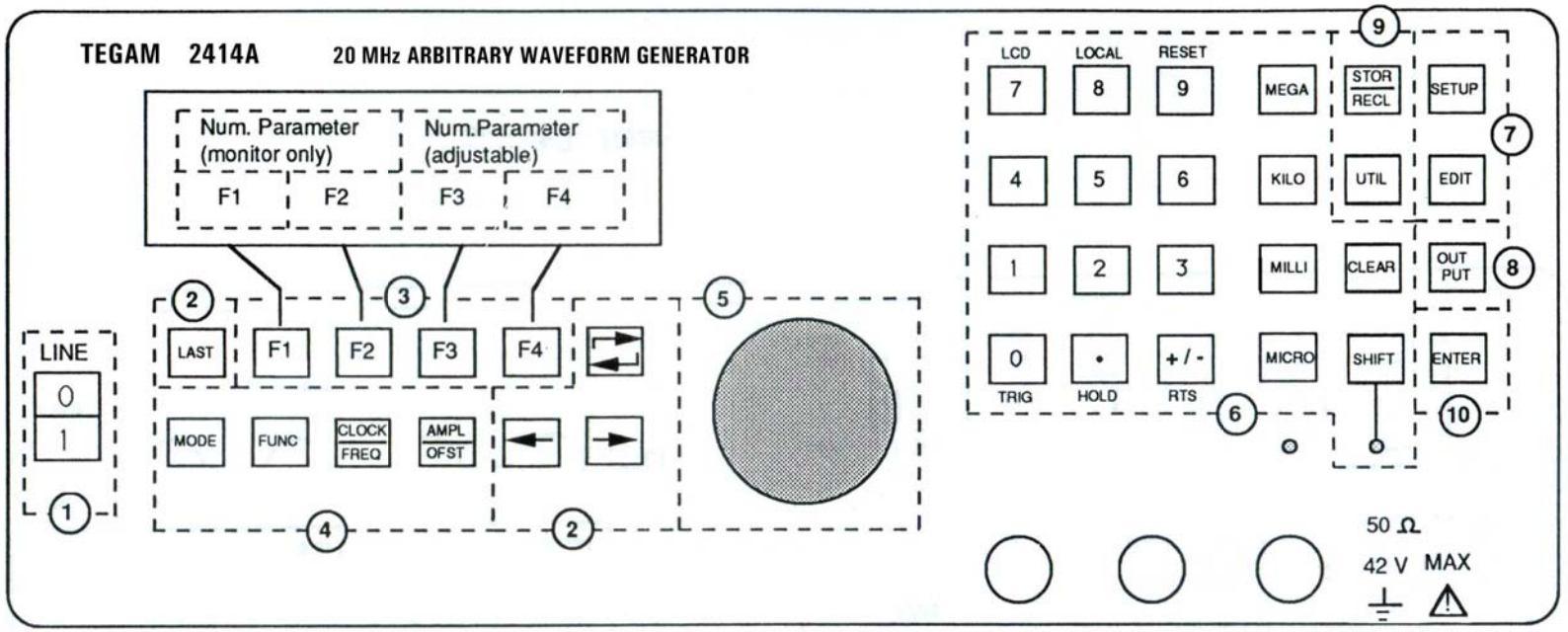
Output waveform frequency is a function of both the clock frequency and the number of samples. (Output f = Clock f/# Samples.) Since the default clock frequency is 10 MHz and the standard waveform default memory allocation is 1000 samples, the default output frequency is 10 MHz/1000 = 10 kHz.

Press the CLOCK/FREQ key. Either the clock frequency (SCLK) or the desired output frequency (FREQ) can be entered directly. Use the double-arrow key to move the selection to the top row of the display. Use the edit knob or keypad to set the desired value. (Any change in the number of waveform samples will also affect the output frequency, as discussed in Section 5.)

### 3.4.5 Modes

Modes are selected by pressing the MODE key. Press it successively to display all five modes. Use the softkeys (F1 through F4) to select a mode, and then press ENTER. Each mode performs as follows:

Continuous: Output runs continuously between selected memory address locations.
Triggered: Output at start point until triggered, then runs once.
Gated: As triggered except output is continuous until gate signal ends.
Burst: Each trigger outputs a pre-programmed number of waveforms from 1 to 1,048,575.
Toggled: Alternate triggers start and stop the output waveform.




1. Power Switch
2. LCD Screen Control Keys
3. Soft Keys
4. Waveform Parameter Keys
5. Edit Knob
6. Numeric Keypad
7. Waveform Setup and Edit Keys
8. Output Control Key
9. Store and Recall Key
10. Enter Key

Figure 3-2. Front-Panel Controls and Indicators

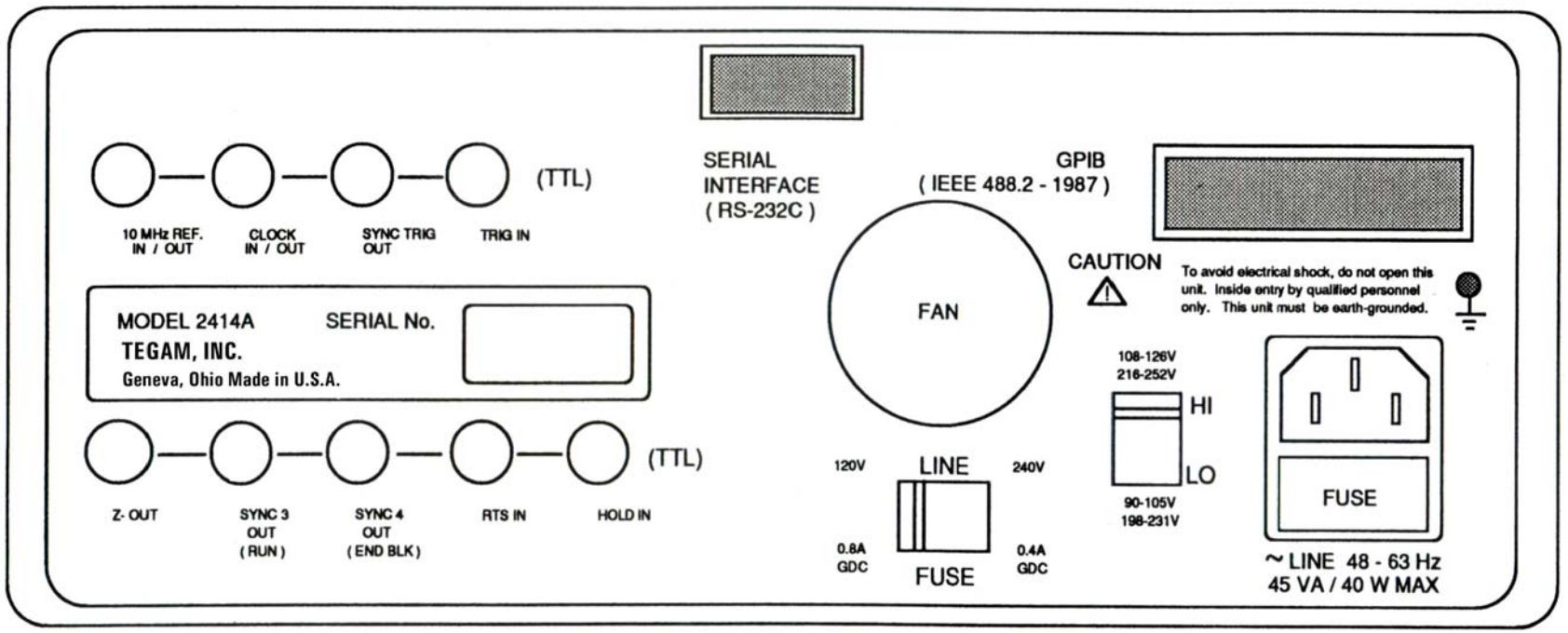



Figure 3-3. Rear-Panel Connectors and Switches

# SECTION 4
# GENERATOR CONCEPT & CONTROL

## 4.1 INTRODUCTION

This section explains how waveforms are digitally synthesized, how the clocking system works and how waveforms are played back.

## 4.2 DIGITAL SYNTHESIS

Both standard and arbitrary waveshapes are created by digitally storing X and Y values in random-access memory (RAM). Standard waveshapes have their X and Y values loaded automatically from stored algorithms. Arbitrary waveshapes are created by downloading X and Y values from a computer or by using the line or vertex edit mode. The line and vertex edit modes allow the X and Y values to be entered manually using the edit knob and/or numeric keypad. Waveform creation can also be accomplished with an optional computer mouse connected to the back of the instrument to position and store X and Y intersects.

Figure 4-1 shows how a waveform is described in RAM memory with a series of **X-Y** values.

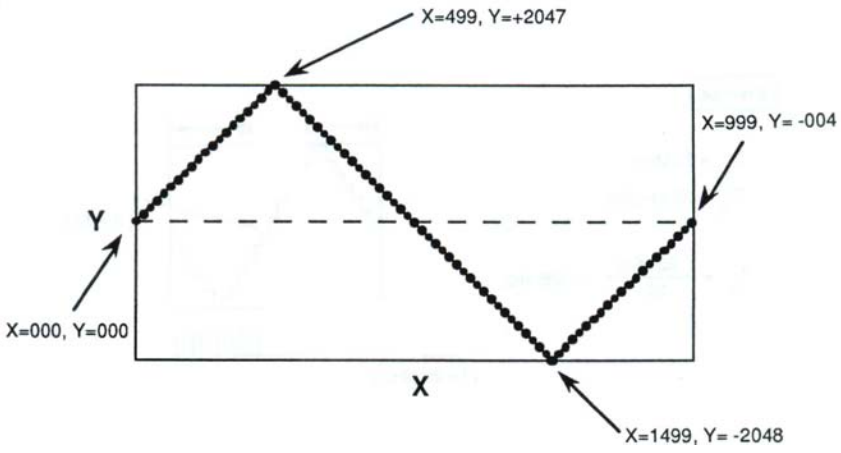



Figure 4-1. Digital Waveform Synthesis

## 4.4 HOW WAVEFORM IS PLAYED BACK

Figure 4-3 shows how the waveform is played back. The digital waveform data stored in the waveform memory is clocked through a D/A converter to create the analog representation of the signal. A low-pass filter can be switched in to remove the sampling noise.

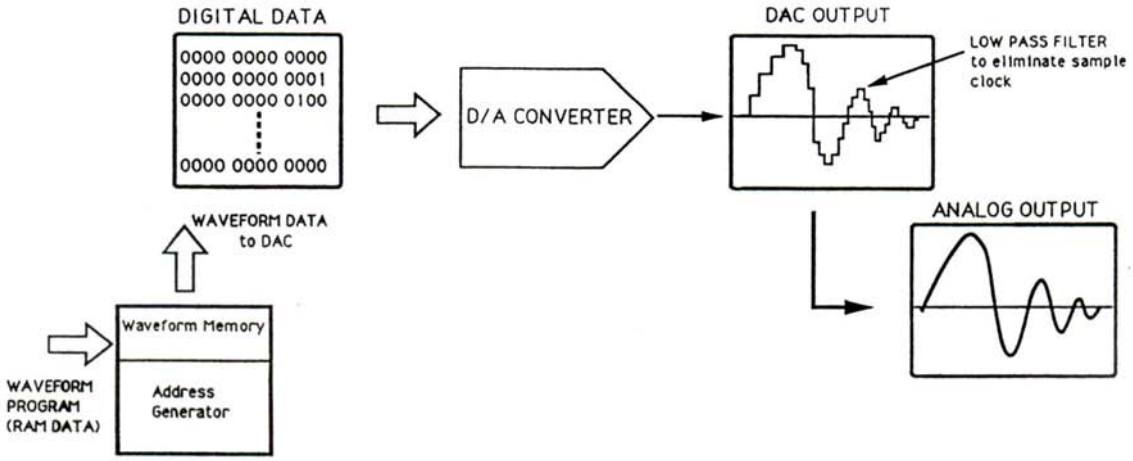



Figure 4-3. Waveform Playback

### 4.4 HOW WAVEFORM IS PLAYED BACK

Figure 4-3 shows how the waveform is played back. The digital waveform data stored in the waveform memory is clocked through a D/A converter to create the analog representation of the signal. A low-pass filter can be switched in to remove the sampling noise.

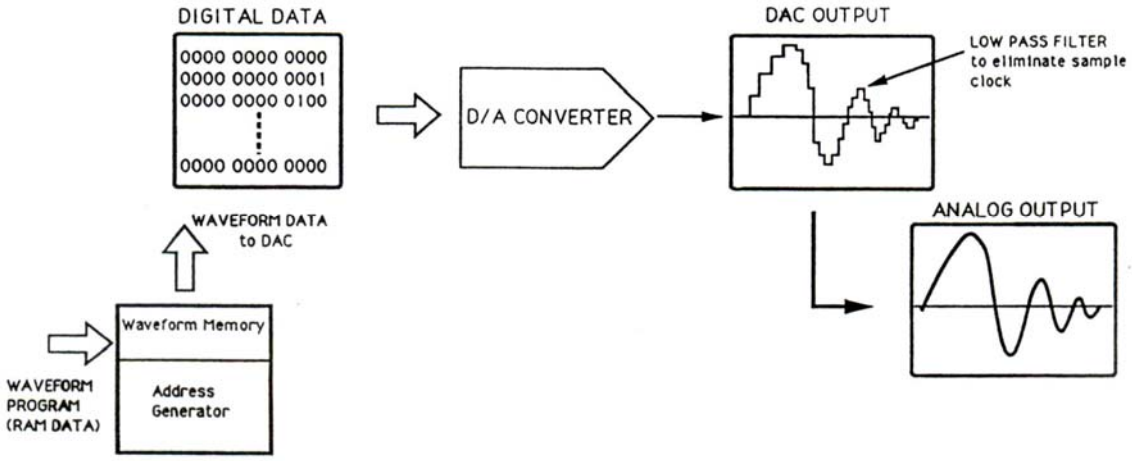



Figure 4-3. Waveform Playback

This page intentionally left blank.

# SECTION 5
# MEMORY ORGANIZATION

## 5.1 INTRODUCTION

This section explains how the waveform memory is organized, the system of waveform numbering, how default memory segments can be changed, and how standard functions can be inserted.

## 5.2 DEFAULT PARTITIONING

The total available waveform memory is 163,808 points. 131,040 points are the active waveform memory, from which the output signals are derived. An additional 32,768 points of buffer memory may be accessed by the RS-232C or optional GPIB remote interface. The buffer memory may be downloaded with new waveform data while signals are outputted from the active memory without interruption. The active memory is initially divided into partitions to provide easy programming of 6 different waveforms. Five blocks of 2000 points each occupy the first 10K of memory. In addition, one of 20 standard waveforms can be readily recalled from the Function menu and downloaded to 1000 points of memory from 10K to UK. (A sinewave is the default Standard Wave.)

## 5.3 WAVEFORM NUMBERING

The first 5 blocks of memory are assigned Waveform Numbers. Additional Waveform Numbers can be assigned (up to total of 1000, depending on the number of points). Waveform Numbers are used to access the memory blocks for initial waveform programming and to recall the waveforms later. Waveform Numbers are arrayed in the memory in ascending order. Figure 5-1 shows the default waveform numbers and partitioning.

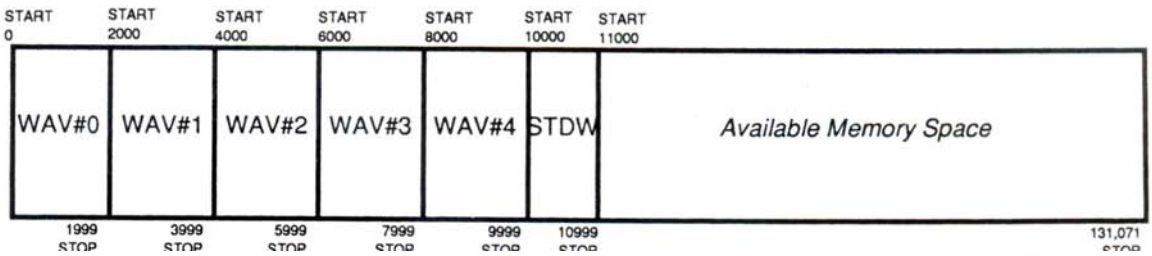



Figure 5-1. Waveform Partitions and Numbers

## 5.4 CHANGING WAVEFORM BLOCK LENGTHS

### 5.4.1 Standard Wave

The standard wave memory can be changed from its default value of 1000 points. (Minimum waveform length is 32 points.)

1) Press SETUP key.
2) Press STDW softkey. The display indicates the present length of the standard wave and the address where it begins in memory.
3) Press LEN softkey. The display indicates the available free memory.
4) Select desired standard wave length with edit knob or keypad. (Do not attempt to exceed the amount of memory available.)
5) Press OK to enter change or CANC to cancel.
6) Press ENTER.

### 5.4.2 Waveform Number Block Lengths

The waveform lengths of the numbered blocks can be changed from their default values of 2000 points. (Minimum waveform length is 32 points.)

1) Press SETUP key.
2) Press WAV# softkey.
3) Select Waveform Number with edit knob. The display indicates the present length and start address of the waveform.
4) Press LEN softkey. The display indicates the available free memory.
5) Select desired waveform block length with edit knob or keypad. (Do not attempt to exceed the amount of memory available.)
6) Press OK to enter change or CANC to cancel.

## 5.5 DELETING WAVEFORMS

A waveform which is no longer needed can be deleted from memory as follows:

1) Press SETUP key.
2) Press WAV# softkey.
3) Select Waveform Number with edit knob.
4) Press DEL softkey.
5) Press OK to delete or CANC to cancel.

## 5.6 INSERTING NEW WAVEFORM NUMBERS

Unused Waveform Numbers can be activated as follows:

1) Press SETUP key.
2) Press WAV# softkey.
3) Press NEW softkey.
4) Screen will show all unused Waveform Numbers when edit knob is turned. Select desired number.
5) Move waveform length to right side of display with double-arrow key and set desired length with edit knob or keypad. *(New waveform cannot be longer than available free memory. See 5.4.2.)*
6) Press OK to insert new Waveform Number or CANC to cancel.

# SECTION 6
# CREATING AND EDITING WAVESHAPES

## 6.1 INTRODUCTION

This section explains how to create arbitrary, non-standard waveshapes. These custom waveforms can be created a segment at a time using line or vertex edit modes. Waveshapes can also be created or modified a point at a time by using the point edit mode. Each step in the waveform construction may be viewed on an oscilloscope connected to the instrument output. The mouse is recommended for line and vertex editing, although it is possible to construct waveforms without it. Arbitrary and standard waveforms can be interspersed. More complex waveforms can be created by adding, subtracting, and multiplying any two standard or arbitrary waveforms that have been previously stored in memory. It is also possible to download waveforms from a computer using the standard RS-232C or optional GPIB interface. In addition, waveforms can be transferred directly from most digital storage oscilloscopes via RS-232C or GPIB.

**NOTE**

Define Waveform Number block length before creating an arbitrary wave shape (see paragraph 5.4.2).

## 6.2 LINE MODE

With the line-editing mode, waveforms are created a segment at a time from a left-hand start or anchor point. A line is drawn from the start point or anchor to a "vertex" point which is positioned to the right. The vertex becomes a new start point and the process is interactively repeated until the new arbitrary waveform is completed. This process is illustrated in Figure 6-1. All or any portion of a selected waveform block can be edited. Editing begins either at a start point or left anchor. A start point can be placed at any X and Y position within the selected waveform block. Alternatively, a left anchor can be positioned at any X address, but the Y value follows that of any previously programmed waveform (or baseline).

### 6.2.1 Editing From Start Point

1) Press EDIT key.
2) Press LINE softkey.
3) Select Waveform Number using edit knob or keypad (press ENTER after using keypad).
4) Arbitrary waveform construction can begin at any point within the selected waveform number block. Press STRT softkey.
5) Use edit knob or keypad to select X and Y addresses to start waveform editing within the selected block. Use double-arrow key to move each active parameter to right side of display. (Remember to press ENTER if keypad is used.)
6) Press OK to store start point or CANC to cancel.
7) Press ANCH softkey.
8) Use double-arrow key to move Right Anchor (AR) to right side of LCD. Use edit knob or keypad to select X address for the end of the edited portion of the waveform.
9) Press OK to store right anchor or CANC to cancel.

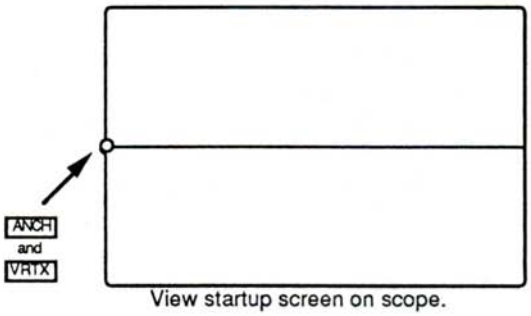
### 6.2.2 Editing From Left Anchor

1) Press EDIT key.
2) Press LINE softkey.
3) Select Waveform Number using edit knob or keypad (press ENTER after using keypad).
4) Arbitrary waveform construction can begin at any X address within the selected waveform number block. Press ANCH softkey.
5) Use double-arrow key to move Left Anchor (AL) to right side of display. Use edit knob or keypad to select X address. (Remember to press ENTER if keypad is used.)
6) Use double-arrow key to move Right Anchor (AR) to right side of LCD. Use edit knob or keypad to select X address for the end of the edited portion of the waveform.
7) Press OK to store anchors or CANC to cancel.

**NOTE**

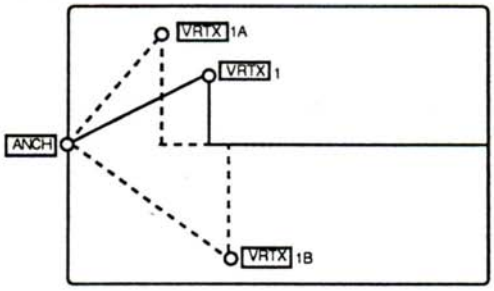
Anchors and vertexes will appear on the oscilloscope screen as intensified points on the edited output waveform. Connect a BNC/BNC cable between the Z-OUT connector on the 2414A rear panel and the Z-AXIS input connector of the oscilloscope. Adjust the oscilloscope intensity for good cursor definition. (If necessary, refer to paragraph 9.3 to adjust the Z-Axis level.)

1. Select LINE mode.



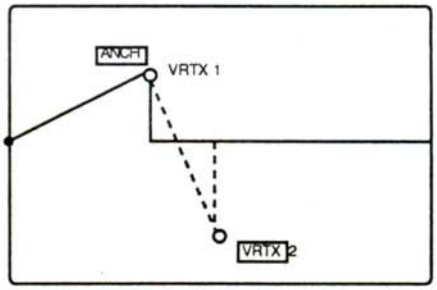

View startup screen on scope.

2. Move VRTX with mouse.




Monitor VRTX coordinate (X,Y) on front panel.
Press LEFT mouse button to anchor VRTX 1.

3. Anchor the vertex and create a new line.




The anchor has automatically moved to a new location. Ready for a new vertex.

4. Continue this process until desired waveform is created.

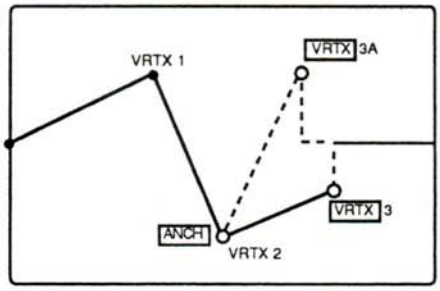



Figure 6-1. Line Mode Waveform Creation and Editing

### 6.2.3 Creating Line Segments

1) Press CHRD (chord) softkey. (Set anchors first per paragraph 6.2.2.)
2) Use edit knob, keypad or optional mouse to select X and Y addresses for the destination of the first line segment (chord). If the mouse is used, LCD readouts will continually indicate mouse position.
3) When the desired position is reached, press OK (or the left mouse button) and the line segment will be stored. Press CANC (or the right mouse button) to cancel.
4) Create the next line segment by again using the knob, keypad or optional mouse to set the next X-Y coordinate, as before.
5) Continue adding line segments up to the limit established by the right anchor.

## 6.3 VERTEX MODE

With the vertex editing mode, waveforms are created by establishing two anchor points at selected addresses, positioning a vertex in the active region between the two anchors and then connecting the vertex to the anchors with two line segments. The vertex mode also permits waveform scaling and the insertion of standard functions. Vertex editing is illustrated in Figure 6-2.

1) Press EDIT key.
2) Press VRTX softkey.
3) Select Waveform Number using edit knob or keypad (press ENTER after using keypad).

### 6.3.1 Selecting Left and Right Anchor Points

1) Press ANCH softkey.
2) Set left anchor X value using edit knob, keypad or optional mouse (press ENTER if keypad is used).
3) Use double-arrow key to move right anchor (AR) to right side of LCD. Set right anchor X value using edit knob, keypad or optional mouse.
4) Press OK to store anchors or CANC to cancel.

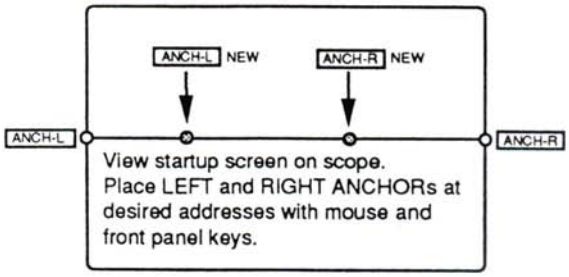
**NOTE**

The difference between the left and right anchors is limited to 8000 points or the waveform length, whichever is less.

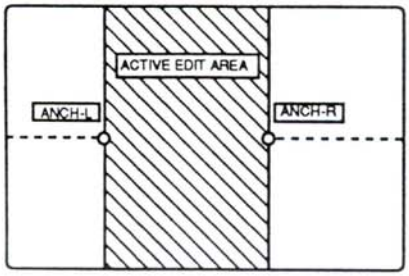
### 6.3.2 Selecting Vertex Point

1) Press ADDV softkey.
2) Set vertex X and Y values using edit knob, keypad or optional mouse.
3) When the desired position is reached, press OK (or the left mouse button) and the two line segments will be stored. Press CANC (or the right mouse button) to cancel.
4) Continue adding anchors and vertices until the waveform is completed.

1. Select VERTEX mode.

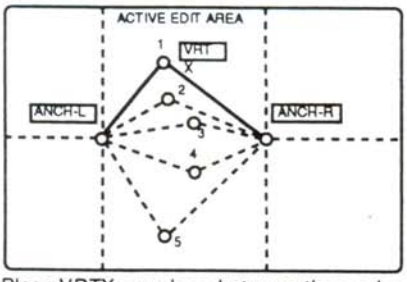



2. Active edit area is defined.



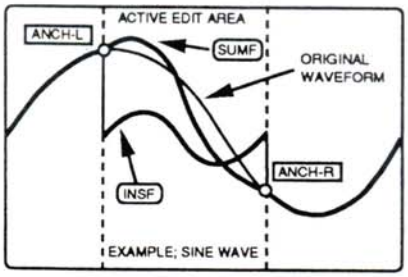

3. Select (a) vertex, (b) sum or insert function, or (c) scale.

(a) VERTEX




Place VRTX anywhere between the anchor points.
After the placement, move the two anchors to a new area.

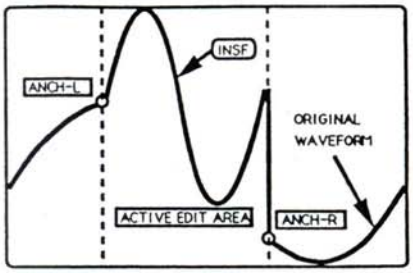
(b) SUM or INSERT FUNCTION




SUMF(sum function) is similar to analog sum.

INSF(Insert Function) cuts the active edit area of the waveform and pastes a new function.

3. (c) scale.




Between the anchor-left and the anchor-right, digitally rescale the waveform with DA (digital amplitude) and DO (digital offset).

Figure 6-2. Vertex Mode Waveform Creation and Editing

### 6.3.3 Scaling

Scaling allows any portion of a waveform designated by the left and right anchors to be scaled in amplitude and offset.

1) Select anchors as in 6.3.1.
2) Press SCAL softkey.
3) Set digital amplitude (DA) and digital offset (DO) values for selected portion of waveform using edit knob, keypad or optional mouse. Observe changes on output oscilloscope. Use double-arrow key to select active parameter.
4) Press OK to store the scaled waveform or CANC to cancel.

**NOTE**

The digital amplitude default value is 4095. The available range of settings is ±8191. A negative setting inverts the edited waveform portion. Re-scaling amplitude from the default value of 4095 to the maximum value of 8191 doubles the output amplitude. Steps 1 - 4 above can be repeated **as** necessary to enlarge small signals.

**CAUTION**

The waveform will be clipped if the scaling factors cause the waveform to exceed the 4095 points available in the waveform memory.

### 6.3.4 Smoothing

A smoothing factor may be applied to any part or all of a waveform. It is computed as a moving average over a specified number of samples.

1) Select anchors as in 6.3.1.
2) Press either the left or right arrow key.
3) Press SMOO softkey.
4) Select the number of samples to be averaged (up to 250) using the edit knob or keypad.
5) Press SHOW softkey to preview.
6) Press CANC softkey to cancel.
7) Press OK softkey to store smoothed waveform.

### 6.3.5 Inserting Standard Functions

Any one of 20 standard functions can be inserted between the left and right anchors.

1) Select anchors as in 6.3.1.
2) Press INSF key.
3) Select desired standard waveshape by pressing softkey. Use left and right arrow keys to view all 20 waveshapes. Available ancillary functions for the selected waveshape may be accessed by pressing an arrow key.

4) Press SHOW softkey to preview selection.
5) Press CANC softkey to cancel selection.
6) Press OK softkey to store selection.

### 6.3.6 Summing Standard Functions

Any one of the 20 standard functions can be algebraically summed to any part or all of any other standard function.

1) Select the first standard waveshape as in 6.3.5.
2) If the second standard waveshape is to be summed to only a portion of the first standard waveshape, reposition the anchors as described in paragraph 6.3.1.
3) Press either the left or right arrow key.
4) Press SUMF softkey.
5) Select desired standard waveshape by pressing softkey. Use left and right arrow keys to view all 20 waveshapes. Access any desired ancillary functions by pressing the arrow keys. *(Reduce the digital amplitude value as necessary to prevent clipping.)*
6) Press SHOW softkey to preview summed waveforms.
7) Press CANC softkey to cancel.
8) Press OK softkey to store summed waveforms.

### 6.3.7 Dump Function

Dump Function permits a standard waveform to be conveniently loaded into the entire length of a Waveform Number without specifying left and right anchors. This also permits standard waveforms to be inserted in Waveform Numbers with lengths greater than 8000 points.

1) Select the desired Waveform Number as in 6.3.
2) Press either the left or right arrow key.
3) Press DMPF softkey.
4) Select desired standard waveshape by pressing softkey. Use left and right arrow keys to view all 20 waveshapes. Access any desired ancillary functions by pressing the arrow keys.
5) Press DO softkey to store the waveform in memory and permit further changes or press OK softkey to store waveform and return to previous menu.

**CAUTION**

It is not possible to preview a function when using Dump Function. Both the DO and OK commands above will cause the new waveform to write over any previous waveforms.

### 6.3.8 Move

The Move commands allow a section of a waveform as defined by the left and right anchors to be copied and pasted into another section of the same or another Waveform Number.

1) Select the desired Waveform Number as in 6.3.
2) Press either the left or right arrow key.
3) Press MOVE softkey.
4) Set left and right anchors to the waveform section to be copied.
5) Press COPY softkey.
*6) Press LAST key.
*7) Set Waveform Number to receive pasted section.
*8) Press MOVE softkey.
9) Set left and right anchors to the destination waveform section.
10) Press PSTE softkey.

** These steps may be omitted if copy and paste are within the same Waveform Number.*

NOTE

The pasted waveform will be truncated if the destination waveform section has fewer points than the original.

## 6.4 POINT MODE

With the point-editing mode, waveforms are created or modified a point at a time.

1) Press EDIT key.
2) Press PNTS softkey.
3) Select Waveform Number using edit knob or keypad (press ENTER after using keypad).

### 6.4.1 Selecting Left and Right Anchor Points

1) Press ANCH softkey.
2) Set left anchor X value using edit knob, keypad or optional mouse (press ENTER if keypad is used).
3) Use double-arrow key to move right anchor (AR) to right side of LCD. Set right anchor X value using edit knob, keypad or optional mouse.
4) Press OK to store anchors or CANC to cancel.

**NOTE**

The difference between the left and right anchors is limited to 8000 points or the waveform length, whichever is less.

### 6.4.2 Entering Point Values

1) Press MODP softkey.
2) Set the X and Y coordinates using the edit knob or keypad. Press OK to store each point value.
3) Continue adding coordinates until the desired waveshape is obtained. The PX- and PX+ softkeys may be used to decrement or increment the X address.

## 6.5 MATH OPERATIONS

Math operations permit the contents of any two Waveform Numbers of equal size to be algebraically added, subtracted or multiplied together. Complex composite signals can thus be created, such as shaped tone bursts, amplitude modulation, etc. To enter the Math Mode:

1) Press EDIT key.
2) Press MATH softkey.

### 6.5.1 Selecting Math Function

The three math functions have the following forms:

A*B (Multiply; output amplitude normalized to full-scale waveform memory.)
A+B (Add; output amplitude divided by two.)
A-B (Subtract; output amplitude divided by two.)

To select one of the math functions:

1) Press OP softkey. .
2) Press the softkey for the desired math function. The selected function will be capitalized.

See paragraph 6.6.2 for a detailed example.

### 6.5.2 Selecting Waveform Numbers

Waveform Numbers to be combined must be selected and a destination assigned for the combined waveform. (To assign a destination Waveform Number, refer to paragraph 5.6.)

1) Press the arrowhead [◀] softkey.
2) The displayed equation has the following form:
   Destination Waveform ## = Waveform ## (*, + or -) Waveform ##.
3) Define each Waveform Number by moving the arrowhead to each location in the equation and selecting a Waveform Number with the edit knob or keypad (followed by ENTER).
4) Press the DO softkey after the three Waveform Numbers have been assigned. The result is visible at the output.

**CAUTION**

Be sure only equal-length waveform blocks are combined using a math function. The destination waveform may be larger.

## 6.6 EXAMPLES

Waveform editing in the Model 2414A is so flexible that often the same complex waveform can be created several different ways. For example, let us construct a waveform described by the equation: A = sin cot + 1/6 sin Scot. (It is presumed that the waveform lengths are at the default values of LEN = 2000.)

### 6.6.1 Insert and Sum Functions

1) Press SETUP key.
2) Press WAV# softkey.
3) Press NEW softkey to select a new Waveform Number.
4) Press OK softkey.
5) Press EDIT key.
6) Press VRTX softkey. Set Waveform Number to that selected in step 3.
7) Press INSF softkey.
8) Press SIN softkey.
9) Set Phase = 0.000 and Number = 1.00.
10) Press arrow key until Digital Offset (DO) and Digital Amplitude (DA) are displayed.
11) Set DO = 0 and DA = 4095. (A full-amplitude signal is obtained with DA = 4095.)
12) Press OK softkey. Observe that the fundamental frequency (sin cot) is stored in memory and uses maximum vertical resolution capacity at the waveform peaks, as shown in Figure 6-3a.
13) Press either left or right arrow key.
14) Press SUMF softkey.
15) Press SIN softkey.
16) Set Phase = 0.000 and Number = 3.00.

17) Press arrow key until Digital Offset (DO) and Digital Amplitude (DA) are displayed.
18) Set DO = 0 and DA = 683. (Scaling of the amplitude adjusts for the 1/6 amplitude coefficient.)
19) Press SHOW softkey to preview the results and monitor the output for the desired signal. (Care must be exercised to limit the final waveforms to be within the memory limits.)
20) Press OK softkey.
21) Observe the resulting composite waveform at the output, as shown in Figure 6-3b.

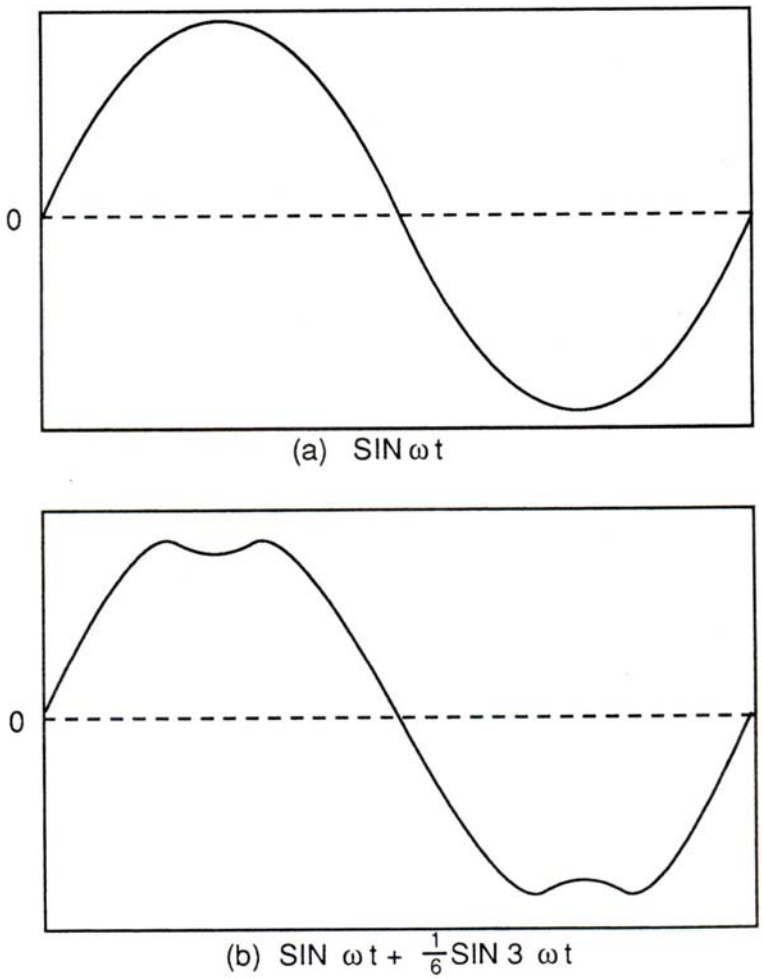



Figure 6-3. Insert and Sum Waveform Creation

### 6.6.2 Math Function

1) Press SETUP key.
2) Press WAV softkey.
3) Use existing Waveform Numbers or press NEW and OK softkeys three times to select three new Waveform Numbers. (As in the previous example, the waveform length is presumed to be LEN = 2000.)
4) Press EDIT key.
5) Press VRTX softkey.
6) Select first Waveform Number from step 3.
7) Press INSF softkey.
8) Press SIN softkey.
9) Set Phase = 0.000 and Number = 1.00.
10) Press arrow key until Digital Offset (DO) and Digital Amplitude (DA) are displayed.
11) Set Digital Offset = 0 and Digital Amplitude = 4095.
12) Press OK softkey and observe fundamental output as shown in Figure 6-4a.
13) Select second Waveform Number from step 3.
14) Press INSF softkey.
15) Press SIN softkey.
16) Set Phase = 0.000 and Number = 3.00.
17) Press arrow key until Digital Offset (DO) and Digital Amplitude (DA) are displayed.
18) Set Digital Offset = 0 and Digital Amplitude = 683 (4095/6).
19) Press OK softkey and observe third harmonic as shown in Figure 6-4b.
20) Press LAST key.
21) Press MATH softkey.
22) Setup the equation so that the third Waveform Number from step 3 equals the sum of the first waveform (step 6) and the second waveform (step 13).
23) Press DO softkey.
24) Press FUNC key.
25) Press WAV# softkey.
26) Select third Waveform Number.
27) Observe composite waveform as shown in Figure 6-4c. Also note the result of the math summing operation is to multiply the signal by 1/2.
28) Press EDIT key and perform the following four steps to increase the amplitude to the level attained in the exercise in paragraph 6.6.1:
29) Press VRTX softkey.
30) Press SCAL softkey.
31) Set Digital Offset = 0 and Digital Amplitude =8191.
32) Press OK softkey.

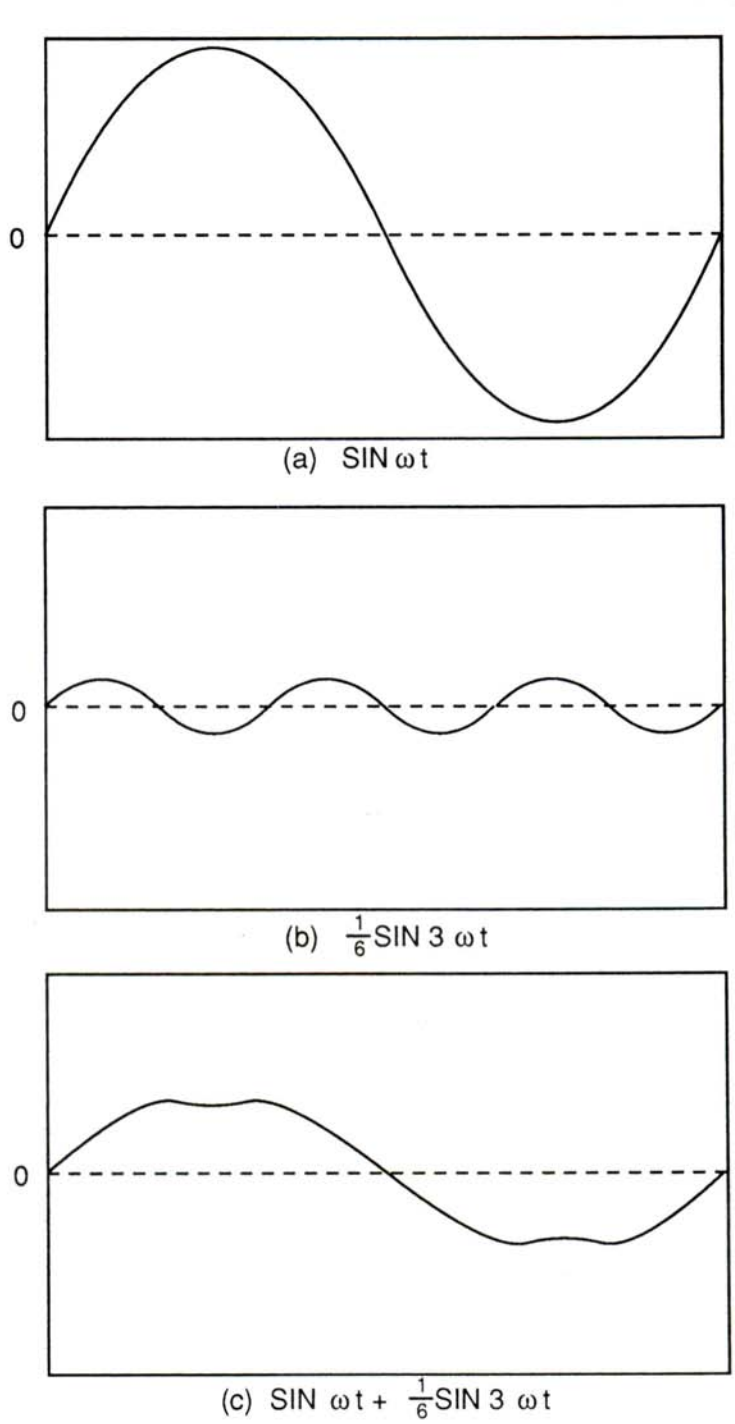



Figure 6-4. Math Function Waveform Creation

This page intentionally left blank.

# SECTION 7
# SEQUENCE GENERATOR

## 7.1 INTRODUCTION

This section explains how to program and use the optional Sequence Generator. The Sequence Generator permits different waveforms to be repeated and/or linked with each other in any order. The list of programmed instructions for each "loop and link" series is called a Sequence. Up to 100 different Sequences can be programmed. All sequences combined can have up to 1000 steps. Each step defines one Waveform Number (up to 1000) and the number of times it repeats (up to 1, 048, 575). Figure 7-1 shows a typical sequence of waveforms. Table 7-1 is a Sequence Programming Worksheet with sample entries corresponding to Figure 7-1. A blank worksheet suitable for reproduction is located in the Appendix.

## 7.2 PROGRAMMING A SEQUENCE

Before programming a sequence, be sure each desired waveform has been created and stored in a Waveform Number location. In addition, prepare a programming worksheet in a manner similar to Table 7-1. Then, follow this procedure:

1) Press SETUP key.
2) Press SEQ# softkey.
3) Press NEW softkey.
4) Select desired new sequence file number with edit knob or keypad.
5) Press OK to enter number or CANC to cancel.
6) Press OPEN softkey.
7) Press ADDS softkey.
8) Select first step number with edit knob or keypad. *(To allow for future changes to the program, it is a good idea to leave room between step numbers, i.e., 10, 20, 30 .....etc.)*
9) Press OK to enter number or CANC to cancel.
10) Select desired Waveform Number for this step with edit knob or keypad.
11) Press double-arrow key to move burst number to right side of display.
12) Set desired number of waveform repetitions with edit knob or keypad.
13) Press OK to enter numbers or CANC to cancel.
14) Select next step number, repeating steps 7 through 12 above.
15) After all steps have been programmed, view finished results by selecting Sequence Number in Function menu and pressing ENTER. Additional sequences can be programmed and stored by selecting a different sequence file number in the steps above.

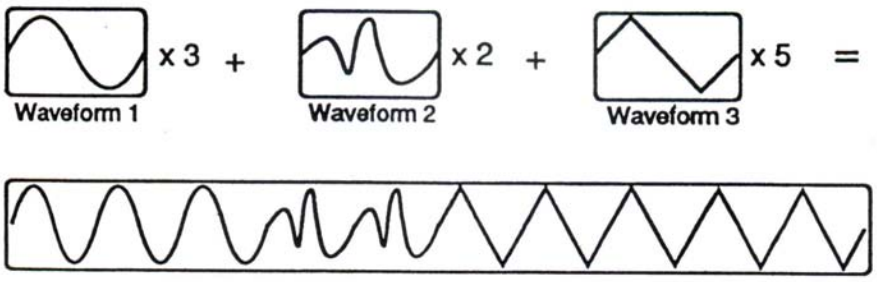



Figure 7-1. Typical Waveform Sequence

**WAVEFORM/SEQUENCE WORK SHEET**

TEGAM 2414A

STANDARD WAVEFORM

SEQUENCE# 10

APPLICATION Demo

| | LENGTH |
|---|---|
| STDW | 1000 |

CUSTOM WAVEFORMS

| WAV # | LENGTH | TYPE |
|---|---|---|
| 1 | 2000 | Sine |
| 2 | 2000 | Arbitrary |
| 3 | 2000 | Triangle |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| STEP # | WAV # | NUMBER of BURST |
|---|---|---|
| 10 | 1 | 3 |
| 20 | 2 | 2 |
| 30 | 3 | 5 |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

Table 7-1. Sequence Programming Worksheet

### 7.3 DELETING A SEQUENCE

Sequences no longer required may be deleted from memory by the following procedure:

1) Press SETUP key.
2) Press SEQ# softkey.
3) Select sequence file number to be deleted with edit knob or keypad. .
4) Press DELF softkey.
5) Press OK to delete file or CANC to cancel.

### 7.4 ADDING A STEP TO AN EXISTING SEQUENCE

To add a step to an existing sequence:

1) Press SETUP key.
2) Press SEQ# softkey.
3) Select appropriate sequence file number.
4) Press OPEN softkey.
5) Press ADDS softkey.
6) Select desired new step number with edit knob or keypad..
7) Program new step as in paragraph 7.2, steps 8 to 13.

### 7.5 DELETING A STEP FROM AN EXISTING SEQUENCE

1) Press SETUP key.
2) Press SEQ# softkey.
3) Select appropriate sequence file number.
4) Press OPEN softkey.
5) Select step number to be deleted with edit knob or keypad.
6) Press DELS softkey.
7) Press OK to delete step or CANC to cancel.

### 7.6 MODIFYING A STEP WITHIN AN EXISTING SEQUENCE

1) Press SETUP key.
2) Press SEQ# softkey.
3) Select appropriate sequence file number.
4) Press OPEN softkey.
5) Select step number to be modified with edit knob or keypad.
6) Press MODS softkey.
7) Select desired Waveform Number for this step with edit knob or keypad.
8) Press double-arrow key to move burst number to right side of display.
9) Set desired number of waveform repetitions with edit knob or keypad.
10) Press OK to enter numbers or CANC to cancel.

# SECTION 8
# MULTIPLE UNITS

## 8.1 INTRODUCTION

Multiple Model 2414A's can be operated synchronously in parallel or series. Synchronous operation of multiple units eliminates triggering jitter and minimizes clock delays. In synchronous operation, the units must share the same sample clock and be programmed for a synchronous trigger interconnect. Refer to Figures 8-1 and 8-2 for diagrams of parallel and series operation.

## 8.2 PARALLEL OPERATION

Parallel operation is appropriate for applications requiring multi-phase signals, X and Y sweeps, etc.

### 8.2.1 Clock Connections

To operate multiple units in parallel, designate one unit as the master unit. Connect the rear-panel CLOCK IN/OUT signal from the master unit to the CLOCK IN/OUT connectors of the remaining slave units. Program the slave unit clock inputs to be external as follows: *(The master unit operates with its normal internal clock.)*

1) Press UTIL key.
2) Press SCLK softkey.
3) Press EXT softkey to select external sample clock.
4) Press ENTER.

### 8.2.2 Trigger Connections

Connect the rear-panel SYNC TRIG OUT signal from the master unit to the TRIG IN connectors of the slave units. Program the master unit sync trigger for parallel operation as follows:

1) Press OUTPUT key.
2) Press an arrow key.
3) Press STRG softkey.
4) Press PAR softkey.
5) Press ENTER.

Program the slave unit trigger inputs for synchronous operation as follows:

1) Press UTIL key repeatedly until Trigger In (TGIN) appears.
2) Press TGIN softkey.
3) Press SYNC softkey to select synchronous trigger input.
   (Press ASNC to return to asynchronous mode wh*en returning to single-unit operation.)*

When the master unit is operated in the continuous mode, synchronize the units:

1) Press SHIFT key on the master unit.
2) Press TRIG key.

## 8.3 SERIES OPERATION

Series operation is especially appropriate for complex signal sequences requiring extra-long memory.

### 8.3.1 Clock Connection

To operate two units in series, designate one unit as the master unit. Connect the rear-panel CLOCK IN/OUT signal from the master unit to the CLOCK IN/OUT connector of the remaining slave unit. Program the slave unit clock input to be external as follows:

1) Press UTIL key.
2) Press SCLK softkey.
3) Press EXT softkey to select external sample clock.
4) Press ENTER.

### 8.3.2 Trigger Connection

Connect the rear-panel SYNC TRIG OUT signal from the master unit to the TRIG IN connector of the slave unit. Program the master unit sync trigger to be serial as follows:

1) Press OUTPUT key.
2) Press an arrow key.
3) Press STRG softkey.
4) Press SER softkey.
5) Press ENTER.

Program the slave unit trigger input to be synchronous as follows:

1) Press UTIL key repeatedly until Trigger In (TGIN) appears.
2) Press TGIN softkey.
3) Press SYNC softkey to select synchronous trigger input.
   *(Press ASNC to return to asynchronous mode when returning to single-unit operation.)*

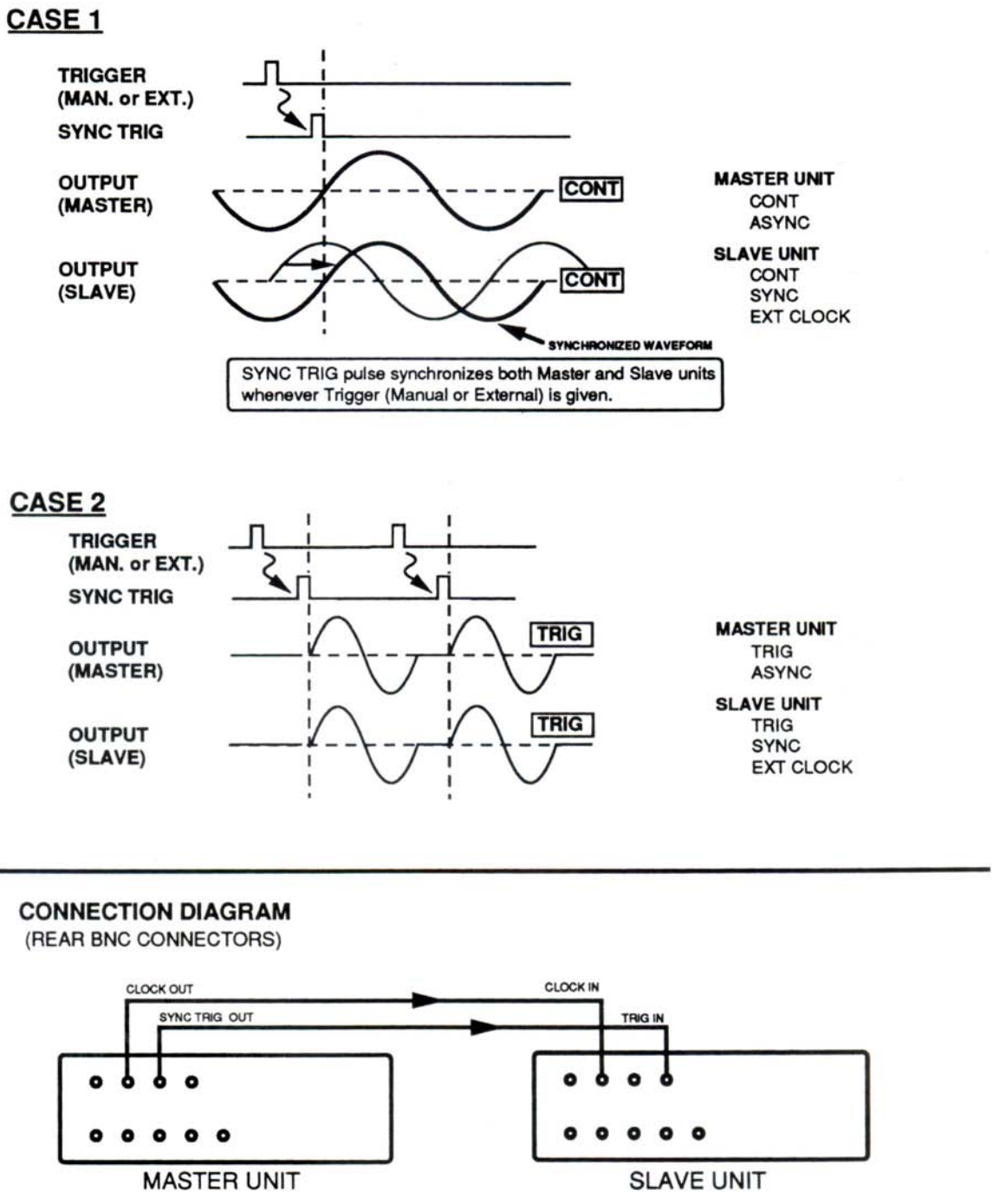



Figure 8-1. Parallel Operation

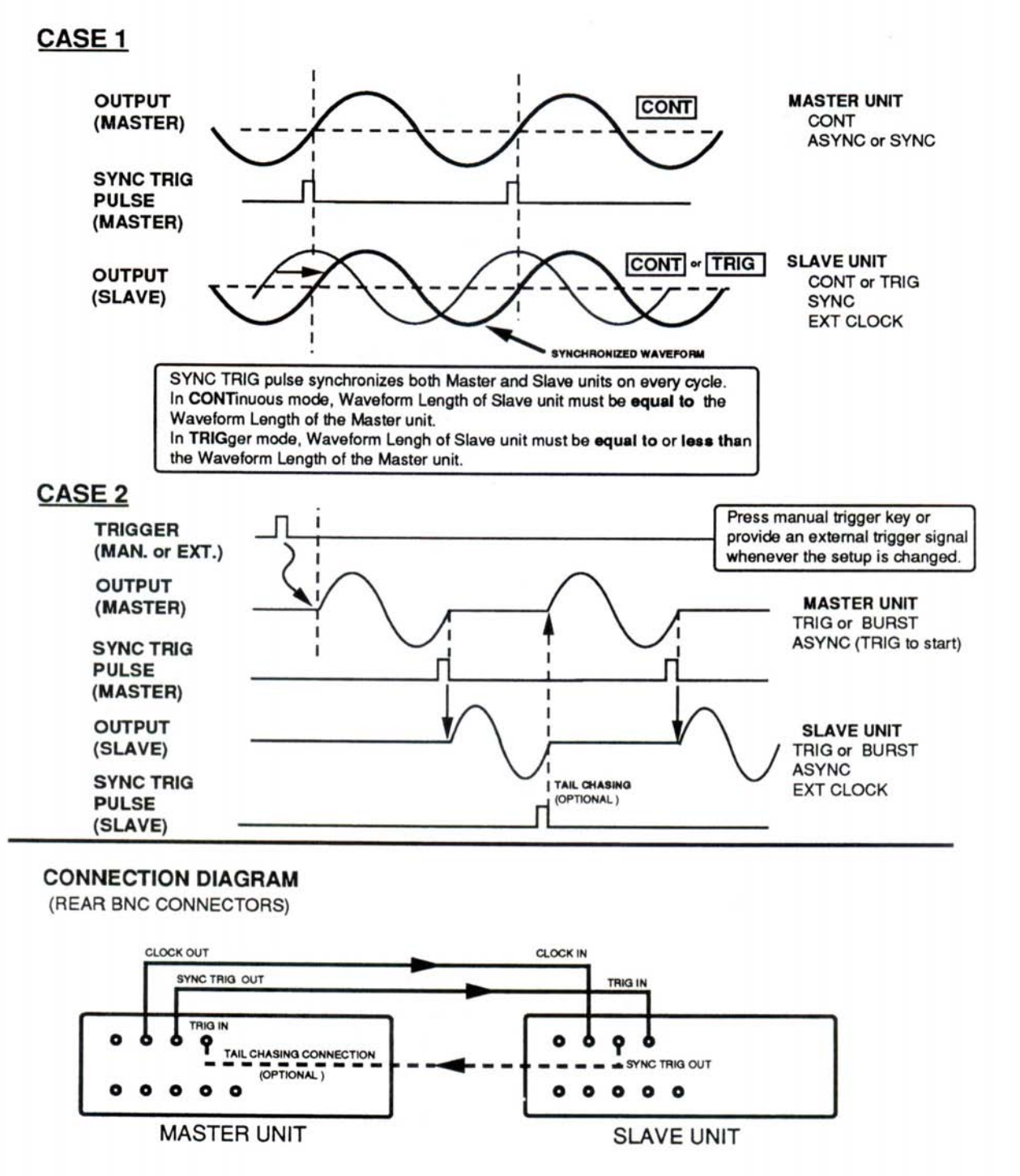



Figure 8-2. Series Operation

# SECTION 9
# OTHER FEATURES

## 9.1 VIEW FUNCTION

The View function allows all or any three segments of the 128K active waveform memory to be viewed at the output. Segments of memory are selected by programming start and stop X addresses, rather than Waveform Numbers. Thus, several consecutive waveforms occupying different Waveform Numbers can be viewed together. To use the View function:

1) Press FUNC key.
2) Press VIEW softkey.
3) Select Segment 1, 2 or 3.
4) Set Left view address with edit knob or keypad.
5) Use arrow key to reverse positions and set Right view address.
6) Press ENTER key.
7) Set addresses for other segments in the same manner, if desired.

The ALL softkey may be used to view the entire 131,072 points in memory.

## 9.2 SYNC OUTPUTS

The Model 2414A provides three separate sync output signals, as listed in Table 9-1.

| SYN | LOCATION | FUNCTION |
|---|---|---|
| SYNC OUT | Front Panel | End Pulse or Programmable Address |
| SYNC 3 OUT | Rear Panel | Run or Programmable Address |
| SYNC 4 OUT | Rear Panel | End Block or Programmable Address |

Table 9-1. Sync Outputs

Each of the sync outputs has a unique default function, as listed in Table 9-1. In lieu of the default function, each may be programmed to provide a sync output pulse at any address and for any length for each waveform stored in the memory. Selections of the sync functions are made in the Output Menus. Programmable sync address and length settings are made in the Setup Menus. The default sync address is 0 and the default sync pulse length is one clock period.

**NOTE**

Programmable syncs may need to be re-programmed if they are in a section of memory deleted when the length of a Waveform Number is shortened.

### 9.2.1 End Pulse

End Pulse is the normal output from the front-panel SYNC OUT connector. It is a TTL level which is high during the last clock interval of the output waveform. In continuous and triggered modes, it is at the end of each cycle. In gated and burst modes, it is at the end of the last cycle. With the optional sequence mode, the End Pulse occurs at the end of the sequence.

### 9.2.2 Run

Run is the normal output from the rear-panel SYNC 3 OUT (RUN) connector. It is a TTL level, which is high whenever an output signal is present.

### 9.2.3 End Block

End Block is the normal output from the rear-panel SYNC 4 OUT (END BLK) connector. It is used only with the optional Sequence Generator to provide a TTL level which is high during the last clock interval at the end of each step in the sequence.

## 9.3 Z-AXIS LEVEL

Z-Axis is the output from the rear-panel Z-OUT connector. It provides a variable level pulse at times coincident with the cursor and anchor positions when in the waveform edit modes. When connected to the Z-Axis input of the monitor oscilloscope, it provides intensity modulation of the display to show the cursor and anchor positions. The level setting depends upon the sensitivity of the oscilloscope. To adjust the Z-Axis level:

1) Press EDIT key.
2) Press LINE softkey.
3) Press ZLVL softkey.
4) Use edit knob or keypad to set cursor and anchor intensity to desired level.
   (Adjust scope trace intensity at the same time for best results.)

## 9.4 OUTPUT FILTER

A 7-MHz, 7th-order low-pass filter may be switched into the output signal circuit. This filter is effective in removing sampling step noise when the maximum clock frequency of 20 MHz is used. The default setting for the output filter is OFF. To enable the output filter:

1) Press OUTPUT key.
2) Press FLTR softkey (capital letters = ON; lower-case letters = OFF).
3) Press ENTER key.

## 9.5 INTERNAL TRIGGER GENERATOR

The triggered, toggled and burst modes require an external, manual or internal trigger. The internal trigger generator provides a periodic trigger at a variable rate from .02 to 10 seconds. To select the internal trigger:

1) Press UTIL key.
2) Press TGEN softkey.
3) Press ON softkey.
4) Set desired trigger interval with edit knob or keypad. Press ENTER key if keypad is used.

**NOTE**

The internal trigger generator provides only short pulses, which are not usable for gated mode.

**CAUTION**

Remember to turn the internal trigger generator OFF when it is no longer needed.

## 9.6 RTS

Return To Start interrupts the output signal and returns the output signal back to the start level. RTS may be implemented by applying a TTL level to the rear-panel RTS IN connector or by pressing the SHIFT and RTS keys on the front panel.

## 9.7 HOLD

Hold stops the output signal and holds it at its present level while applied. Hold may be implemented by applying a TTL level to the rear-panel HOLD IN connector or by pressing the SHIFT and HOLD keys on the front panel.

## 9.8 MONITOR BURST COUNT

In the burst mode, the output cycle count can be monitored at any time. This is most appropriate for slow, low-frequency signals. To monitor the burst count:

1) Press UTIL key twice.
2) Press MBST softkey.

Each time the MBST softkey is pressed the counter is updated and displays the burst count at that time.

### 9.9 DSOLINK™

DSOLink™ permits waveforms, which have been captured by a digital storage oscilloscope (DSO) to be directly loaded into the active waveform memory. Either the RS-232 or GPIB interface may be used, depending on the DSO interface. Almost any oscilloscope with a remote interface which supports an HPGL plotter can be used.

#### 9.9.1 RS-232 Waveform Generator Setup

1) Press UTIL key.
2) Press R232 softkey.
3) Press DSO softkey and select Waveform Number in which to store the DSO waveform.
4) Use the appropriate softkeys to check the settings of the baud rate, parity, bits and handshake. Make sure the settings of the generator and the DSO match.
5) Press TYPE softkey and select the appropriate DSO number, if known (see table 9-2).
6) Press ENTER.
7) Press PLOT or HARD COPY on the DSO.
8)If the DSO number is not known, try each one in turn until the waveform appears at the output. (DSO numbers correspond to pen numbers.)

Refer to the Appendix for specific setup procedures for selected DSO's.

#### 9.9.2 GPIB Waveform Generator Setup

1) Press UTIL key.
2) Press GPIB softkey.
3) Select the Waveform Number in which to store the DSO waveform.
4) Select the appropriate DSO number, if known (see table 9-2).
5) Press ENTER.
6) Press PLOT or HARD COPY on the DSO.
7) If the DSO number is not known, try each one in turn until the waveform appears at the output.

Refer to the Appendix for specific setup procedures for selected DSO's.

| | |
|---|---|
| Biomation | |
| DS04080 | DS01 |
| Hewlett-Packard | |
| 54510B | DS01 |
| 54600 | None* |
| *(Incompatible Format, Axes Rotated 90°) | |
| Hitachi | |
| VC-6155 | DS03 |
| Philips | |
| PM3375 | DS01 |
| Tektronix | |
| TEK2232 | DS02 |
| TEK2440 | DS02 |

Table 9-2. Supported Oscilloscopes

### 9.9.3 DSO Setup

Refer to the DSO manual for specific instructions on how to place the DSO in the PLOT or HARD COPY mode. It will need to output HPGL (or HP7475) plotter commands. For RS-232, make sure the proper cable is used and all the settings match. For GPIB, set the DSO to TALK ONLY. (NOTE: Since the DSO waveforms are formatted for a plotter, superfluous data may appear in the margins.)

## 9.10 WaveWorks Pro™ SOFTWARE

Optional waveform creation software for the Model 2414A is provided by WaveWorks Pro Software. WaveWorks Pro™ consists of a PC program disk, RS-232 cable and an instruction manual. The software features the following:

1) 30 Standard Waveforms
2) 20 Math Transfer Functions
3) 13 Math Operations
4) Digital Pattern
5) FFT and IFFT
6) Sequence Programming

Refer to the WaveWorks Pro™ data sheet in the Appendix at the back of this manual.

This page intentionally left blank.

# SECTION 10
# RS-232C & GPIB

## 10. 1 INTRODUCTION

This section describes how to remotely control all instrument functions and how to download and upload waveform data using either the standard RS-232C or optional GPIB interface. Included are an overview of both methods of remote control, a complete tabulation and explanation of control commands, and several programming examples.

## 10.2 QUICK REFERENCE

This command summary is provided as a quick reference and overview of the complete command list for the Model 2414A. For a detailed explanation, including command syntax, parameters and data limits see Section 10.5. For convenience, commands are listed here in the same order as in Section 10.5.

| **Command** | | **Command** | |
|---|---|---|---|
| **Long Form** | **Short Form** | **Long Form** | **Short Form** |
| Common Commands | | System Commands | |
| *CLS | | EXECUTE | EXEC |
| *ESE (GPIB only) | | HOLD | |
| *ESE? (GPIB only) | | RECALL | RCLL |
| *ESR? | | REF_CLK_ADJ | RADJ |
| *IDN? | | REF_CLK_ADJ? | RADJ? |
| *OPC (GPIB only) | | REF_CLOCK | RCLK |
| *OPC? | | REF_CLOCK? | RCLK? |
| *RST | | RTNTOSTRT | RTST |
| *SRE (GPIB only) | | RESET | |
| *SRE? (GPIB only) | | SAMPLECLOCK | SCLK |
| *STB? (GPIB only) | | SAMPLECLOCK? | SCLK? |
| *OPT? | | STORE | STOP |
| *TRG | | TGENERATOR | TGEN |
| *TST? | | TGENERATOR? | TGEN? |
| *WAI | | TGRRATE | TGRR |
| | | TGRRATE? | TGRR? |
| Configuration Commands | | TRIGGER | TRIG |
| CONFIGURE | CONF | | |
| HEADERS | HDRS | | |
| HEADERS? | HDRS? | | |

| Command Long Form | Short Form |
|---|---|
| Output Commands | |
| AMPLITUDE | AMPL |
| AMPLITUDE? | AMPL? |
| BURST | CIRCLE |
| BURST? | COPY |
| CLOCK_SEL | CLKSEL |
| CLOCK_SEL? | CLKSEL? |
| FILTER | FM |
| FILTER? | FREE? |
| FREQUENCY | FREQ |
| FREQUENCY? | FREQ? |
| FUNCTION | FUNC |
| FUNCTION? | FUNC? |
| MODE | LOGSWEEP LOGS |
| MODE? | NOISE |
| OFFSET | OFST |
| OFFSET? | OFST? |
| OUTPUT_SWTCH | OUTSW |
| OUTPUT_SWTCH? | OUTSW? |
| READ_BURST? | RBRS? |
| SYNCSEL | SYSEL |
| SYNCSEL? | SYSEL? |
| TRGINMODE | |
| TRGINMODE? | |
| TRGOUTMODE | |
| TRGOUTMODE? | |
| RS-232 Specific Commands | |
| *GTR_RS232 | |
| *GTL_RS232 | |
| *LLO_RS232 | |
| Sequence Generator Commands | |
| WAVEFORM | WVFM |
| ADDSEQUENCE | ADDSEQ |
| ADDSEQUENCE? | ADDSEQ? |
| AUTOSEQUENCE | AUTO |
| SEQUENCE | SEQ |
| SEQBURST | SEQB |
| SEQBURSTNUM | SEQBN |

| Command Long Form | Short Form |
|---|---|
| Waveform Editing Commands | |
| WAVEFORM | WVFM |
| AM | |
| DC | |
| EXPONENTIAL | EXP |
| GAUSSIAN | GAUSS |
| HAVERSINE | HSIN |
| LINE | |
| L1NEARSWEEP | LINS |
| PULSE | |
| SAWTOOTH | SAW |
| SCM | |
| SINE | |
| SQUARE | |
| SINE_X_OVR_X | SXX |
| TRIANGLE | |
| LENGTH | LEN |
| LENGTH? | LEN? |
| MAXY | |
| MAXY? | |
| MINY | |
| MINY? | |
| POSITION | POSN |
| POSITION? | POSN? |
| SIZE | |
| SIZE? | |
| SYNC | |
| SYNC? | |
| WAVE | |
| WAVE? | |
| MEMORY | MEM |
| MEMORY? | MEM? |
| MEM_BLOCK? | MBLK? |

## 10.3 RS-232 OVERVIEW

### 10.3.1 Introduction

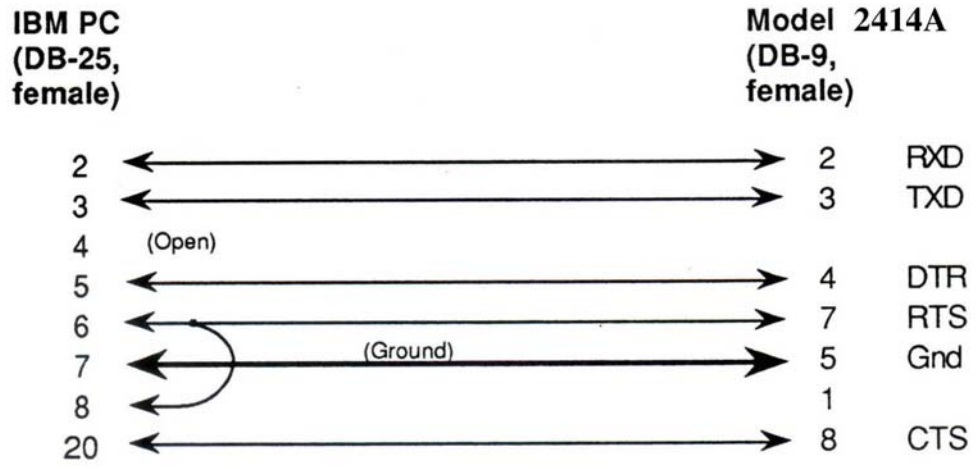
RS-232 is an industry-standard method of sending data back and forth between two pieces of equipment. With the Model 2414A, a computer can remotely control the instrument, download waveform data and upload waveform data. This overview explains the interface requirements, 2414A setup, how to verify communications and the command syntax structure. A sample program is also provided.

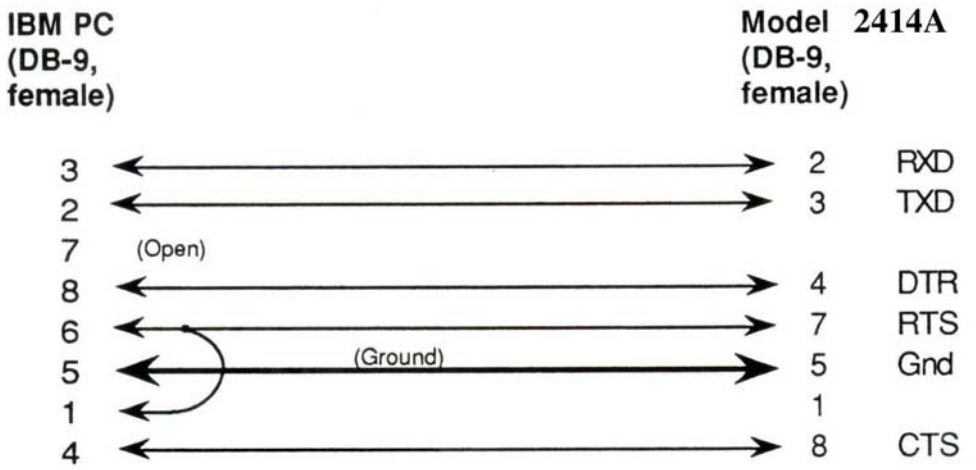
### 10.3.2 Interface Requirements

All IBM (or IBM compatible) personal computers (PCs) should be equipped with at least one serial interface port. It may be either a 9-pin DB-9 or a 25-pin DB-25 connector. An 8-foot 9-pin to 9-pin cable is included with the Model 2414A. If desired, a cable may be constructed per Figure 10-1. Most any software which defines communication protocols may be used. This includes the programming languages Quick Basic, GW Basic, Quick C, Turbo C and Turbo C++. Communications programs such as ProComm, a "shareware" version, are also usually acceptable. A "local echo" feature is helpful to monitor your typing.

## RS-232 ADAPTER CABLES

The following wiring diagrams illustrate proper interconnects between the serial port of an IBM (or compatible) PC and the Model 2414A RS-232 connector.




NOTE:
1) The above cables work for both hardware (HW) and software (SW) handshake. However, binary waveform download is not supported in the software handshake Setting.
2) Special software is required for the PC to support software handshaking.

Figure 10-1. RS-232 Cable Schematic

### 10.3.3 2414A Setup

The following communication protocol parameters are recommended:

| | |
|---|---|
| Baud Rate | 19.2k |
| Parity | None |
| Bits | 8 Data, 1 Stop |
| Handshake[1] | Hardware |

To setup the 2414A:

1) Press UTIL key.
2) Press R232 softkey.
3) Press PC softkey.
4) Press BAUD softkey.
5) Press 19k2 softkey.
6) Press LAST key.
7) Press PAR softkey.
8) Press NONE softkey.
9) Press LAST key.
10) Press BITS softkey.
11) Press 8D1S softkey.
12) Press LAST key.
13) Press HAND softkey.
14) Press HW softkey.
15) Press ENTER key.

[1]With software handshaking, flow control of data to/from the instrument is controlled by the XON/XOFF ASCII characters.
Sending data to 2414A:
Instrument will send a XOFF (ASCII CTRL-S) when the instrument buffer fills to 200 characters.
Instrument will send a XON (ASCII CTRL-Q) when instrument buffer emties to 80 characters.
Receiving data from 2414A:
Instrument will stop sending data when a XOFF (ASCII CTRL-S) is received.
Instrument will resume sending data when a XON (ASCII CTRL-Q) is received.

With hardware handshaking, flow control of data to/from the instrument is controlled by the DTR/CTS lines of the RS-232 interface.
Sending data to 2414A:
Instrument will turn the DTR line off (-12V) when the instrument buffer fills to 200 characters. Instrument will turn the DTR line on (+12V) when the instrument buffer empties to 80 characters.
Receiving data from 2414A:
Instrument will stop sending data when the CTS line is off and resume sending data when it is on.

### 10.3.4 Verify Communication

After the PC and the 2414A have been connected together and programmed for compatible interface parameters, the interface should be tested for proper operation. The following program notation conventions will be observed: The symbol ^ represents the computer Control Key. The use of braces { } around two characters means the two keys must be pressed simultaneously.

To test the interface, type the following:

*IDN?{^J}{^D}

The 2414A should identify itself with the following:

TEGAM, Inc. MODEL 2414A, 0, V1.XX

(XX represents the current firmware revision number.)

### 10.3.5 Command Syntax

The command syntax of the Model 2414A closely relates to the Institute of Electrical and Electronics Engineers (IEEE) Standard 488.2 - 1987.

Commands can be divided into two major categories; common commands and instrument specific commands.

Overviews of the following topics are provided, as they relate specifically to the RS-232C interface:

Common Commands
Event Register and Status and Error Reporting
Functional Syntax Elements

Instrument specific commands, which are identical to both RS-232C and GPIB interfaces, are explained in Section 10.5.

#### 10.3.5.1 Common Commands

Common commands, recognizable by their leading "*" character, are defined by the IEEE 488.2 standard. Common commands dealing with GPIB protocols or status reporting are not supported in the RS-232C interface. Therefore, RS-232C uses the following subset of the mandated GPIB common commands:

| Command | Description |
|---|---|
| ***CLS** | Clear Status |
| *** E S R?** | Standard Event Status Register Query |
| ***IDN?** | Identification Query |
| ***OPC?** | Operation Complete Query |
| ***TRG** | Trigger Command |
| ***TST?** | Self-Test Query |
| ***WAI** | Wait-to-Continue |
| ***OPT?** | System Option Query |
| ***GTR_RS232** | Go To Remote (Brings RS232 into REMOTE state with front panel inactive.) |
| ***GTL_RS232** | Go To Local (Returns RS232 to LOCAL and clears LOCAL LOCKOUT state.) |
| ***LLO_RS232** | Local Lockout[2] |

[2] During REMOTE with LOCAL LOCKOUT, the "LOCAL" key on the front panel is disabled. Remote to Local transitions can only be controlled by the computer. The LOCAL LOCKOUT state is cleared by a "*GTL_RS232" command or on power on.

See Section 10.5 for further descriptions of command formats, operation and expected responses from queries.

### 10.3.5.2 Event Register and Status and Error Reporting

The Standard Event Status Register (ESR) may be utilized to indicate the instrument status. Each of six bits within the eight-bit register indicates a different condition within the 2414A:

ESR - STANDARD EVENT STATUS REGISTER

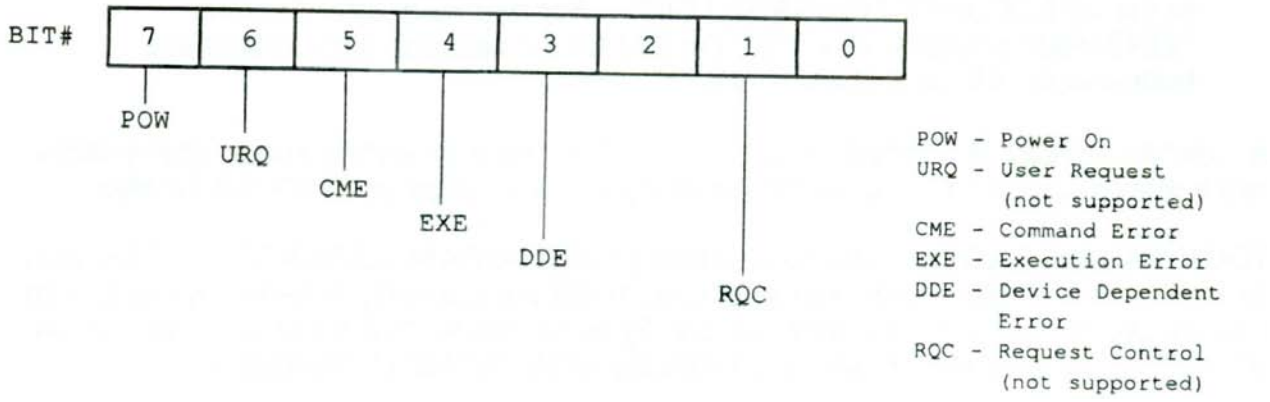



Figure 10-2 ESR Register

### 10.3.5.3 Functional Syntax Elements

In order to establish programming consistency among different manufacturers' devices, IEEE-488.2 has defined a set of rules governing message headers, mnemonics, separators and data types. The following overview will familiarize the programmer with the fundamentals of these rules. As seen below, the rules for command and query messages are much more flexible than their precise response message counterparts.

**<PROGRAM MESSAGE UNIT>** - This is the basic message, and represents an operation to be performed by the device. As an example, if you wanted to create a sinewave with 5 cycles in the currently selected waveform, the appropriate <PROGRAM MESSAGE UNIT> would be:

**WVFM: SINE 5,0**

Notice that a colon (ASCII 3A) is used to separate the <program mnemonio **WVFM** from **SINE.** For query responses the similarly structured <RESPONSE MESSAGE UNIT> is used. A complete list of <PROGRAM MESSAGE UNITS> and <RESPONSE MESSAGE UNITS> for the 2414A, with definitions, mnemonics, and limitations is given in Section 10.5.

General rules:

1. <program mnemonio> shall have maximum length of 12 characters.
2. ":" is used to separate <program mnemonics>, and when preceding a <PROGRAM MESSAGE UNIT>, it indicates that the following <program mnemonio is at the root level.

<PROGRAM MESSAGE UNIT>

1. <program mnemonio can be either upper or lower case alpha.
2. <white space> is only allowed at the end of the header, or next to a"," or a ";". (See note on <white space> below.)

<RESPONSE MESSAGE UNIT>

1. <program mnemonic shall contain upper case alpha only.
2. No <white space> allowed in message header.

To receive a response to a query command, append '^D" (ASCII 4 or CTRL-D) to the <RESPONSE MESSAGE UNIT>. For example, after sending *IDN? <RMT> followed by "^D", the 2414A will respond with "TEGAM Inc., Model 2414A, 0, V1 .XX".

Note: <white space> is defined as a length of 1 or more of <white space characters>.

A <white space character> is a single ASCII byte in the range of 00-09, 0B-20 Hex.

**<PROGRAM MESSAGE> -** This is a series of <PROGRAM MESSAGE UNITS> sent to the device in a single string. For example, to set the currently selected waveform to 1) a sinewave output with a single period, 2) burst mode, 3) a burst of 5, and 4) an amplitude of 2.5V, you would send the following <PROGRAM MESSAGE>:

**WVPM: SINE 1,0;: MODE BURST;:BURST 5;:AMPL 2.5;:EXEC**
{ msg 1 } { msg 2} { msg 3} { msg 4}

The semicolon (ASCII 3B) is required to separate <PROGRAM MESSAGE UNITS> within a <PROGRAM MESSAGE>. For the 2414A a <PROGRAM MESSAGE> can be virtually any length. The structure for a <RESPONSE MESSAGE> for query responses is similar.

**<PROGRAM MESSAGE TERMINATOR>** or **<PMT>** - In order for the device to recognize the end of a <PROGRAM MESSAGE>, a special terminator is required. For command messages, the <PMT> can take one of two different formats:

**CR LF** CR represents carriage return and is an ASCII 0D.
LF represents line feed and is an ASCII 0A.

**LF** LF represents line feed and is an ASCII 0A.

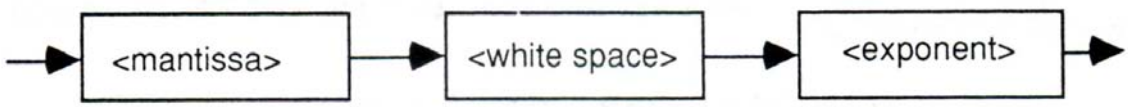
**<RESPONSE MESSAGE TERMINATOR>** or **<RMT>** - For query messages the required terminator is:

**CR LF** (as above)

### 10.3.5.4 Data Formats

Many of the <PROGRAM MESSAGE UNITS> and <RESPONSE MESSAGE UNITS> include numeric data in the message, e.g., the '5' in the message **SINE 5,0.** This section defines acceptable data formats, and Section 10.5 indicates which formats are used with specific commands and responses.

**<DECIMAL NUMERIC PROGRAM DATA>** or **<NRf>** - This is the most flexible of the numeric representations and takes the following general form:




where <mantissa> = [±][**x...x**][**.x...x**] with a maximum length of 255 characters (excluding leading zeros), and the

optional <exponent> = **E**[±][**x...x**] or **e**[±][**x...x**], with a maximum value of ±32000.

Note in the above definitions: x represents digits 0-9
[ ] means enclosed characters are optional

The following example demonstrates several acceptable ways to represent the number 1,234,567,890 in <NRf> format:

**+1234567890** **123456.7890**<white space>**e04**

**1.234567890E+9** **+.1234567890E+10**

Three other numeric data formats are used in <RESPONSE MESSAGE UNITS>, and are subsets of the more general <NRf> format.

**<NR1 NUMERIC RESPONSE DATA>**or**<NR1>**-
[±][**x...x**] e.g. -98765432

**<NR2 NUMERIC RESPONSE DATA>**or**<NR2>**-
[±]**x**[**x...x**].**x**[**x...x**] e.g. +98765.432

**<NR3 NUMERIC RESPONSE DATA>or<NR3> -**
[±]**x**[**x...x**].**x**[**x...x**]**E±x**[**x..x**] e.g. 987.65432E+05

**<NON-DECIMAL NUMERIC PROGRAM DATA>** - Numeric values may also be represented as a binary, octal, or hex number, as follows:

Binary - **#Bx**[**x...x**] or **#bx**[**x...x**] where x is a **0** or **1**

Octal - **#Qx**[**x...x**] or **#qx**[**x...x**] where x is a **0** thru **7**

Hex - **#Hx**[**x...x**] or **#hx**[**x...x**] where x is a **0** thru **F**

**<ARBITRARY BLOCK PROGRAM DATA> -** This data format is used to speed bus transfer in cases where large amounts of data are sent to or from a device (such as waveform or buffer memory data in the 2414A). Only the <DEFINITE LENGTH> block data format is acceptable.

<DEFINITE LENGTH> **- #zy..yx...x** where z is a number 1-9, and represents the number of y digit elements. The y digits taken together as a decimal integer equal the number of 8-bit bytes that follow.

For example to send 4 data bytes<DAB> using the <DEFINITE LENGTH> format, you could send:

**#14<DAB><DAB><DAB><DAB>**

or

**#204<DAB><DAB><DAB><DAB>**

Refer to Section 10.8 for detailed instructions on how to enter data into the waveform memory of the 2414A.

### 10.3.6 Sample Program

The following sample RS-232 program is written in QuickBasic. It reads the instrument identification and writes a 3000-point sine wave into Waveform Number 000:

```
'*************************************************************************
'****      MODEL 2414A : Programming in RS232    (REV 0  2-18-93)       ****
'****                                                                     ****
'****                                                                     ****
'*************************************************************************

'
'   QuickBASIC program using COM1 serial port on IBM PC.
'   Program   a) Reads Instrument Identification  (Query)
'             b) Writes a 3000 point sine wave into WAV#000

'   The 2414A communication settings must be programmed to:
'   BAUD: 9.6K     PAR: NONE     BITS: 8D1S     HAND: HW
CLS                                  ' Clear Screen
CONST Pi = 3.14159
CrLf$ = CHR$(13) + CHR$(10)          ' Command terminator
Talk$ = CHR$(4)                      ' Request query response

' OpenCommLink

  OPEN "COM1: 9600, N, 8, 1, CS5000, DS5000, BIN" FOR RANDOM AS #1

  PRINT #1, CrLf$ + CrLf$;              ' Flush 2414A receive buffer


' Read Identification

   PRINT #1, "*IDN?" + CrLf$;           ' Identification query command

   PRINT #1, Talk$;                      ' Request query response

   LINE INPUT #1, QueryResponse$         ' and read it.

   Flush$ = INPUT$(1, 1)                 ' Flush trailing 'Line Feed'

   PRINT "Instr. ID: "; QueryResponse$   ' which is left in buffer


' Reset 2414A

  PRINT #1, "*Rst" + CrLf$;

' Size WAV#000 to 3000 points and run it
  PRINT #1, ":Wvfm:Wave 0;  Size 0" + CrLf$; 'Erase old WAV#000
  PRINT #1, ":Wvfm:Wave 0  Size 3000" + CrLf$;
  PRINT #1, ":Func Wave,0  :exec" + CrLf$;
  PRINT #1, ":0utsw On; :exec" + CrLf$;


' Draw sine in WAV#000, starting at address 0.

  Slice = 2 * Pi / 3000
  PRINT #1, ":Wvfm:Mem O";                ' Command header, followed by
  FOR Addr = 0 TO 2999                     ' 3000 points of sine wave.
        DataPoint = INT(2047.5 * SIN(Addr * Slice))
        PRINT #1, "," + STR$(DataPoint);   ' Data ',' separated
  NEXT Addr
  PRINT #1, CrLf$;                         ' Terminate command now

CLOSE

'~~~~~~~~~~~~~~~~~~~
```

## 10.4 GPIB (IEEE-488.2) OVERVIEW

### 10.4.1 Introduction

The Model 2414A with the GPIB option, conforms to the Institute of Electrical and Electronics Engineers (IEEE) Standard 488.2-1987. The specific implementation of IEEE-488.1 includes the following functions and subsets:

| Interface Function | Subset |
|---|---|
| Source Handshake | SH1 |
| Acceptor Handshake | AH1 |
| Talker | T6 |
| Listener | L4 |
| Service Request | SR1 |
| Remote Local | RL1 |
| Parallel Poll | PP0 |
| Device Clear | DC1 |
| Device Trigger | DT1 |
| Controller | C0 |
| Electrical Interface | E1 |

To facilitate programming, a brief overview of the IEEE-488.2 Standard (as it specifically applies to the 2414A) is provided.

This section includes: **Common Commands, Status and Event Registers, Functional Elements** (including syntax and nomenclature), **Data Formats, and Error Reporting.** For a more detailed discussion of these topics, a copy of IEEE Standard 488.2-1987 may be obtained from:

*The Institute of Electrical and Electronics Engineers, Inc. 345 East 47th Street, New York, NY10017*

### 10.4.2 Common Commands

Commands can be divided into two major categories: **common commands and instrument specific commands.** Instrument specific commands are detailed in Section 10.5. Common commands are defined by the standard, and, among other things, are used to manage status registers and synchronization. The following is a list of common commands as implemented in the 2414A:

| Command | Description |
|---|---|
| ***CLS** | Clear Status |
| ***ESE** (GPIB only) | Standard Event Status Enable |
| *** ESE?** (GPIB only) | Standard Event Status Enable Query |
| *** ESR?** | Standard Event Status Register Query |
| ***IDN?** | Identification Query |
| ***OPC** (GPIB only) | Operation Complete |
| ***OPC?** | Operation Complete Query |
| ***SRE** (GPIB only) | Service Request Enable |
| ***SRE?** (GPIB only) | Service Request Enable Query |

| Command (Cont.) | | Description (Cont.) |
|---|---|---|
| ***STB?** | (GPIBonly) | Status Byte Query |
| ***TRG** | | Trigger Command |
| ***TST?** | | Self-Test Query |
| ***WAI** | | Wait-to-Continue |
| ***OPT?** | | System Option Query |

See Section 10.5 for further descriptions of command formats, operation, and expected responses from queries.

### 10.4.3 Status and Event Registers

There are four required status or event registers. They are: 1)Standard Event Status Enable (ESE) Register, 2)Standard Event Status (ESR) Register, 3)Service Request Enable (SRE) Register, and 4)Status Byte (STB). These registers indicate device status, and allow the programmer to specify which device events will enable a service request.

ESR and ESE Registers - Each bit of the 8 bit ESR Register indicates a different condition within the device (see Figure 10-2). The ESE Register provides a bit by bit mask of the ESR register. When a bit in the ESE Register is set TRUE, it enables the corresponding ESR bit to generate a Service Request (SRQ), if the ESB bit (bit 5) in the SRE Register has also been enabled. (See discussion on SRE and STB Registers.)

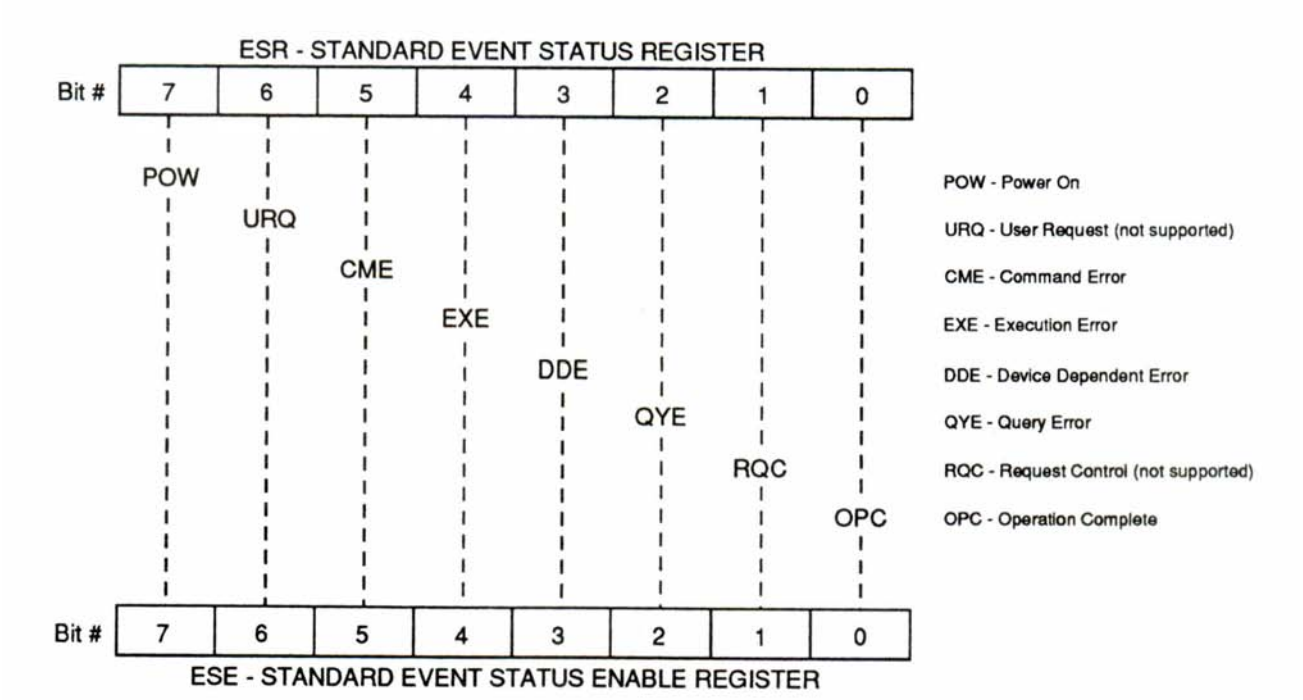
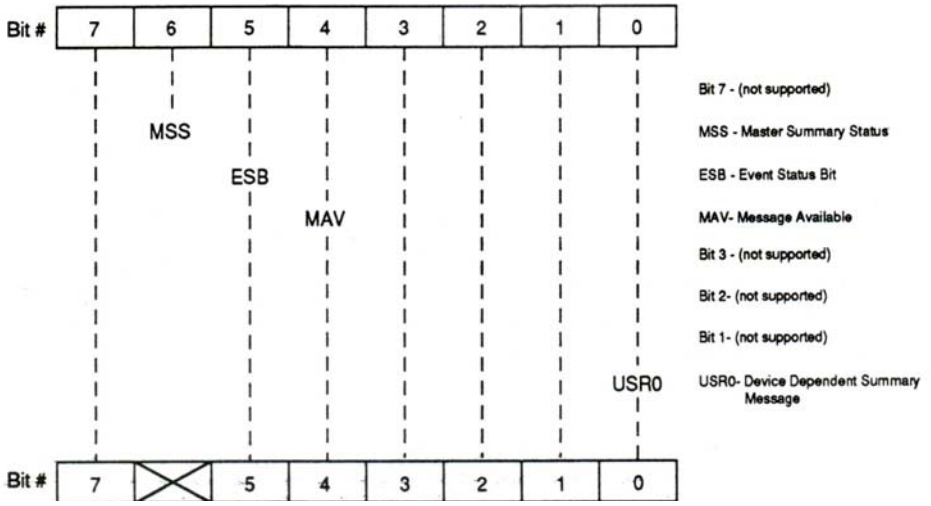



Figure 10-3. ESR and ESE Registers

For example, if you wanted to generate an SRQ on 1) power on (bit 7), or 2) command error (bit 5), or 3) query error (bit 2), you would first set the ESB bit in the SRE Register (see discussion below) and then send the following:

***ESE 164** (164 = decimal equivalent of binary 10100100)
The ***ESE?** and ***ESR?** command queries allow reading of the Standard Event Status Enable and Standard Event Status Registers, respectively. (Ref. Section 10.5.4)
**SRE and STB Registers** - The Status Byte (STB) Register of the 2414A has four active bits (see Figure 10-3) which summarize the current status of the event registers, output queue, and certain instrument specific functions. Similar to the ESE Register, the SRE Register provides an 8-bit mask to allow the programmer to enable each STB bit (with the exception of the MSS bit 6) to generate an SRQ.

STB - STATUS BYTE REGISTER




SRE - SERVICE REQUEST ENABLE REGISTER

Figure 10-4. SRE and STB Registers

As an example, to enable the ESB and MAV bits, but not the USR0, of the Status Byte Register, you would send the following command:

***SRE 48** (48 = Decimal equivalent of B00110000)
The ***SRE?** and ***STB?** command queries allow reading of the Service Request Enable and Status Byte Registers, respectively. (Ref. Section 10.5.4)
The USRO (bit 0) is user-defined, and summarizes the instrument status registers (reference Section 10.5.4, Instrument commands).

**10.4.4 Functional Elements -** Syntax and Nomenclature
In order to establish programming consistency among different manufacturers' devices, IEEE-488.2 has defined a set of rules governing message headers, mnemonics, separators, data types and terminators. The following overview will familiarize the programmer with the fundamentals of these rules. As seen below, the rules for command and query messages are much more flexible than their precise response message counterparts.

**<PROGRAM MESSAGE UNIT> -** This is the basic message, and represents an operation to be performed by the device. As an example, if you wanted to create a sinewave with 5 cycles in the currently selected waveform, the appropriate <PROGRAM MESSAGE UNIT> would be:

**WVFM:SINE 5,0**

Notice that a colon (ASCII 3A) is used to separate the <program mnemonio> **WVFM** from **SINE.** For query responses the similarly structured <RESPONSE MESSAGE UNIT> is used. A complete list of <PROGRAM MESSAGE UNITS> and <RESPONSE MESSAGE UNITS> for the 2414A, with definitions, mnemonics, and limitations is given in Section 10.5.

General rules:

1. <program mnemonio shall have maximum length of 12 characters.
2. ":" is used to separate <program mnemonics>, and when preceding a <PROGRAM MESSAGE UNIT>, it indicates that the following <program mnemonio> is at the root level.

<PROGRAM MESSAGE UNIT>

1. <program mnemonic can be either upper or lower case alpha.
2. <white space> is only allowed at the end of the header, or next to a "," or a ";". (See note on <white space> below.)

<RESPONSE MESSAGE UNIT>

1. <program mnemonio shall contain upper case alpha only.
2. No <white space> allowed in message.

Note: <white space> is defined as a length of 1 or more of <white space characters>. A <white space character> is a single ASCII byte in the range of 00-09, 0B-20 Hex.

**<PROGRAM MESSAGE> -** This is a series of <PROGRAM MESSAGE UNITS> sent to the device in a single string. For example, to set the currently selected waveform to 1) a sinewave output with a single period, 2) burst mode, 3)a burst of 5, and 4) an amplitude of 2.5V, you would send the following <PROGRAM MESSAGE>:

**WVFM:SINE 1,0;:MODE BURST;:BURST 5;:AMPL 2.5;:EXEC**
{ msg 1 } { msg 2 } { msg 3} { msg 4}

The semicolon (ASCII 3B) is required to separate <PROGRAM MESSAGE UNITS> within a <PROGRAM MESSAGE>. For the 2414A a <PROGRAM MESSAGE> can be virtually any length. The structure for a <RESPONSE MESSAGE> for query responses is similar.

**<PROGRAM MESSAGE TERMINATOR>** or **<PMT> -** In order for the device to recognize the end of a <PROGRAM MESSAGE>, a special terminator is required. For command messages, the <PMT> can take one of three different formats:

> ***^END* -** This is defined as sending EO1 TRUE and ATN FALSE with the last byte of the message.
>
> **NL -** NL represents newline and is an ASCII 0A.
>
> **NL^END** - A *"END* sent along with NL as the last byte.

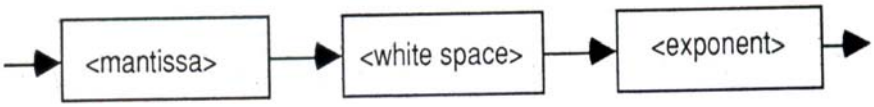
**<RESPONSE MESSAGE TERMINATOR>** or **<RMT> -** For query messages the required terminator is:

> **NL^END** - A ^ND sent along with NL as the last byte.

### 10.4.5 Data Formats

Many of the <PROGRAM MESSAGE UNITS> and <RESPONSE MESSAGE UNITS> include numeric data in the message, e.g., the '5' in the message **SINE 5,0.** This section defines acceptable data formats, and Section 10.5 indicates which formats are used with specific commands and responses.

**<DECIMAL NUMERIC PROGRAM DATA>** or **<NRf>** - This is the most flexible of the numeric representations and takes the following general form:




where <mantissa> = [±][**x...x**][**.x...x**] with a maximum length of 255 characters (excluding leading zeros), and the

optional <exponent> = **E**[±][**x...x**] or **e**[±][**x...x**], with a maximum value of ±32000.

Note in the above definitions: x represents digits 0-9
[ ] means enclosed characters are optional

The following example demonstrates several acceptable ways to represent the number 1,234,567,890 in <NRf> format:

> **+1234567890** **123456.7890**<white space>**e04**
> **1.234567890E+9** **+.1234567890E+10**

Three other numeric data formats are used in <RESPONSE MESSAGE UNITS>, and are subsets of the more general <NRf> format.

**<NR1 NUMERIC RESPONSE DATA>**or**<NR1>-**
[±][**x...x**] e.g. -98765432

**<NR2 NUMERIC RESPONSE DATA>**or**<NR2>-**
[±]**x**[**x...x**]**.x**[**x...x**] e.g. +98765.432

**<NR3 NUMERIC RESPONSE DATA>** or **<NR3> -**
[±]**x**[**x...x**]**.x**[**x...x**]**E**±**x**[**x..x**j e.g. 987.65432E+05

**<NON-DECIMAL NUMERIC PROGRAM DATA>** - Numeric values may also be represented as a binary, octal, or hex number, as follows:

Binary - **#Bx**[**x...x**] or **#bx**[**x...x**] where x is a **0** or **1**
Octal - **#Qx**[**x...x**] or **#qx**[**x...x**] where x is a **0** thru **7**
Hex - **#Hx**[**x...x**] or **#hx**[**x...x**] where x is a **0** thru **F**

**<ARBITRARY BLOCK PROGRAM DATA> -** This data format is used to speed bus transfer in cases where large amounts of data are sent to or from a device (such as waveform or buffer memory data in the 2414A). Both <INDEFINITE LENGTH> and <DEFINITE LENGTH> block data formats are acceptable.

<INDEFINITE LENGTH> - **#0**[**x...x**]<RMT> where x is an 8-bit byte of decimal value 0-255, and <RMT> is the message terminator.

<DEFINITE LENGTH> - **#zy..yx...x** where z is a number 1-9, and represents the number of y digit elements. The y digits taken together as a decimal integer equal the number of 8-bit bytes that follow.

For example to send 4 data bytes<DAB> using the <DEFINITE LENGTH> format, you could send:

**#14<DAB><DAB><DAB><DAB>**
or
**#204<DAB><DAB><DAB><DAB>**

Refer to Section 10.8 for detailed instructions on how to enter data into the waveform memory of the 2414A.

### 10.4.6 Error Reporting

There are four basic types of errors that are reported by a device:

**Command -** In general, when a <PROGRAM MESSAGE> is sent with an error in the syntax, a command error is reported. The command parser (the module that recognizes individual commands) will report the bad command, and look for the next valid command in sequence.

**Execution** - This error represents either program data, which is out of range, or a message, which was not properly executed due to some device condition. In this case the faulty command will generate the error, but not be performed.

**Device Specific** - As the name implies, this error is defined by the specific instrument. Currently there are no Device specific errors generated by the Model 2414A.

**Query** - When a controller or other device attempts to read data from the Output Queue when no data is present or pending, or when output data is lost, a query error is generated.

To clear an error:

1. Correct the condition, which caused the error.
2. Send the ***CLS** command or read the standard event status register by using the ***ESR?** query command.

## 10.5 REMOTE COMMAND SET

### 10.5.1 Introduction

Most of the command set has an equivalent front panel counterpart. Therefore, it is highly recommended that the reader be thoroughly familiar with the front panel operation of the Model 2414A before beginning any programming.

*****IMPORTANT NOTE*****
In order to eliminate some of the common errors encountered while programming instruments, the Model 2414A has adopted a unique approach to sending and executing <PROGRAM MESSAGES>. The following discussions on **command sequence** and **command execution** explain this approach.

**Command Sequence -** Normally a series of commands (or <PROGRAM MESSAGE UNITS>) are sent as a single <PROGRAM MESSAGE> according to the rules outlined in Section 10.4. While these commands are generally executed sequentially, there are certain conditions where the absolute sequential execution of the commands would cause a device error. Take for example the following situation:

Output amplitude is 5Vpp, with offset of 0V

The new test setup calls for an output setting of 1Vpp with +7.5V offset.
(Note: Maximum output amplitude is 10Vpeak)

With most instruments that execute commands sequentially it would be required to first change the amplitude, then change the offset, in order to prevent a device error (caused by the sum of amplitude and offset exceeding 10.4Vpeak; see "ERROR CODES" in the Appendix). Because the Model 2414A is not sequence dependent (except for "WVFM" commands), the command order (within a single <PROGRAM MESSAGE>) is of no consequence. Thus both of the following messages would execute properly without a device error.

**AMPL 1.0;OFFSET 7.5;EXEC**
or
**OFFSET 7.5;AMPL 1.0;EXEC**

**Command Execution -** As already alluded to in the preceding paragraph, in order for the Model 2414A to recognize and execute a command, or series of commands, they usually must be followed by the EXECUTE command. This command is equivalent to the front panel <ENTER> button, and allows the programmer to send a complete <PROGRAM MESSAGE> into an input buffer, before executing any of the individual <PROGRAM MESSAGE UNITS>. The advantage of this method is two fold: 1) it allows the Model 2414A to process the commands very quickly as a group, rather than wait for the slower bus transfers to complete, and 2) it offers the non-sequence dependent benefits as outlined above.

### 10.5.2 Command Set Hierarchy

The command set of the Model 2414A uses a hierarchial structure similar to the file structure on many computer systems. Figure 10-4 shows an example of this structure.

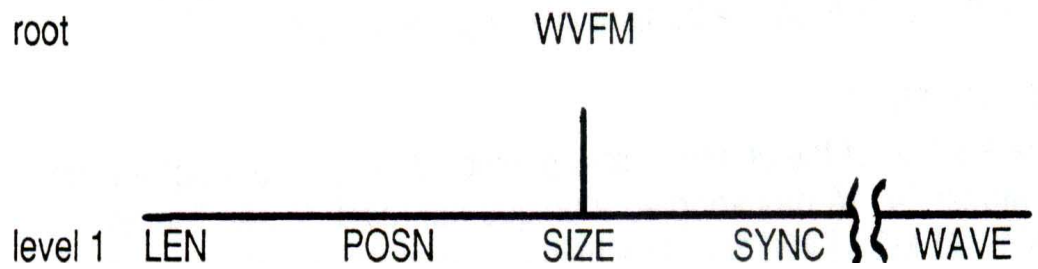



Figure 10-5. Command Hierarchy

While some instruments use several levels within the command set structure, the Model 2414A uses mostly one or two levels. The top level (represented by the mnemonic **WVFM**) is called the root, and the next lower level is level 1. With this structure you must follow a **path** through the root in order to reach the commands on level 1.

Referring to Figure 10-4, if we wanted to execute the command **SIZE,** we would need to indicate the path (through the root) as follows:

**WVFM: SIZE 100;**

Finally, it is important to note that 1) the path rules of the Model 2414A, allow the programmer to delete the root from the command if the level 1 command has the same root as the preceding command, and 2) if the preceding command is at level 1, you must specify any new root by using a":" in front of the root mnemonic . To illustrate 1):

**WVFM: WAVE 10;:WVFM:SIZE 100;**
is the same as
**WVFM:WAVE 10;SIZE 100;**

### 10.5.3 Stacked Queries

In general the Mode! 2414A allows stacked queries, returning the responses in the same order in which the queries were received. The only exceptions to this are the ***IDN?** and ***OPT?** queries. Any queries that are placed after the ***IDN?** or ***OPT?** queries in a <PROGRAM MESSAGE> will be ignored.

### 10.5.4 Command Set

This section gives a complete explanation of all commands, and their structure for the Model 2414A, The following abbreviations are used for convenience:

<ws> - Whitespace as defined in Section 10.4.4
<NR> - This refers to the flexible numeric representation <NRf>, or non-decimal numeric data as defined in Section 10.4.5.
[ ] - Brackets indicate that the enclosed characters or parameters are optional. In the case of the command header either the long form or the short form may be used, but not both.
<arbblk> - This refers to arbitrary block data as described in Section 10.4.5.

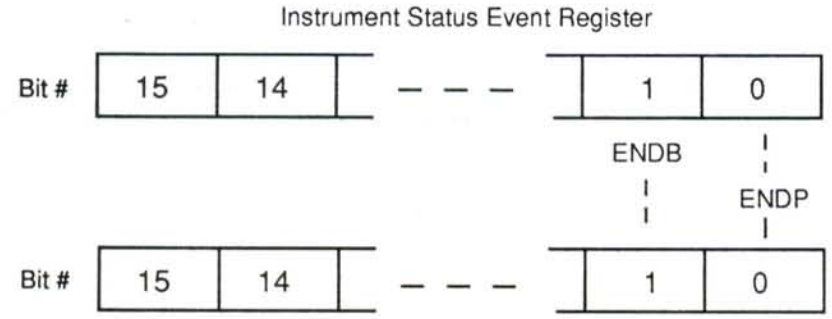
| Root Command [Short Form]<br>▸ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Common Commands*** | | | |
| ***CLS** | | | Clears all event and status registers. Will also clear output queue if it immediately follows a <PMT>. |
| ***ESE**<ws><8-bit mask> (GPIB only) | <NR> | 0/255 | Sets the 8-bit mask in the ESE register. (Ref. Section 10.4.3.) |
| ***ESE?** (GPIB only) | <NR1> | 0/255 | Returns the contents of the ESE register. |
| ***ESR?** | <NR1> | 0/255 | Returns the contents of the ESR register. Once read, the ESR register is cleared. |
| ***IDN?** | | | The specific response to this query is: **Pragmatic Instruments, Model 2414A,0,** <firmware level>, where <firmware level> is of the form **Vx.xx** . |
| ***OPC** (GPIB only) | | | Sets the OPC bit (bit 0) in the ESR register, when all pending instrument operations are complete. |
| ***OPC?** | | | This query waits for all pending instrument operations to complete, then places an ASCII "1" in the output queue. |
| ***RST** | | | Equivalent to the front panel RESET CURRent Key, this returns the instrument to a factory default state. (See Section 10.6.) |
| ***SRE**<ws><8-bit mask> (GPIB only) | <NR> | 0/255 | Sets the 8-bit mask to enable/disable bits in the STB register. (Ref. Section 10.4.3) |
| ***SRE?** (GPIB only) | <NR1> | 0/255 | Returns the value of the SRE register. |
| ***STB?** (GPIB only) | <NR1> | 0/255 | Returns the value of the STATUS BYTE. The Master Summary Status bit (bit 6) is cleared with the first read, but all other bits remain unchanged until the conditions are cleared. (Ref. Section 10.4.3) |
| ***TRG** | | | Equivalent to the front panel TRIG key. |
| ***TST?** | | | Generally this is a selftest command, however it presently does not initiate any action, except to return an ASCII "1". |

| Root Command [Short Form]<br>▸ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Common Commands*** (cont.) | | | |
| ***WAI** | | | This Wait-to-Continue command has no effect, since commands are processed sequentially. |
| **OPT?** | | | No Option Installed: ASCII "0".<br>Option(s) Installed: SEQ, GPIB.<br>(ASCII "0" appears in any one option not installed.) |
| ***Configuration Commands*** | | | |
| **CONFIGURE [CONF]** | | | |
| ▸**HEADERS[HDRS]**<ws><state> | ON/OFF | | With headers ON, query responses include the header. With headers OFF, responses return only the data. |
| ▸**HEADERS?** | ON/OFF | | Returns the current header configuration. |
| ***System Commands*** | | | |
| **EXECUTE [EXEC]** | | | Instructs the instrument to execute pending commands. Equivalent to the front panel ENTER key. (See Section 10.5.1.) |
| **HOLD**<ws><state> | ON/OFF | | Holds or releases the present level of the output voltage. Equivalent to a front panel HOLD. |
| **RECALL [RCLL]**<ws><memory #> | <NR> | 0/30 | Recall front panel setups from specified memory. |
| **REF_CLK_ADJ [RADJ]**<ws><correction #> | <NR> | -2048/ 2047 | Adjusts reference clock by the specified factor. |
| **REF_CLK_ADJ [RADJ]?** | <NR1> | -2048/ 2047 | Returns current ref. clock adjustment factor. |
| **REF_CLOCK[RCLK]**<ws><state> | INT/EXT | | Sets reference clock source to INTernal or EXTernal. |
| **REF_CLOCK [RCLK]?** | INT/EXT | | Returns the present state of the reference clock source. |
| **RESET** | CURR/ ALL | | Resets instrument settings to default values. (See Section 10.6.) |
| **RTNTOSTRT [RTST]** | ON/OFF | | Returns to the starting point of the output waveform when ON. Equivalent to front panel RTS. |
| **SAMPLECLOCK [SCLK]**<ws><frequency> | <NR> | .1/20E6 | Sets the sample clock frequency. |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***System Commands*** (cont.) | | | |
| **SAMPLECLOCK [SCLK]?** | <NR3> | 0.1/20E6 | Returns the present sample clock frequency. |
| **STORE [STOR]**<ws><memory #> | <NR> | 0/30 | Store front panel setups into specified memory. |
| **TGENERATOR[TGEN]**<ws><state> | ON/OFF | | Sets trigger generator ON or OFF. |
| **TGENERATOR [TGEN]?** | ON/OFF | | Returns the present state of the trigger generator. |
| **TGRRATE [TGRR]**<ws><rate> | <NR> | .02/10 | Sets the trigger rate in seconds. |
| **TGRRATE [TGRR]?** | <NR2> | .02/10 | Returns the current trigger rate. |
| **TRIGGER [TRIG]**<ws><state> | ON/OFF/ PULSE | | Sets the trigger ON, OFF, or PULSEd. |

***Instrument Status Commands***

The 16-bit Instrument Status Event Register and Instrument Status Enable Register are laid out as follows. The summarized status is routed to Bit #0 (USR0) in the STB Register.




| Root Command [Short Form]<br>◗ Level 1 Command [Short Form]<br>◗Level 2 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Instrument Status Commands*** | | | |
| **STATUS** | | | |
| **◗INSTRUMENT [INST]** | | | |
| **◗CLEAR [CLR]** | | | Clears Instrument Status Register |
| **◗ENABLE [ENBL]** | <NR> | 0/65535 | Sets the Mask in the Instrument Status Event Register. Bit#0 enables ENDP (End Pulse) and Bit#1 enables ENDB (End Block). |
| **◗ENABLE? [ENBL?]** | <NR1> | 0/65535 | Returns the contents of the Instrument Status Event Enable Register. |
| **◗EVENT? [EVNT?]** | <NR1> | 0/65535 | Returns the contents of the Instrument Status Event Register.<br>BIT #0: ENDP (End Pulse)<br>BIT #1: ENDB (End Block) |

| Root Command [Short Form]<br>♦ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Output Commands*** | | | (Amplitude levels are referenced from 50Ω source impedance into 50Ω load impedance.) |
| **AMPLITUDE [AMPL]**<ws><peak-to-peak> | <NR> | .010/10.2 | Sets the peak-to-peak output voltage. |
| **AMPLITUDE [AMPL]?** | <NR2> | .010/10.2 | Returns the peak-to-peak output voltage. |
| **BURST**<ws><burst #> | <NR> | 1/1048575 | Sets the burst count. |
| **BURST?** | <NR1> | 1/1048575 | Returns the present burst number setting. |
| **CLOCK_SEL [CLKSEL]**<ws><state> | INT/EXT | | Selects either INTernal or EXTernal sample clock source. |
| **CLOCK_SEL [CLKSEL]?** | INT/EXT | | Returns the present setting for the sample clock source. |
| **FILTER**<ws><state> | ON/OFF | | Sets output filter ON or OFF. |
| **FILTER?** | ON/OFF | | Returns the present output filter state. |
| **FREQUENCY [FREQ]** | <NR> | 0/10E6 | Selects the waveform output frequency. |
| **FREQUENCY [FREQ]?** | <NR3> | 0/10E6 | Returns the calculated output frequency. |
| **FUNCTION [FUNC]**<ws><type>, <wav/seq #> | WAVE/ SEQ <NR> or STDW | WAVE: 0/999 SEQ: 0/99 | Selects the specified WAVEform or SEQuence number to send to the output, e.g., FUNC WAVE,1; FUNC WAVE,STDW. |
| **FUNCTION [FUNC]?** | WAVE/ SEQ <NR1> or STDW | | Returns the currently selected WAVEform or SEQuence number in the form **WAVE[SEQ],**<# or **STDW**>. |
| **MODE**<ws><setting> | CONT/ TRIG/ GATE/ BURST/ TOGGLE | | Sets the output signal mode to CONTinuous, TRIGgered, GATEd, BURST, or TOGGLEd. |
| **MODE?** | " " | | Returns present mode of the output signal. |
| **OFFSET [OFST]**<ws><level> | <NR> | -10.2/10.2 | Sets output offset voltage. |
| **OFFSET [OFST]?** | <NR2> | -10.2/10.2 | Returns present offset voltage value. |
| **OUTPUT_SWTCH [OUTSW]**<ws><state> | ON/MUTE or: ON/OFF | | Turns ON, or MUTEs the output. |

| Root Command [Short Form]<br>▸ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Output Commands*** (cont.) | | | |
| **OUTPUT_SWTCH [OUTSW]?** | ON/MUTE | | Returns the state of output switch. |
| **READ_BURST [RBRS]?** | <NR1> | 1/1048575 | Returns the value of the completed burst count |
| **SYNCSEL [SYSEL]**<ws><sync #>,<br><state><br><br>Alternately,<br>SYNC 1: ADDR/ENDP<br>SYNC 2: ADDR/WRUN<br>SYNC 3: ADDR/ENDB | <NR><br>ADDR/<br>STATE | 1,3,4 | Sets the selected SYNC pulse to either an ADDRess, or specific STATE within the waveform. For SYNC1, STATE inserts a pulse at the waveform END Point (or ENDP). For SYNC3, STATE sets the sync pulse high during Waveform RUN (WRUN), and for SYNC4, STATE sets the sync pulse at the END point of each waveform Burst (ENDB) within a sequence. (Ref. SYNC under Waveform Edit Commands.) |
| **SYNCSEL [SYSEL]?**<ws><sync #><br><br>Query Response<br>SYNC 1: ADDR/ENDP<br>SYNC 2: ADDR/WRUN<br>SYNC 3: ADDR/ENDB | <NR> | 1,3,4 | Returns the present state (ADDR or STATE) of the specified SYNC pulse. |
| **TRGINMODE**<ws><state> | SYNC/<br>ASYNC | | Sets the trigger input mode to SYNChronous or ASYNChronous. (See Sec. 8) |
| **TRGINMODE?** | " " | | Returns the present mode of the input trigger. |
| **TRGOUTMODE**<ws><state> | SERIAL/<br>PARALLEL | | For multi-instrument triggering, sets outputs to trigger in serial or parallel. (See Sec8) |
| **TRGOUTMODE?** | " " | | Returns the current output trigger mode. |
| ***Waveform Edit Commands***<br>*(For **Arbitrary** Waves only)*<br>**WAVEFORM [WVFM]** | | | **REFER TO SECTION 10.7 FOR AN OVERVIEW OF WAVEFORM EDITING.** |
| ▸**AM**<ws><# of carrier cycles>,<br><starting carrier phase>,<br><# of modulation cycles>,<br><starting mod. phase>,<br><modulation index> | <NR><br><NR><br><NR><br><NR><br><NR> | 0/1E4<br>-360/360<br>0/1E4<br>-360/360<br>0/200 | Generates a sinewave amplitude-modulated by a sinewave, in the selected waveform memory. (Ref. **WAVE** command.) The first two parameters specify the carrier characteristics, and the second two the modulating waveform. Modulation index can vary from 0 to 200%. |
| ▸**CIRCLE**<ws><# of cycles>,<br><starting phase>,<br><invert> | <NR><br><NR><br>NORM/<br>INVERT | 0/1E4<br>-360/360 | Generates a semicircle in the selected waveform memory. The first parameter specifies the number of cycles, while the second specifies starting phase in degrees. The keyword sets the first part of the waveform NORMal or INVERTed (first half cycle positive or negative, resp.). |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>***(Arbitrary)***<br>**WAVEFORM [WVFM]** | | | |
| ◗ **DC**<ws><Y value> | <NR> | -2048/<br>2047 | Generates a horizontal line at "Y value" in the selected waveform memory. |
| ◗ **EXPONENTIAL**<br>**[EXP]**<ws><exponent>,<br><invert> | <NR><br>NORM/<br>INVERT | 0/20 | Generates a decaying exponential with the specified exponent, $e^{-x}$, in the selected waveform memory. The keyword sets the waveform to a NORMal or INVERTed (positive going or negative going, resp.) decay. |
| ◗ **FM**<ws><# of carrier cycles>,<br><starting carrier phase>,<br><# of modulation cycles>,<br><starting mod. phase>,<br><modulation index> | <NR><br><NR><br><NR><br><NR><br><NR> | 0/1E4<br>-360/360<br>0/1E4<br>-360/360<br>0/1000 | Generates a sinewave frequency-modulated by a sinewave, in the selected waveform memory. The first two parameters specify the carrier characteristics, and the second two the modulating waveform. Modulation index can vary from 0 to 1000. |
| ◗**GAUSSIAN**<br>**[GAUSS]**<ws><exponent>, | <NR> | 0/20 | Generates a gaussian pulse with the specified exponent, $e^{-x^2}$. Where x varies between +/- <exponent>. |
| ◗**HAVERSINE**<br>**[HSIN]**<ws><#cycles> | <NR><br><NR><br><NR> | 0/1E4 | Generates a haversine wave, with the number of cycles specified. The basic shape of this waveform is a sinewave shifted by $-90^{o}$. |
| **WAVEFORM [WVFM]**<br>◗ **LINE**<ws><starting Y value>,<br><ending Y value> | <NR><br><NR> | -2048/<br>2047 | Generates a straight line in the selected waveform memory, with the specified starting and ending Y coordinates. |
| ◗**LINEARSWEEP**<br>**[LINS]**<ws><starting # of cycles>,<br><ending # of cycles>,<br><starting phase> | <NR> | .001/1E4<br>.001/1E4<br>-360/360 | Generates a linearly-swept sinewave, with the number of starting and ending cycles as specified. |
| ◗ **LOGSWEEP**<br>**[LOGS]**<ws><starting # of cycles>,<br><ending # of cycles>,<br><starting phase> | <NR> | .001/1E4<br>.001/1E4<br>-360/360 | Generates a logarithmically-swept sinewave, with the number of starting and ending cycles as specified. |
| ◗ **NOISE** | <NR> | | Generates psuedo-random noise in the selected waveform memory. (Ref. **WAVE** command.) |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>***(Arbitrary)***<br>**WAVEFORM [WVFM]** | | | |
| ◗ **PULSE**<ws><# of pulses>,<br><delay>,<br><risetime>,<br><high time>,<br><falltime>,<br><invert> | <NR><br><NR><br><NR><br><NR><br><NR><br>NORM/<br>INVERT | 1/1E4<br>0/100<br>0/100<br>0/100<br>0/100 | Generates a pulse train, with the number of pulses as specified in the first parameter. Delay, rise/high/falltime are all expressed in percentages of the period of the pulse. |
| ◗ **SAWTOOTH**<br>**[SAW]**<ws><# of cycles>,<br><duty cycle><br><invert> | <NR><br><NR><br>NORM/<br>INVERT | 1/1E4<br>0/100 | Generates a sawtooth waveform with the number of cycles as specified. The keyword sets the sawtooth to a NORMal (rising) or INVERTed (falling) waveform. |
| ◗ **SINE**<ws><# of cycles>,<br><starting phase> | <NR><br><NR> | .001/1E4<br>-360/360 | Generates a sinewave with the number of cycles as specified. |
| ◗**SCM**<ws><# of carrier cycles><br><starting carrier phase><br><# of modulation cycles><br><starting modulation phase> | <NR><br><NR><br><NR><br><NR> | 0/1E4<br>-360/360<br>0/1E4<br>-360/360 | Generates a sinewave amplitude-modulated waveform with suppressed carrier, in the selected waveform memory. (Ref. **WAVE** command.) The first two parameters specify the carrier characteristics, and the second two the modulating waveform. |
| ◗ **SQUARE**<ws><# of cycles>,<br><duty cycle><br><invert> | <NR><br><NR><br>NORM/<br>INVERT | 1/1E4<br>0/100 | Generates a squarewave with the number of cycles as specified. The keyword sets the first half of the squarewave high (NORMal), or low (INVERTed). |
| ◗ **SINE_X_OVR_X**<br>**[SXX]**<ws><# of cycles>,<br><invert> | <NR><br>NORM/<br>INVERT | 4/1E4 | Generates the function sine(x)/x as a waveform, with the number of specified cycles. The keyword sets the waveform NORMal or INVERTed. |
| ◗**TRIANGLE**<ws><# of cycles>,<br><invert> | <NR><br>NORM/<br>INVERT | 1/1E4 | Generates a triangle wave, in the currently selected waveform memory, with the number of specified cycles. The keyword sets the output to initially rise (NORMal), or fall (INVERTed). |
| ◗ **LENGTH [LEN]**<ws><length> | <NR> | 0/131008 | Sets the length (in data points) that any succeeding waveform generation function will create, for the selected waveform. (Ref. to **SIZE** and **POSITION** commands, and Section 10.7.) Functional limits are 0 to SIZE-POSITION, otherwise a device error is generated. |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>***(Arbitrary)***<br>**WAVEFORM [WVFM]** | | | |
| ◗ **LENGTH [LEN]?** | <NR1> | 0/131008 | Returns the current value for LENGTH. |
| ◗ **MAXY**<ws><value> | <NR> | -2048/<br>2047 | Selects the maximum Y value to be produced when generating a waveform. This command is not valid when the standard waveform is selected. |
| ◗ **MAXY?** | <NR1> | -2048/<br>2047 | Returns the currently selected maximum Y value. |
| ◗ **MINY**<ws><value> | <NR> | -2048/<br>2047 | Selects the minimum Y value to be produced when generating a waveform. This command is not valid when the standard waveform is selected. |
| ◗ **MINY?** | <NR1> | -2048/<br>2047 | Returns the currently selected minimum Y value. |
| ◗ **POSITION**<br>**[POSN]**<ws><write position> | <NR> | 0/131007<br>(0/32767<br>for buffer) | Sets the starting position, in the currently selected waveform memory, where new waveform points will be written. The maximum starting position is the size of the memory -1. (Ref. **SIZE** command.) This command is not valid when the standard waveform is selected. After a function such as SINE, etc. is written, **POSITION is automatically incremented** to "POSITION SIZE" to point to the next "new" data point. |
| ◗ **POSITION [POSN]?** | <NR1> | 0/131007 | Returns the current starting position for writing into waveform memory. |
| ◗ **SIZE**<ws><waveform size> | <NR> | 0,<br>32/131008<br><br>(0,<br>32/32768<br>for buffer) | Sets the memory size of the currently selected waveform, in number of points, . The size can be from zero to the total amount of free memory space. If the selected waveform is the standard waveform (STDW), the existing waveform is stretched or squeezed to fit the new size. If the selected waveform is other than the STDW, if enlarging the size, new points (set to 0) are added at the end of the waveform reducing the size. Sending "zero" size will delete waveform. |
| ◗ **SIZE?** | <NR1> | 0/131008 | Returns the present value of SIZE. |

| Root Command [Short Form]<br>♦ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| *Waveform Edit commands* (cont.)<br>*(Arbitrary)*<br>**WAVEFORM [WVFM]** | | | |
| ♦ **SYNC**<ws><sync #>,<br><start position>,<br><length> | <NR><br><NR><br><NR> | 1,3,4<br>0/131007<br>0/131008 | Installs a sync pulse into the specified channel. The start position indicates where in the selected waveform memory the pulse begins, and length specifies the total length of the pulse. Start position can be from 0 to SIZE - 1, while length can range from 0 to SIZE - POSITION. |
| ♦ **SYNC?**<ws><sync #><br><start position>,<br><length> | <NR1><br><NR1><br><NR1> | 1,3,4<br>0/131007<br>0/131008 | Returns the starting position, and length of the specified sync pulse. |
| ♦ **WAVE**<ws><waveform #>,<br><selector> | <NR> or STDW<br><br>WVFM/ BUFR | 0/999 (WVFM), 0/99 (BUFR) | Selects either the Standard Waveform (STDW) or one of the numbered locations within the waveform or buffer memories. POSITION is set to 0, and LENGTH is set to SIZE. (See Sec. 10-7) |
| ♦ **WAVE?** | <NR1> or STDW<br><br>WVFM/ BUFR | 0/999 or 0/99 or STDW | Returns the number of the currently selected waveform memory. |

## Waveform Edit Commands (For Standard [STDV] Waves only)

Following are Waveform Edit Commands for Standard (STDW) Waveforms as opposed to the Arbitrary Waveforms described in the preceding section. General rules for STDW drawing commands are:

1. For STDW, the same parameter limits and function drawing
   rules from the front panel apply.

2. The following commands do NOT apply for STDW:

   POSN, LEN, MINY, MAXY, LINE, MEM

3. The <invert> flag is not allowed except for the following commands:

   EXP, SAW, PULSE

   For example:
   :wvfm:exp 5, **NORM** is used to draw "Exp+" function
   :wvfm:exp 5, **INVERT** is used to draw "Exp-" function

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>***(Standard [STDW])*** (cont.)<br>**WAVEFORM [WVFM]** | | | **REFER TO SECTION 10.7 FOR AN OVERVIEW OF WAVEFORM EDITING.** |
| ◗ **AM**<ws><# of carrier cycles>,<br><starting carrier phase><br><# of modulation cycles>,<br><starting mod. phase><br><modulation index> | <NR><br><NR><br><NR><br><NR><br><NR> | 0/1E4<br>0/360<br>0/1E4<br>0/360<br>0/200 | Generates a sinewave amplitude-modulated by a sinewave, in the selected waveform memory. (Ref. **WAVE** command.) The first two parameters specify the carrier characteristics, and the second two the modulating waveform. Modulation index can vary from 0 to 200%. |
| ◗ **CIRCLE**<ws><# of cycles>,<br><starting phase> | <NR><br><NR> | 0/1E4<br>0/360 | Generates a semicircle in the selected waveform memory. The first parameter specifies the number of cycles, while the second specifies starting phase in degrees. |
| ◗ **DC**<ws><Y value> | <NR> | -2048/<br>2047 | Generates a horizontal line at "Y value" in the selected waveform memory. |
| ◗ **EXPONENTIAL**<br>**[EXP]**<ws><exponent>,<br><invert> | <NR><br>NORM/<br>INVERT | 0/20 | Generates a decaying exponential with the specified exponent, $e^{-x}$, in the selected waveform memory. The vertical range of the waveform is always between 0 and 2047. The keyword sets the waveform to a NORMal or INVERTed (positive going or negative going, respectively) decay. |
| ◗ **FM**<ws><# of carrier cycles>,<br><starting carrier phase><br><# of modulation cycles>,<br><starting mod. phase><br><modulation index> | <NR> | 0/1E4<br>0/360<br>0/1E4<br>0/360<br>0/100 | Generates a sinewave frequency-modulated by a sinewave, in the selected waveform memory. The first two parameters specify the carrier characteristics, and the second two the modulating waveform. Modulation index can vary from 0 to 100. |
| ◗ **GAUSSIAN**<br>**[GAUSS]**<ws><exponent>, | <NR> | 0/20 | Generates a gaussian pulse with the specified exponent, $e^{-x^2}$. Where x varies between +/- <exponent>. |
| ◗ **HAVERSINE**<br>**[HSIN]**<ws><#cycles> | <NR> | 0/1E3 | Generates a haversine wave, with the number of cycles specified. The basic shape of this waveform is a sinewave shifted by $-90^{o}$. |
| ◗ **LINEARSWEEP**<br>**[LINS]**<ws><starting # of cycles>,<br><ending # of cycles>,<br><constant> | <NR><br><NR><br><NR> | 1/1E3<br>1/1E3<br>0/360 | Generates a linearly-swept sinewave, with the number of starting and ending cycles as specified. "Constant" must be sent as a third parameter, but will be ignored. |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>(***Standard [STDW]***) (cont.)<br>**WAVEFORM [WVFM]** | | | |
| ◗ **LOGSWEEP**<br>**[LOGS]**<ws><starting # of cycles>,<br><ending # of cycles>,<br><constant> | <NR><br><NR><br><NR> | .001/1E3<br>.001/1E3<br>0/360 | Generates a logarithmically-swept sine-wave, with the number of starting and ending cycles as specified. "Constant" must be sent as third parameter, but will be ignored. |
| ◗ **NOISE** | | | Generates pseudo-random noise in the selected waveform memory. (Ref. **WAVE** command.) |
| ◗ **PULSE**<ws><# of pulses>,<br><delay>,<br><risetime>,<br><high time>,<br><falltime>,<br><invert> | <NR><br><NR><br><NR><br><NR><br><NR><br>NORM/<br>INVERT | 1/1E3<br>0/100<br>0/100<br>0/100<br>0/100 | Generates a pulse train, with the number of pulses as specified in the first parameter. Delay, rise/high/falltime are all expressed in percentages of the period of the pulse. <Invert> specifies whether the pulses will be inverted or not. |
| ◗ **SAWTOOTH**<br>**[SAW]**<ws><# of cycles>,<br><duty cycle>,<br><invert> | <NR><br><NR><br>NORM/<br>INVERT | 1/1E2<br>0/100 | Generates a sawtooth waveform with the number of cycles as specified. Duty cycle is 50% unless set otherwise. <Invert> sets the sawtooth to a NORMal (rising) or INVERTed (falling) waveform. |
| ◗ **SINE**<ws><# of cycles>,<br><starting phase> | <NR><br><NR> | 0/1E3<br>0/360 | Generates a sinewave with the number of cycles as specified. |
| ◗ **SCM**<ws><# of cycles><br><starting carrier phase><br><# of modulation cycles><br><starting modulation phase> | <NR><br><NR><br><NR><br><NR> | 0/1E4<br>0/360<br>0/1E4<br>0/360 | Generates a sinewave amplitude-modulated waveform with suppressed carrier. (Ref. **WAVE** command.) The first two parameters specify the carrier characteristics, and the second two the modulating waveform. |
| ◗ **SQUARE**<ws><# of cycles>,<br><duty cycle>, | <NR><br><NR> | 1/1E3<br>0/100 | Generates a squarewave with the number of cycles as specified. |
| ◗ **SINE_X_OVR_X**<br>**[SXX]**<ws><# of cycles>, | <NR> | 4/1E3 | Generates the function sine(x)/x as a waveform, with the number of specified cycles. |
| ◗ **TRIANGLE**<ws><# of cycles>, | <NR> | 1/1E3 | Generates a triangle wave, in the currently selected waveform memory, with the number of specified cycles. |
| ◗ **LENGTH [LEN]?** | <NR1> | 0/131040 | Returns the current value for length. |

| Root Command [Short Form]<br>▸ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Edit Commands*** (cont.)<br>***(Standard [STDW])*** (cont.)<br>**WAVEFORM [WVFM]** | | | |
| ▸ **SIZE**<ws><waveform size> | <NR> | 32/1310040<br><br>(0,<br>32/32768<br>for buffer) | Sets the memory size of the currently selected waveform, in number of points. The size can be from 32 to the total amount of free memory space. If the selected waveform is the standard waveform (STDW), the existing waveform is stretched or squeezed to fit the new size. If the selected waveform is other than the STDW, if enlarging the size, new points (set to 0) are added at the end of the waveform. |
| ▸ **SIZE?** | <NR1> | 0,<br>32/131008 | Returns the present value of SIZE. |
| ▸ **SYNC**<ws><sync #>,<br><start position>,<br><length> | <NR><br><NR><br><NR> | 1,3,4,<br>0/131039<br>0/131040 | Installs a sync pulse into the specified channel. The start position indicates where in the selected waveform memory the pulse begins, and length specifies the total length of the pulse. Start position can be from 0 to SIZE - 1, while length can range from 0 to SIZE - POSITION. |
| ▸ **SYNC?**<ws><sync #><br><start position>,<br><length> | <NR1><br><NR1><br><NR1> | 1,3,4<br>0/131039<br>0/131040 | Returns the starting position, and length of the specified sync pulse. |
| ▸ **WAVE**<ws><waveform ?#>,<br><selector> | <NR> or<br>STDW<br><br>WVFM/<br>BUFR | 0/999<br>(WVFM),<br>0/99<br><br>(BUFR)<br>or STDW | Selects either the Standard Waveform (STDW) or one of the numbered locations within the waveform or buffer memories. POSITION is set to 0, and LENGTH is set to SIZE. (See Sec. 10-7) |
| ▸ **WAVE?** | <NR1> or<br>STDW<br><br>WVFM/<br>BUFR | 0/999 or<br>STDW | Returns the number of the currently selected waveform memory. |

| Root Command [Short Form]<br>◗ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Waveform Transfer Commands***<br>**WAVEFORM [WVFM]** | | | |
| ◗ **MEMORY**<br>**[MEM]**<ws><address>,<br><data><br>[,...] | <NR><br><NR><br>or<br><arbblk> | 0/131007<br>(0/32767<br>for buffer)<br>-2048/<br>2047 | **This command applies to all waveforms except STDW.** Sends either individual data points or a block of data into the selected wave- form memory, beginnning at the address specified. The data block may be sent either as individual data points in the <NR> format, or as an arbitrary block of data, high byte first (Ref Section 10.7). |
| ◗ **MEMORY**<br>**[MEM]?**<ws><address> | <NR> | 0/131007<br>for WVFM | Returns a single word of data in the range of -2048 to 2047, beginning at the specified address. (For both ARB and STDW waves). |
| ◗ **MEM_BLOCK**<br>**[MBLK]?**<ws><address>,<br><length> | <NR><br><NR> | 0/131007<br>0/2048 | Returns the number of data points as specified in length (each data point is a 2-byte word, high byte first), beginning at the address as stipulated. The response is made up of first the address in an <NRI> format separated by a comma, then the data in the definite length arbitrary block data format (Ref. Section 10.7) |
| ◗ **COPY**<ws><src NR>, | <NR> | 0/999<br>(WVFM)<br>0/99 | Used to copy entire waveforms between or within the WAVEFORM MEMORY (WVFM) or BUFFER MEMORY (BUFR). |
| <selector>, | WVFM/<br>BUFR | (BUFR) | The first <src NR>,<selector> pair specifies SOURCE waveform number and SOURCE waveform system selector. |
| <dest NR>, | <NR> | 0/999<br>(WVFM)<br>0/99 | Similarly, the second <dest NR>, <selector> pair defines the DESTINATION parameters. |
| <selector> | WVFM/<br>BUFR | (BUFR) | The command "WVFM:COPY 0,BUFR,1,WVFM" copies the contents of BUFR,WAV#0 into WVFM,WAV#1.<br>The COPY command never affects the size of the source or destination waveforms. The transfer process begins by copying the data point at the first location (address 0) of the source waveform to the first location of the target waveform. Then the second data point of the source waveform is copied to the second location (address 1) of the target waveform. This process terminates after the last data point of the source waveform is copied OR the last data point of the destination waveform has been written. The COPY command, using BUFR, can be used to rapidly backup or restore waveforms. Transfer rates up to 400k points/sec provide extremely fast execution speed. |
| ◗ **FREE?**<ws><selector>, | WVFM/<br>BUFR/<br>SEQ | | Returns the number of unused data points in WVFM memory, BUFR memory or the number of unused sequence steps in the SEQuence file system. This command provides support for applications programs to check resource availability before allocating new waveform data or sequence steps. |

| Root Command [Short Form]<br>▸ Level 1 Command [Short Form] | Data Format | Limits Min/Max | Command Description |
|---|---|---|---|
| ***Sequence Generator Commands***<br>(See Section 10.5.5 for Applications) | | | |
| **WAVEFORM [WVFM]** | | | |
| **▸ ADDSEQUENCE** | | | |
| **[ADDSEQ]**<ws><sequence #>,<br><waveform #>,<br><burst count>,<br><sequence step #><br>[,...] | <NR><br><NR><br><NR><br><NR> | 0/99<br>0/999<br>0/1048575<br>0/999 | Adds to the specified sequence a series of waveforms. Waveform parameters come in triplets. The first specifies the number of the waveform, the second, the number of times the waveform is repeated, and the third indicates the sequence step # for that waveform. |
| **▸ ADDSEQUENCE?** | | | |
| **[ADDSEQ]?**<ws><sequence #>,<br><step #> | <NR1><br><NR1> | 0/99<br>0/999 | Returns the waveform #, burst count, and step number (all as integers) of the specified sequence and step number. |
| **▸ AUTOSEQUENCE** | | | |
| **[AUTO]**<ws><1st step #>,<br><step increment> | <NR><br><NR> | 0/999<br>0/999 | Configures the automatic sequence step generator to begin at the selected first sequence step, and increment by the step number to the next sequence step number. This command is used in conjunction with **SEQ, SEQB**. |
| **▸ SEQUENCE** | | | |
| **[SEQ]**<ws><sequence #><br>[,<waveform #> etc.] | <NR><br><NR> | 0/99<br>0/999 | Constructs a sequence with the specified sequence number, composed of a series of waveforms as specified. The burst count is set to 1 for each waveform. *If no waveform number is supplied, the sequence is deleted.* |
| **▸ SEQBURST** | | | |
| **[SEQB]**<ws><sequence #>,<br><waveform #>,<br><burst count><br>[,...] | <NR><br><NR><br><NR> | 0/99<br>0/999<br>0/1048575 | Constructs a sequence with the specified sequence number, composed of a series of waveforms. Waveform parameters come in pairs, with the first being the number of the next waveform in sequence, and the second being the number of times the waveform is repeated. |
| **▸ SEQBURSTNUM** | | | |
| **[SEQBN]**<ws><sequence #>,<br><waveform #>,<br><burst count>,<br><sequence step #><br>[,...] | <NR><br><NR><br><NR><br><NR> | 0/99<br>0/999<br>0/1048575<br>0/999 | Constructs a sequence with the specified sequence number, composed of a series of waveforms. Waveform parameters come in triplets. The first specifies the number of the waveform, the second, the number of times the waveform is repeated, and the third indicates the sequence step # for that waveform. |

### 10.5.5 SEQUENCE GENERATOR APPLICATION NOTES

**ADDSEQUENCE [ADDSEQ]** adds to the specific sequence file, one or more sequence steps. This is the only way to modify an existing sequence file using GPIB or RS-232 once it has been entered. The first parameter is the sequence file to be modified. After the first parameter, the following parameters always come in triplets. The first of these three being the number of the waveform to be sequenced. The second is the number of times this waveform will be repeated, and the third indicates the sequence step # for that waveform.

**Example:**
Assume that sequence file #10 already exists and steps 35 and 53 are NOT used. The following command will add two steps (35 and 53) to sequence #10. Step #35 will be wave #3 repeated 5 times. Step #53 will be wave #2 repeated 4 times.

```
WVFM:ADDSEQ 10,3,5,35,2,4,53
```

**AUTOSEQUENCE [AUTO]** configures the automatic sequence step number generator by setting the sequence step number to start at first parameter and the increment value for the following sequence step numbers. This command is used for the SEQUENCE [SEQ] and SEQBURST [SEQB] commands only.

**Example:**
The following command will start numbering the sequence steps at 10 and increment them by 10, i.e., the first sequence step will be 10, the second sequence step will be 20, etc.

```
WVFM:AUTO 10,10
```

**SEQUENCE [SEQ]** makes a new sequence file, DELETING the existing sequence file if one already exists. The first parameter is the sequence file to create. The following parameters are the waves that are to be sequenced. For the SEQUENCE [SEQ] command, the Burst count (the number of times the wave is repeated) is always set to one. If no waves are supplied, then this command will only delete the specified waveform. The AUTO command is used in conjunction with this command.

**Example:**
The following command will delete sequence file number 23 (if it exists).

```
WVFM:SEQ23
```

The following commands will create a sequence file (#12) that will sequence waves 3, 1, 5, 8 and 3. Step 10 will be wave 3 repeated 1 time. Step 20 will be wave 1 repeated 1 time. Step 30 will be wave 5 repeated 1 time. Step 40 will be wave 8 repeated 1 time. Step 50 will be wave 3 repeated 1 time.

```
AUTO 10,10
WVFM:SEQ 12,3,1,5,8,3
```

**SEQBURST [SEQB]** makes a new sequence file, DELETING the existing sequence file if one already exists. The first parameter is the sequence file to create. The following parameters are pairs, the first parameter of the pair is the wave number, the second parameter is the burst count (the number of times the wave is repeated). The AUTO command is used in conjunction with this command.

**Example:**
The following commands will create sequence file #5. Step 10 will be wave 2 repeated 4 times. Step 20 will be wave 6 repeated 1,000 times. Step 30 will be wave 45 repeated 10,000 times.

```
AUTO 10,10 WVFM:SEQB
5,2,4,6,1000,45,10000
```

**SEQBURSTNUM [SEQBN]** makes a new sequence file, DELETING the existing sequence file if there is one. The first parameter is the sequence file to create. The following parameters are in triplets. First of the three is the wave number, second is the burst count (the number of times the wave is repeated), and the third and last indicates the sequence step number.

**Example:**
The following command will create sequence file #23. Step 5 will be wave 1 repeated 2 times. Step 10 will be wave 3 repeated 4 times. Step 15 will be wave 5 repeated 6 times.

```
WVFM:SEQBN 23,1,2,5,3,4,10,5,6,15
```

### 10.5.6 Programming Example

The following programming example shows how to communicate with the 2414A over the GPIB, using a National Instruments AT-GPIB card, installed in an IBM compatible PC.

The program illustrates in Quick Basic the following:

1) Read 40 words of data, one at a time from the selected memory.
2) Read a block of 10 data words from the selected memory.
3) Create 3 waveforms in waveform memories 0 to 2, then set up these waveforms in a sequence with a single burst for each waveform.

**Quick Basic Programming Example**

```
DECLARE SUB SendArb (ArbString$)
DECLARE SUB WaitDelay (Sec!)
REM $INCLUDE: 'qbdecl.bas'
COMMON SHARED Arb%
CrLf$ = CHR$(13) + CHR$(10)

GOSUB InitGpibResetArb

StartProgram:

    PRINT "Read 40 points using mem command"
    FOR Address = 0 TO 39
        CALL    ibwrt(Arb%, "wvfm:me.m?" + STR$(Ad.dress) + CrLf$)
        GOSUB ReadArb
        PRINT Address; ","; ArbString$,
    NEXT Address

    PRINT "Read a block of 10 using Mblk command"
        CALL     ibwrt(Arb%, "wvfm:Mblk? 0,10" + CrLf$)
        GOSUB ReadArb
        PRINT "Response : >> "; ArbString$; " <<"

    PRINT "Load Sinewave into waveform 0"
    CALL        ibwrt(Arb%, "wvfm:Wave 0;MaxY 2047;MinY -2048;PosN 0;Sine 1,0" +
        CrLf$)
        CALL    ibwrt(Arb%, ":Func Wave,0;:exec" + CrLf$)
        GOSUB WaitForlnput

    PRINT "Load Line into waveform 1"
        CALL    ibwrt(Arb%, "wvfm:Wave 1.MaxY 2047;MinY -2048;PosN   0;Line
        2047, -2048" + CrLf$)
        CALL     ibwrt(Arb%, ":Func Wave,1 ;:exec" + CrLf$)
        GOSUB WaitForlnput

    PRINT "Load LinearSweep into waveform 2"
        CALL     ibwrt(Arb%, "wvfm:Wave 2;MaxY 2047;MinY    -2048;PosN   0;LinS
        1,100,0" + CrLf$)
        CALL    ibwrt(Arb%, ":Func Wave,2;:exec" + CrLf$)
        GOSUB WaitForlnput
```

```
        PRINT "Now sequence waveforms 0 to 2"
            CALL ibwrt(Arb%, "Wvfm:Auto 10,10" + CrLf$)
            CALL ibwrt(Arb%, "WvfnrSeqB 1,0,1,1,1,2,1" + CrLf$)
            CALL ibwrt(Arb%, ":Func Seq,1;:exec" + CrLf$)
            GOSUB WaitForlnput

EndProgram:
STOP
LOOP
END

WaitForlnput:
            CALL ibloc(Arb%)
            INPUT ">>Hit Return to Continue <<"; Scratch$
            RETURN

ReadArb:
        TmpStr$ = SPACE$(100)    'allocate space for gpib response
        WaitDelay (.075)
        CALL ibrd(Arb%, TmpStr$)  'read Arb
         IF ibcnt% > 1 THEN ArbStringS = LEFT$(TmpStr$, ibcnt% - 1) ELSE ArbString$ =
        "TimedOut"
        RETURN

InitGpibResetArb:
        PRINT
        PRINT "Initializing GPIB & resetting ARB"
        PRINT
        CALL IBDEV(0, 16, 0, 12, 1, 0, Arb%)'connect Arb at add 16
        PRINT Arb%
        CALL SendArb("*RST")
        CALL SendArb("*CLS")
        RETURN

SUB SendArb (OutputString$)
        CALL ibwrt(Arb%, OutputString$)
        CALL ibwrt(Arb%, "*ESR?")
        lnputString$ = SPACE$(100)
        CALL ibrd(Arb%, lnputString$)
         IF VAL(lnputString$) = 4 THEN PRINT "»QUERY ERROR«": CALL ibwrt(Arb%,
        "*CLS")
         IF VAL(lnputString$) > 0 THEN PRINT ">>ESR ERROR<< read :"; VAL(lnputString$):
        CALL ibwrt(Arb%, "*CLS")
        END SUB

SUB WaitDelay (Sec)
 StrtTime = TIMER
 CurrentTime = TIMER
 StpTime = StrtTime + (Sec)
 DO WHILE CurrentTime < StpTime
 CurrentTime = TIMER
LOOP
‘        PRINT StrtTime,
CurrentTime END SUB
```

## 10.6 RESET AND FACTORY DEFAULTS

The following Table shows the parameter values, which are affected for both factory default, and reset setting conditions. Factory default can be accessed via the front panel RESET key, and selecting the ALL function, or over the GPIB with the **RESET-ALL** command. Reset settings can be accessed either from the front panel RESET, CURRent function or over the GPIB with the ***RST** or **RESET-CURR** commands.

| Parameter | Default Value | Factory Set | *RST |
|---|---|---|---|
| Reference Clock Adjust | 0 | √ | √ |
| Reference Clock Select | INTernal | √ | √ |
| Sample Clock Freq. | 10MHz | √ | √ |
| Sample Clock Select | INTernal | √ | √ |
| Trigger Generator Rate | 50 ms | √ | √ |
| Trigger Generator State | OFF | √ | √ |
| Amplitude | 5.000 V | √ | √ |
| Burst Count | 3 | √ | √ |
| Filter | OFF | √ | √ |
| Function | STDW Wave | √ | √ |
| Mode | CONTinuous | √ | √ |
| Offset | 0 V | √ | √ |
| Output Switch | MUTE | √ | √ |
| Trigger Input Mode | ASYNChronous | √ | √ |
| Trigger Output Mode | SERIAL | √ | √ |
| Waveform (Edit) Functions: | | | |
| AM <carrier cycles> | 20 | √ | √ |
| <phase> | 0 | √ | √ |
| <mod. cycles> | 1 | √ | √ |
| <phase> | 0 | √ | √ |
| <mod. index> | 100% | √ | √ |
| Circle | 1 cycle, 0 phase | √ | √ |
| DC | 0 | √ | √ |
| Exponential | +5 | √ | √ |
| FM <carrier cycles> | 20 | √ | √ |
| <phase> | 0 | √ | √ |
| <mod. cycles> | 1 | √ | √ |
| <phase> | 0 | √ | √ |
| <mod. index> | 10 | √ | √ |
| Gaussian Exponent | 2.00 | √ | √ |
| HSIN | 1.00 | √ | √ |
| Linear Sweep | | | |
| <start cycles> | 1 | √ | √ |
| <end cycles> | 10 | √ | √ |
| Log Sweep | | | |
| <start cycles> | 1 | √ | √ |
| <end cycles> | 10 | √ | √ |

| Parameter | Default Value | Factory Set | *RST |
|---|---|---|---|
| Pulse <# of pulses> | 1 | √ | √ |
| <delay> | 0% | √ | √ |
| <risetime> | 10% | √ | √ |
| <hightime> | 30% | √ | √ |
| <falltime> | 10% | √ | √ |
| SCM <carrier cycles> | 20 | √ | √ |
| <phase> | 0 | √ | √ |
| <mod. cycles> | 1 | √ | √ |
| <phase> | 0 | √ | √ |
| Sawtooth <cycles> | 1 | √ | √ |
| <duty cycle> | 100%: SAW+ | √ | √ |
| | 0: SAW- | √ | √ |
| Sine | 1 cycle, 0 phase | √ | √ |
| Sine X Over X | 1 cycle | √ | √ |
| Square <cycles> | 1 | √ | √ |
| <duty cycle> | 50% | √ | √ |
| Triangle | 1 cycle | √ | √ |
| Sync Sel (1,3 & 4) | ADDRess | √ | √ |
| Waveform Memory: | | | |
| WAVE | 000,001,002,003,004 | | |
| Size | 2000 points | √ | Unchanged |
| Data | Affected | √ | Unchanged |
| SYNC start/length | 0/1 | √ | Unchanged |
| STDW | | | |
| Size | 1000 points | √ | Unchanged |
| Data | 1 cycle sinewave | √ | Unchanged |
| SYNC start/length | 0/1 | √ | Unchanged |
| Sequence Generator: | | | |
| SEQ | 000,001,002 | √ | Unchanged |
| Buffer Memory: | | | |
| BUFR | 00,01,02,03,04 | | |
| Size | 2000 points | √ | Unchanged |
| Data | Affected | √ | Unchanged |

## 10.7 WAVEFORM EDITING PRINCIPLES

In order to successfully apply the waveform editing commands of the Model 2414A, it is important to understand the principles behind memory allocation, size and the active memory area. For the following discussion refer to Figure 10-4.

There are 131,040 words of active waveform memory and 32,768 words of buffer memory. The active memory is divided up into an area designated for the Standard Waveform (STDW), and the rest of the memory, which can be partioned into waveform files numbered 0 to 999. The buffer memory can be partioned into waveform files numbered 0 to 99. The horizontal size (in digital words) of each partioned waveform can be set by the **SIZE** command, and thus the total number of waveforms is limited by the cumulative size of the individual waveforms.

The minimum and maximum y values for the two memories are -2048 and +2047, respectively.

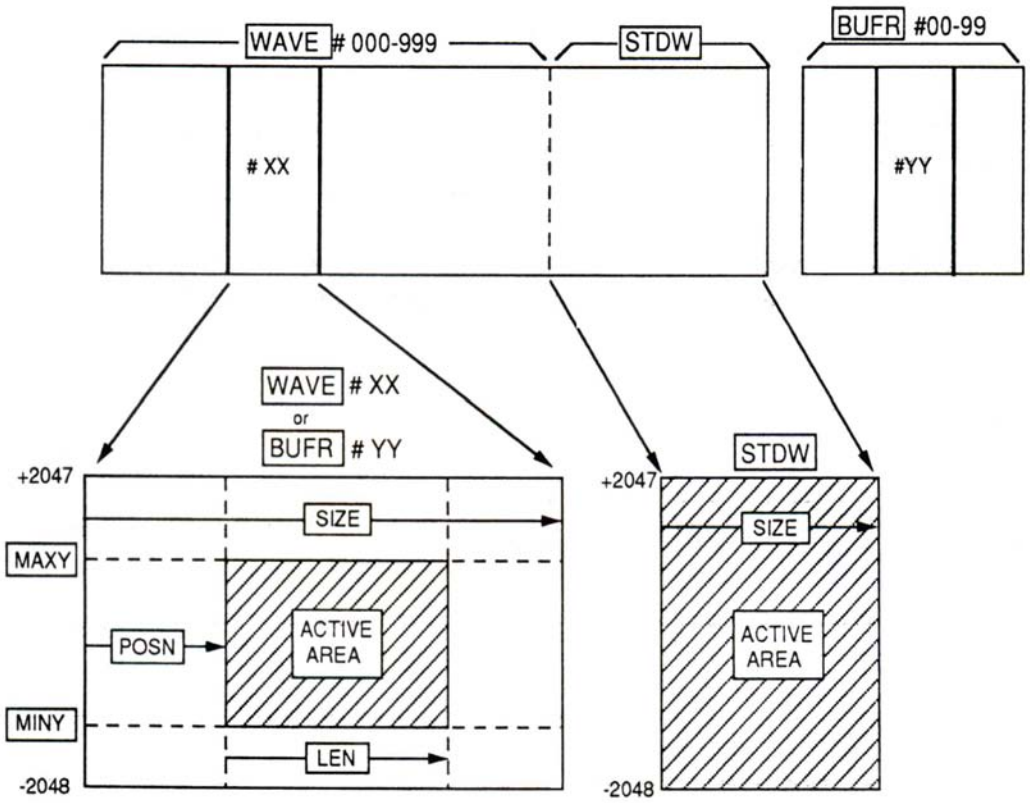



Figure 10-6 Waveform Editing Commands

When using the Waveform editing commands to operate on a specific waveform memory, it is important to realize that these commands only affect the active area (Set Note below). In other words, if you wanted to create a sinewave with three cycles starting at 0 phase, you would send the following command:

**WVFM:SINE 3,0;:EXEC**

This command would place three cycles of sinewaves into the active area, with a maximum and minimum y value the same as the active area. The commands **LEN, MINY, MAXY,** and **POSN** all affect the actual dimensions of the active area as follows:

**LEN** - The length command determines the actual horizontal length of the active area. This maximum length is SIZE-POSN.
**MINY** - The minimum y command sets the lowest y value for the active area.
**MAXY** - The maximum y command sets the highest y value for the active area.
**POSN** - The position command determines the horizontal starting point for the active area. Note that once a waveform is written into the active area (using the waveform edit commands), POSN is changed to the point POSN + LEN+1.

**Important Note:** For the Standard Waveform (STDW), the active area is always the entire area defined by SIZE. Thus, none of the four commands mentioned above effect the STDW.

## 10.8 WAVEFORM MEMORY FORMATS

This section describes the formats of the commands, which enter data into the wave memory of a TEGAM, Inc. arbitrary waveform generator via the GPIB or RS-232 interface.

### 10.8.1 Decimal Waveform Download

The contents of the waveform memory for wave X (X = 0..999) are changed by a single command formatted according to the IEEE 488.2-1987 standard, and has the following syntax:

<command header> <start address>, <data>[,<data> ... ];

As a command for the GPIB or RS-232, a <command header>, followed by the address of the first memory cell to be set, followed by one or more <data> items and terminated by a semicolon. The <start address> and <data> items are separated by commas, with a space or tab separating the <command header> from the <start address> item. The numbers specified by the <data> items are stored in successive addresses of the wave memory.

The <command header> is a string, whose value depends only on the wave to be set:

| Wave | Command Header Value |
|---|---|
| 0 | WVFM:WAVE 0;MEM |
| 1 | WVFM:WAVE 1;MEM |
| 999 | WVFM:WAVE 999;MEM |

The <start address> is a decimal number between 0 and 131007. It sets the starting address where the <data> will be downloaded in waveform memory.

The <data> items are decimal numbers between -2048 and 2047. If the AMPL (Amplitude) parameter is set to 10V, the following data: -2048,0,2047 would produce -10 Volts, 0 Volts, and 10 Volts, respectively, on the output.

Examples:

An 8 point positive Ramp down loaded into wave 1, address 0.

WVFM:WAVE 1;MEM 0,0,292,584,877,1169,1462,1754,2047;

An 8 point sine wave down loaded into wave 2, address 48.

WVFM:WAVE 2;MEM 48,0,1447,2047,1447,0,-1448,-2048,-1448;

### 10.8.2 Binary Waveform Download

The 2414 A also supports binary format for waveform <data> items. Binary format is the fastest way to transfer data.

<command header> <start address>,<binary data><\n>

The <command header> and <start address> are the same as above. The <binary data> has the following format:

<#><num><length><hi byte><lo byte> [<hi><lo>. . . ]<\n>

Where:
<#> is the pound sign, ASCII(35).
<num> is the number of digits in the length param. 1 to 9.
<length> is the length in bytes (two bytes per *waveform data point). 2* to 262016.
<hi byte> contains the upper four bits of a *waveform data point.*
<lo byte> contains the lower eight bits of a *waveform data point.*
<\n> is the new-line character, ASCII(10).

NOTE:
1. The range of a *waveform data point* is 0 to 4095. If the AMPL (Amplitude) parameter is set to 10V, the following *waveform data point* values: 0, 2048, and 4095, produce -10 Volts, 0 Volts, and 10 Volts, respectively, on the output. The *waveform data point* is converted to <hi byte><lo byte> format for binary downloading. Conversion from *waveform data point* to <hi byte><lo byte> can be accomplished as follows:

   QuickBasic

   ```
   high_byte = FIX( pointdata / 256)
   low_byte = pointdata MOD 256
   ```

   C

   ```
   high_byte = pointdata / 256
   low_byte = pointdata % 256
   ```

   Where pointdata is between 0 and 4095.

2. Since two bytes are required for each *waveform data point,* the length of bytes sent must always be even.

Example:

An 8 point positive Ramp down loaded into wave 1, address 0.

WVFM:WAVE 1;MEM 0,#40016**binary_data**\n

Where ***blnary_data*** are the following values as bytes sent to the GPIB or RS-232.

| 8| 0| 9|36|10|72|ll|109|12|145ll3|182|14|218|15|255|

The complete command as bytes in memory would look like the following:

|'W' I'V'I'F'I'M'I':'I'W'I'A'I'V'I'E'I' 'I'1'I':'I'M'|
|'E'|'M'I' '|'0'I','I'#'I'4'I'0'I'0'I'1'I'6'I 8 I 0 I
| 9 | 36| 10| 72| 11|109| 12|145| 13|182| 14|218|
I 15|255| 10|

Where | 'A' | represents one byte in memory containing the character 'A' (i.e. the value ASCII(65)) and I 218 | represents one byte in memory whose value is 218. Therefore: | 'A' | and | 65 | are equal in value.

In QuickBASIC

```
A$= "WVFM:WAVE 1;MEM 0,#40016"+CHR$(8)+CHR$(0)
      +CHR$(9)+CHR$(36)+CHR$(10)+CHR$(72)+CHR$(11)+CHR$(109)
       +CHR$(12)+CHR$(145)+CHR$(13)+CHR$(182)+CHR$(14)+CHR$(218)
       +CHR$(15)+CHR$ (255)+CHR$ (10)
```

In C

```
{
int i;
char wvfm_buffer[100];
char data[16]={8,0,9,36,10,72,11,109,12,145,13,182,14,218,15,
255}
sprintf(wvfm_buffer, "WVFM:WAVE 1;MEM 0,#40016");
for (i=0; i < 16; i++)
    wvfm_buffer[i+24] = data[i]; /* 24 bytes in header */
     wvfm_buffer[16+24] = '\n'; /* end of block */
 }
```

Example:

An 8 point sine wave down loaded into wave 2, address 48.

WVFM:WAVE 2:MEM 48, #40016**binary_data**\n

Where ***binary_data*** *are* the following values as bytes sent to the GPIB or RS-232.

I 8| 0|13|167|15|255|13|167| 8| 0| 2|88| 0| 0| 2|88|

The complete command as bytes in memory would look like the following:

I 'W' I 'V' | 'F' I 'M' I ':' I 'W' I 'A' I 'V' I 'E' I ' ' | '2' | ':' | 'M' |
I 'E' I 'M' I ' ' I ' ' I '4' I '8' I ','I'#'I '4' I '0' I '0' I '1' I '6' I
| 8 | 0 | 13 | 167 | 15 | 255 | 13 | 167 | 8 I 0 | 2 | 88 |
I 0 | 0 I 2 | 881 10 |

**In BASIC**

```
A$= "WVFM:WAVE 2;MEM 48, #40016"+CHR$ (8)+CHR$(0)
      +CHR$ (13)+CHR$(167)+CHR$ (15)+CHR$(255)
      +CHR$(13)+CHR$(167)+CHR$(8)+CHR$(0)+CHR$(2)+CHR$(88)
      +CHR$(0)+CHR$(0)+CHR$ (2)+CHR$(88)+CHR$(10)
```

**In C**

```
{
int i;
 char wvfm__buffer[100];
 char data[16]= {8,0,13,167,15,255,13,167,8,0,2,88,0,0,2,88}
 sprintf(wvfm_buffer,"WVFM:WAVE 2;MEM 48,#40016") ;
 for(i=0; i < 16; i++)
    wvfm_buffer[i+25] = data[i]/ /* 25 bytes in header */
 wvfm_buffer[16+25] = '\n'; /* end of block: */
 }
```

```
REM ************************************************************
REM *****       2414A BINARY DOWNLOAD TEST PROGRAM FOR GPIB  ****
REM ************************************************************'*******
REM  Language   : Microsoft Quick Basic
REM  Computer   : IBM AT or better
REM  GPIB Board : National Instruments AT-GPIB, IEEE488.2
REM
REM Function   : The program downloads a 8400 point Sine Wave
REM               into Wave 1, starting at Address 0.
REM               The data transfer occurs in 5 blocks, the first
REM               four blocks containing 2048 Data Points and the
REM               last containing 208 Data Points.
REM ************************************************************
REM $INCLUDE: 'qbdecl.bas'

COMMON SHARED Arb%

    PRINT
    PRINT "Initializing GPIB & resetting ARB"
    PRINT
     CALL IBDEV(0, 16, 0, 12, 1, 0, Arb%)'connect Arb at add 16
    CALL IBWRT(Arb%, "*idn?")
    ArbResponse$ = SPACE$(100)
    CALL IBRD(Arb%, ArbResponse$)
    IF ibcnt% > 0 THEN
         PRINT "Unit is a : "; MID$(ArbResponse$, 1, ibcnt%)
         GOTO GpibWasOk
        END IF
    PRINT "» GPIB ERROR «"
     PRINT "Corrective Action REQUIRED"
     STOP

GPIBWasOk:
SetupParameters:

MaxBlockSize& = 2048
Lengths = 8400

DIM ArbDatas (Lengths)


    NumBlocks& = INT(Length& / MaxBlockSize&)
    NumPartial& = Length& - NumBlocks& * MaxBlockSize&

    PRINT "Calculating a sample Sinewave"

    Pie = 3.141593

    FOR x& = 0 TO Length&
         ArbData& (x&) = 2047 * SIN((x& / Length& * 10) * 2*Pie) +
         2047
         IF ArbDataS(x&) > 4097 OR ArbDataS(x&) < 0 THEN STOP
    NEXT x&
```

```
    PRINT "Setting up machine"

    CALL IBWRT(Arb%, "reset all;:exec")
    CALL IBWRT(Arb%, "*cls")        'Start clean
    CALL IBWRT(Arb%, "outsw on;:exec")
     CALL IBWRT(Arb%, ":wvfm:wave 1;size " + STR$ (Length&) + ";len "
 + STR$(Length&))
    CALL IBWRT(Arb%, ":wvfm:wave l;posn 0;miny –2047;maxy 2047")
     CALL IBWRT(Arb%, ":Func Wave,1;:exec")
     GOSUB CheckStatus


    BlockNum& = 0

    PRINT "Starting Binary Download of : "; Length&; " points"
    PRINT "Start > "; TIME$

    DO WHILE (BlockNum& < NumBlocks&)
         BlockSize& = MaxBlockSize&
         GOSUB MakeHeader
         StartIndex& = (BlockNum& * MaxBlockSize&)
         BlockNum& = BlockNum& + 1
         StopIndex& = (BlockNum& * MaxBlockSize&) - 1
         PRINT "Assembling Block : " ,• BlockNum&; " in progress";
         GOSUB AssembleData
         PRINT " > DownLoading < ";
         CALL IBWRT(Arb%, "disp 'Block "+STR$(BlockNum&)+
" processing'" +CHR$(10))
         CALL IBWRT(Arb%, "wvfm:mem " + STR$(StartIndex&) + "," +
Header$ + BinData$ + CHR$ (10))
         PRINT " >> completed <<"
         GOSUB CheckStatus
LOOP
    DO WHILE (NumPartial&)
         StartIndex& = (BlockNum& * MaxBlockSize&)
         BlockSize& = NumPartial&
         GOSUB MakeHeader
         PRINT "Assembling a partial bock of : "; BlockSize&;
         CALL IBWRT(Arb%, "disp 'Partial Block " + STR$(BlockNum&)+
"'" + CHR$ (10))
         StopIndex& = NumPartial& - 1 + StartIndex&
         GOSUB AssembleData
         PRINT " > DownLoading > ";
         CALL IBWRT(Arb%, "wvfm:mem " + STR$(StartIndex&) + "," +
Header$ + BinData$ + CHR$(10))
         PRINT " >> completed <<"
         GOSUB CheckStatus
         NumPartial& = 0
     LOOP

     PRINT " Stop > "; TIME$
     CALL ibloc(Arb%)
     INPUT " » Hit Return to continue «", Scratch$

     END
```

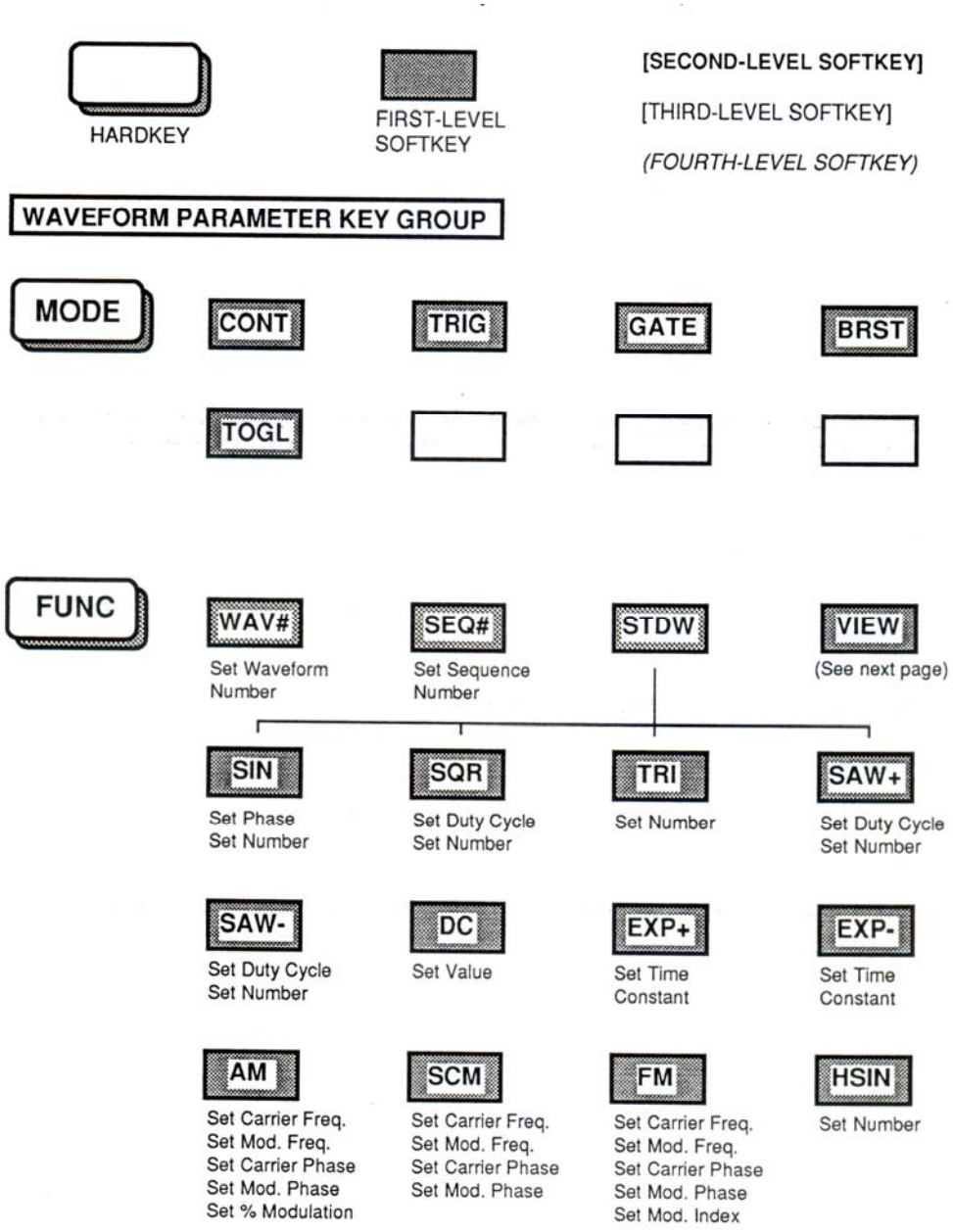
```
MakeHeader:

    templ$ = "#6000000"
    temp2$ = LTRIM$ (STR$(2 * BlockSize&))
    Header$ = LEFT$(templ$, LEN(templ$) - LEN(temp2$)) + temp2$
    RETURN

AssembleData:

    CurrentPosition& = 1
    BinData$ = SPACE$(2 * (StopIndexS - StartIndexS +1))
     FOR IndexPointer& = StartIndex& TO StopIndex&
     MID$(BinData$, CurrentPosition&, 2) = CHR$(INT(ArbData&(
IndexPointer&) / 256)) + CHR$(ArbData&(IndexPointer&) AND 255)
     CurrentPosition& = CurrentPosition& + 2
     NEXT IndexPointer&
     RETURN

CheckStatus:

     CALL IBWRT(Arb%, "*ESR?")
     InputString$ = SPACE$(100)
     CALL IBRD(Arb%, InputString$)
     IF VAL(InputString$) AND 1 THEN
       PRINT “>> Operation Complete <<”;
       END IF
     IF VAL(InputString$) AND 2 THEN
        PRINT “>> Request Control ( N/I ) <<”;
        ENDIF
     IF VAL(InputString$) AND 4 THEN
         PRINT "» QUERY ERROR «";
         ENDIF
     IF VAL(InputString$) AND 8 THEN
         PRINT “>> Device Dependant ERROR <<”;
         ENDIF
     IF VAL(InputString$) AND 16 THEN
         PRINT “>> Execution ERROR <<”;
         ENDIF
     IF VAL(InputString$) AND 32 THEN
         PRINT “>> Command ERROR <<”;
         ENDIF
     IF VAL(InputString$) AND 64 THEN
         PRINT “>> User Request ( N/I ) <<”;
         ENDIF
     IF VAL (InputString$) AND 128
         THEN PRINT “>> Power On “<<;
         ENDIF
     IF VAL(InputString$) > 0 THEN
         PRINT " Status read : "; VAL(InputString$)
         ENDIF
     IF VAL(InputString$) > 0 THEN CALL IBWRT(Arb%, "*CLS"): STOP
     RETURN
```

# APPENDIX

## MENU LOGIC TREE

This Menu Tree provides a complete list of all setup and control display menus.

Pressing a hard key opens a menu with additional choices. The submenus are shown horizontally to the right and below the main menu selections. The second branch submenus are shown vertically below the submenu selections. Keys are identified as shown below: Many entries provide softkey selections of DO, OK and CANCEL.



## WAVEFORM PARAMETER KEY GROUP (Cont.)

**SLIN**
Set Beginning and End Frequencies

**SLOG**
Set Beginning and End Frequencies

**PLS+**
Set Delay %
Set Rise %
Set High %
Set Fall %

**PLS-**
Set Delay %
Set Rise %
Set High %
Set Fall %

**GAUS**
Set Exponent Power

**SX/X**
Set Number

**CIRC**
Set Phase
Set Number

**NOIS**

**VIEW**

**[ALL]**
View entire active memory contents.

**[SEG1]**
Set left address
Set right address

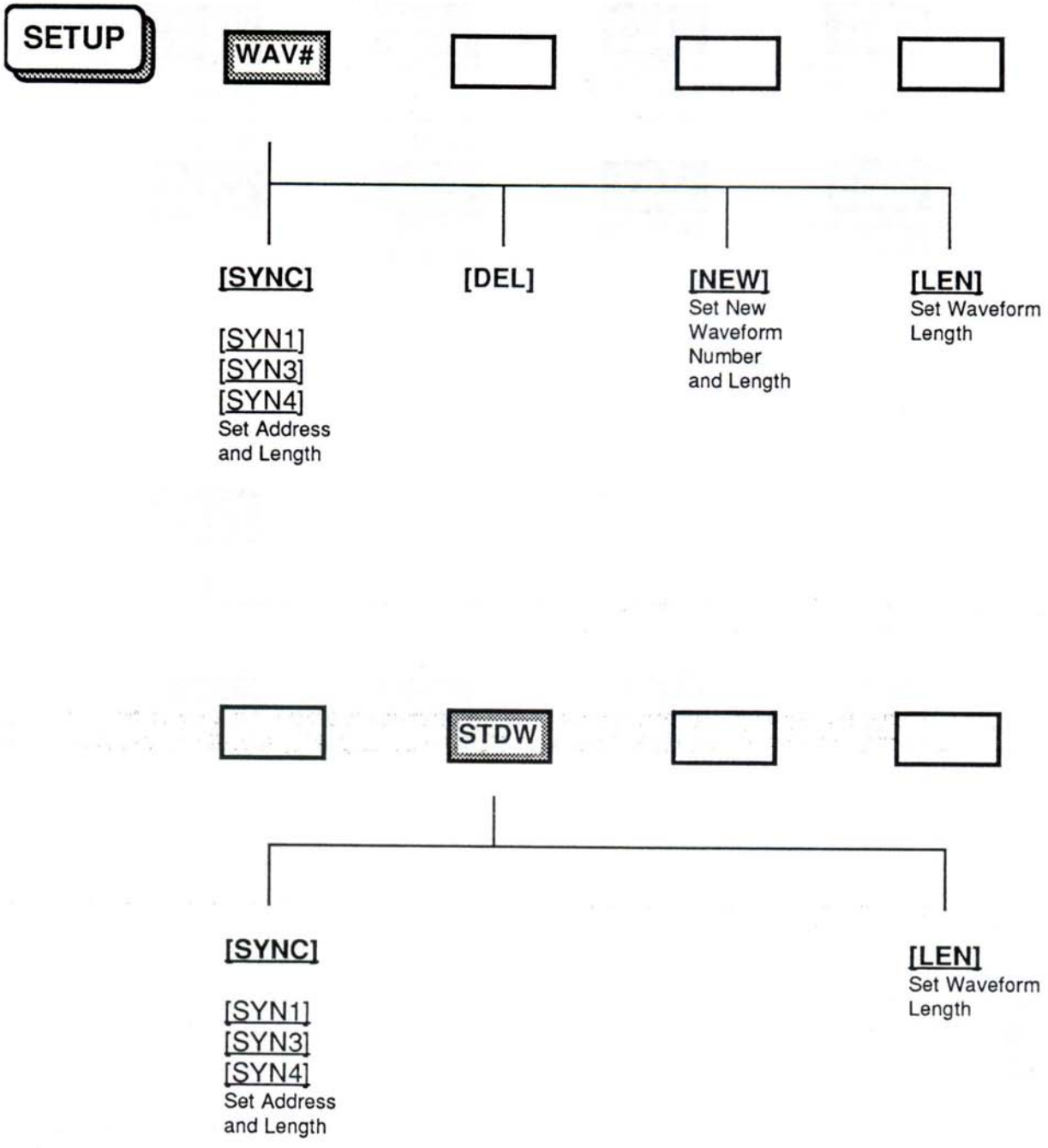
**[SEG2]**
Set left address
Set right address

**[SEG3]**
Set left address
Set right address

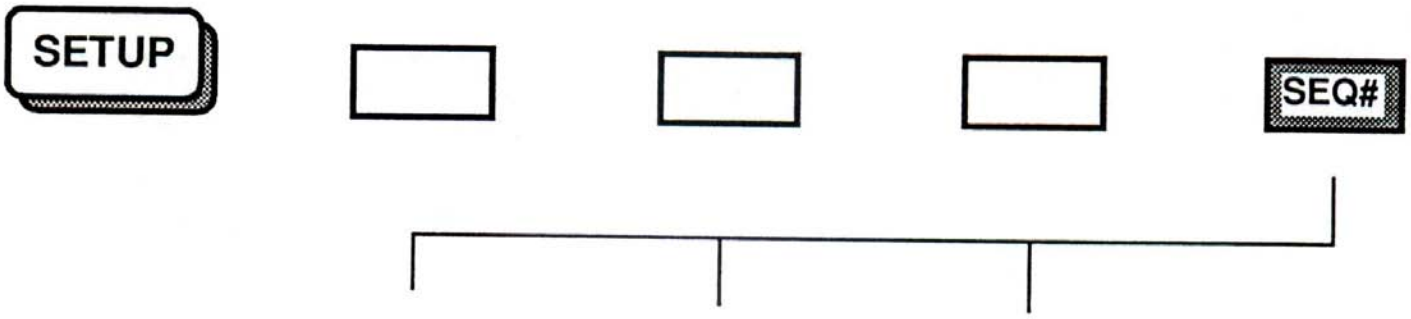
**CLOCK FREQ**
Set Sample Clock
Set Block Frequency

**AMPL OFST**
Set Amplitude
Set Offrset

## SETUP, UTILITY AND EDIT KEY GROUP



## SETUP, UTILITY AND EDIT KEY GROUP (Cont.)




**[DELF]**

**[OPEN]**
Set Step Number

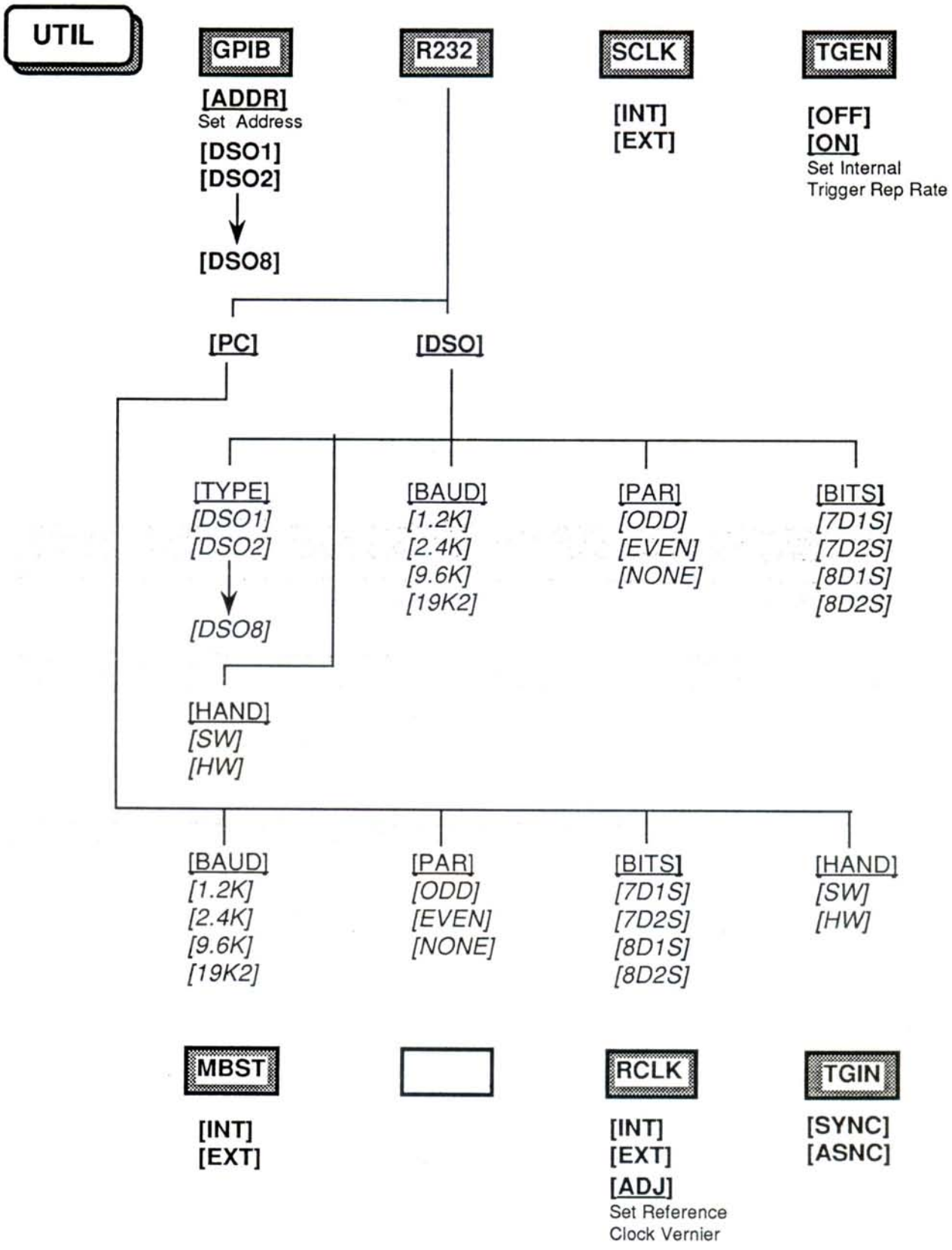
[DELS]

[MODS]
Set Waveform #
Set # Cycles

[ADDS]
Set New Step #
Set Waveform #
Set # Cycles

**[NEW]**
Set New Sequence #

## SETUP, UTILITY AND EDIT KEY GROUP (Cont.)




**PNTS**

**[ANCH]**
Set Left Anchor (AL)
Set Right Anchor (AR)

**[MODP]**
Increment X Value (PX+)
Decrement X Value (PX-)

**LINE**

**[STRT]**
Set X Address (LX)
Set Y Address (LY)

**[ANCH]**
Set Left Anchor (AL)
Set Right Anchor (AR)

**[CHRD]**
Set Chord X
Address (CX)
Set Chord Y
Address (CY)

**[ZLVL]**
Set Z Amplitude

**VRTX**

**[ADDV]**
Set Vertex X
Address (VX)
Set Vertex Y
Address (VY)

**[SCAL]**
Set Digital
Amplitude (DA)
Set Digital
Offset (DO)

**[ANCH]**
Set Left Anchor (AL)
Set Right Anchor (AR)

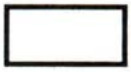
**[INSF]**
Select Function
[SHOW]
[-->]
Set Waveform
Parameters

**MATH**

**[◀]**
Select Three
Waveform Numbers

**[OP]**
[A*B]
[A+B]
[A-B]

**VRTX**



Select Waveform Number

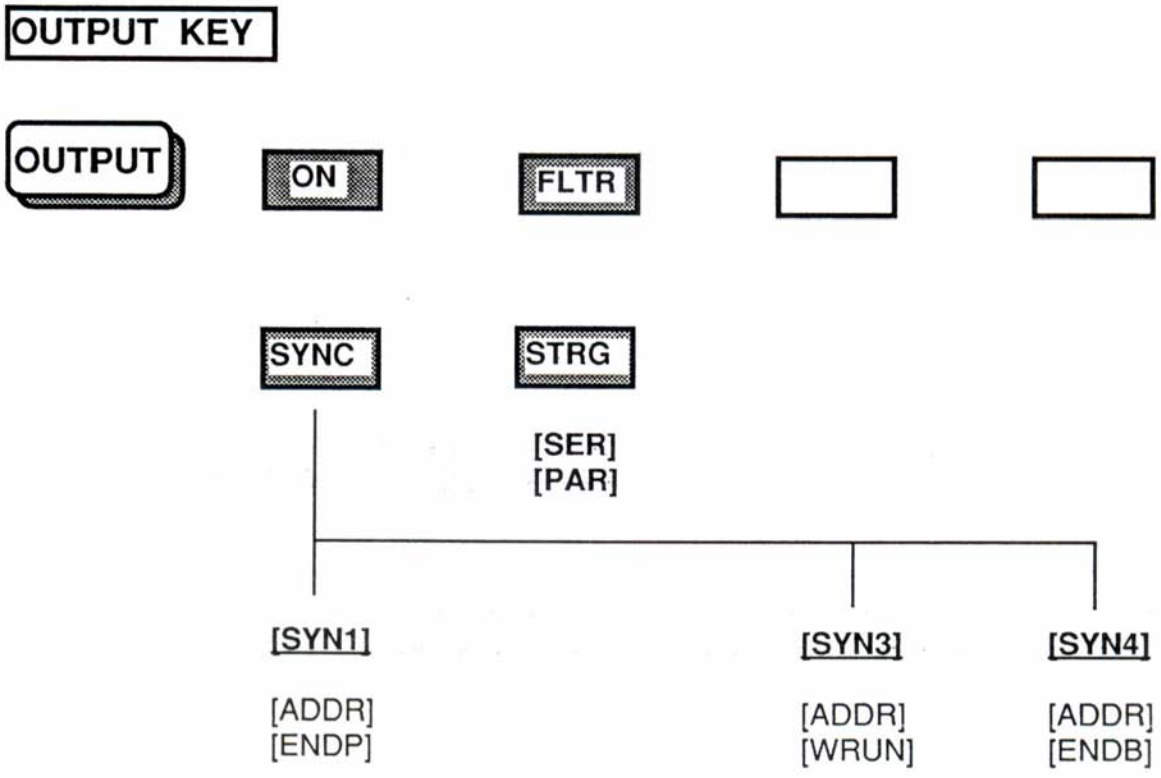
**[SUMF]**
Select Function
Select Ancillary
Function
[SHOW]

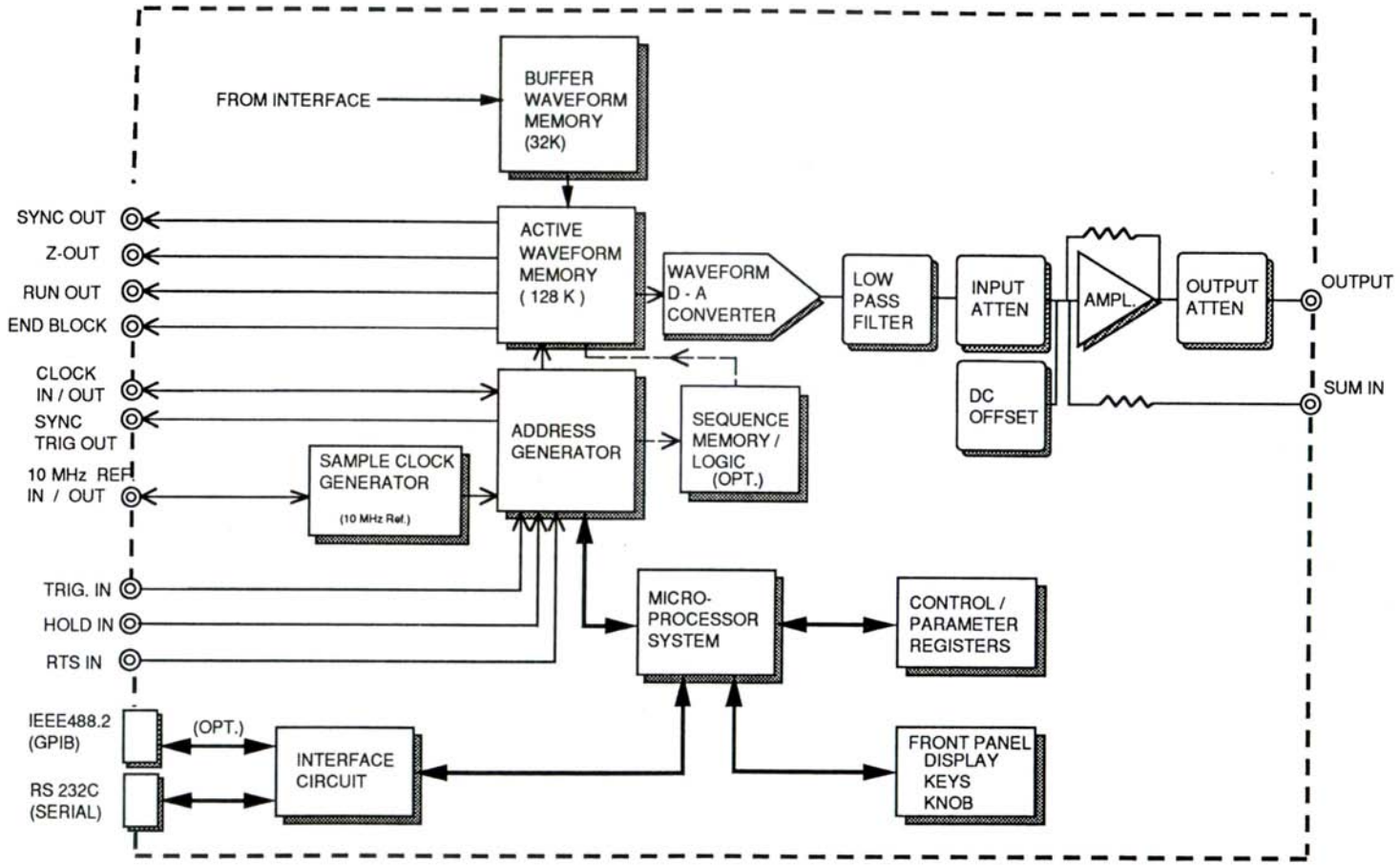
**[DMPF]**
Select Function
Select Ancillary
Function

**[SMOO]**
Select # Samples
to Average
[SHOW]

**[MOVE]**
Set Anchors
[COPY]
[PSTE]


STOR
RECL
STOR
Set Stored
Setting Number
RECL
Set Stored
Setting Number


OUTPUT KEY
OUTPUT
ON
FLTR
SYNC
STRG
[SER]
[PAR]
[SYN1]
[ADDR]
[ENDP]
[SYN3]
[ADDR]
[WRUN]
[SYN4]
[ADDR]
[ENDB]




BLOCK DIAGRAM

# WAVEFORM / SEQUENCE WORK SHEET

## TEGAM 2414A

**STANDARD WAVEFORM**

| | LENGTH |
|---|---|
| STDW | |

SEQUENCE # _____

APPLICATION _____

_____

_____

**CUSTOM WAVEFORMS**

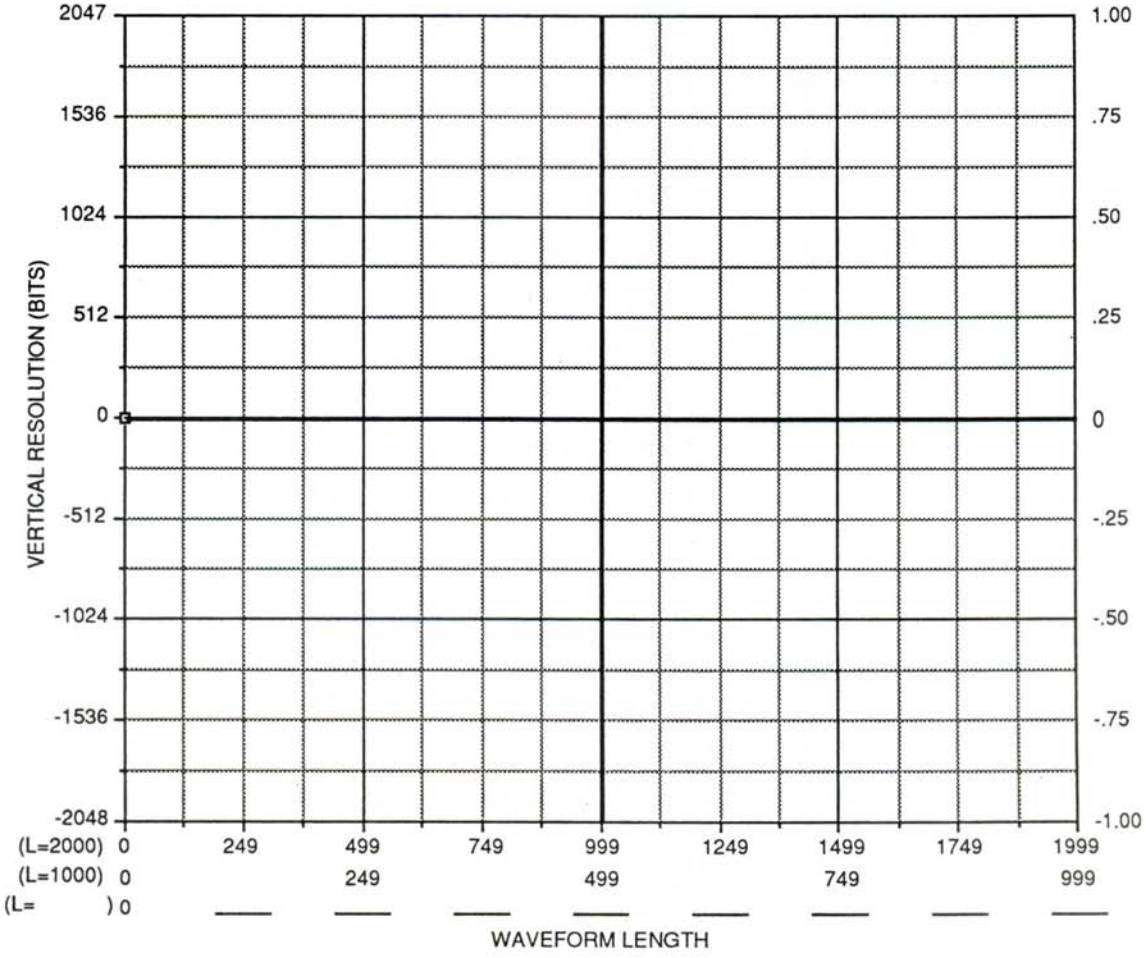
| WAV # | LENGTH | TYPE |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

| STEP # | WAV # | NUMBER of BURST |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# WAVEFORM DESIGN SHEET

**WAV #** _____ **WAVEFORM TYPE**_____

**LENgth**_____




APPLICATION

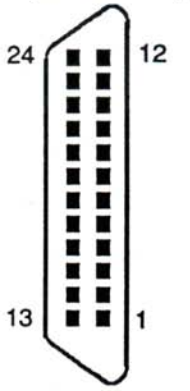
## DICTIONARY OF TERMS

| | |
|---|---|
| ADDR | Address |
| ADDS | Add Step |
| ADDV | Add Vertex |
| AM | Amplitude Modulation |
| AMPL | Amplitude |
| ANCH | Anchor |
| ASNC | Asynchronous |
| BRST | Burst |
| BUFR | Buffer |
| CANC | Cancel |
| CHRD | Chord |
| CIRC | Circle |
| CONT | Continuous |
| DEL | Delete |
| DELF | Delete File |
| DELS | Delete Step |
| DSO | Digital Storage Oscilloscope |
| EXP | Exponential |
| FLTR | Filter |
| FM | Frequency Modulation |
| FREQ | Frequency |
| FUNC | Function |
| GAUS | Gaussian |
| GPIB | General Purpose Interface Bus |
| HSIN | Haversine |
| INSF | Insert Function |
| LCD | Liquid-Crystal Display |
| LEN | Length |
| MBST | Monitor Burst Count |
| MODP | Modify Point |
| NOIS | Noise |
| OFST | Offset |
| OP | Operation |
| PAR | Parallel |
| PLS | Pulse |
| PNTS | Points |
| R232 | RS-232C Interface Bus |
| RCLK | Reference Clock |
| RECL | Recall |
| RTS | Return To Start |
| SAW | Sawtooth |
| SCAL | Scale |
| SCLK | Sample Clock |
| SCM | Suppressed-Carrier Modulation |
| SEQ# | Sequence Number |
| SER | Serial |
| SIN | Sine |
| SLIN | Linear Sweep |
| SLOG | Logarithmic Sweep |
| SMOO | Smooth |
| SQR | Square |
| STDW | Standard Wave |
| STOR | Store |
| STRG | Synchronous Trigger |
| STRT | Start |
| SUMF | Sum Function |
| SX/X | Sine X/X |
| TGEN | Trigger Generator |
| TGIN | Trigger In |
| TOGL | Toggle |
| TRI | Triangle |
| TRIG | Trigger |
| UTIL | Utilities |
| VRTX | Vertex |
| WAV# | Waveform Number |
| ZLVL | Z-Axis Amplitude |

# GPIB AND RS-232 CONNECTOR PINOUTS

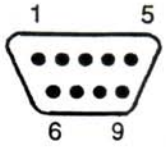
## GPIB

GPIB 24 PIN
(REAR VIEW)




| | |
|---|---|
| 24: GND | 12: SHIELD (CHASSIS GND) |
| 23: GND | 11: ATN |
| 22: GND | 10: SRQ |
| 21: GND | 9: IFC |
| 20: GND | 8: NDAC |
| 19: GND | 7: NRFD |
| 18: GND | 6: DAV |
| 17: REN | 5: EOI |
| 16: DIO8 | 4: DIO4 |
| 15: DIO7 | 3: DIO3 |
| 14: DIO6 | 2: DIO2 |
| 13: DIO5 | 1: DIO1 |

## RS-232C




1: 6.2k Pulldown
2: RXD
3: TXD
4: DTR
5: Ground
6: DSR
7: RTS
8: CTS
9: NC

# STANDARD WAVESHAPE EQUATIONS

The following ten equations show the mathematical basis for the algorithms used to create the indicated waveforms:

$fm(t) = \sin[\omega_c t + \beta \cos(\omega_m t + \phi_m) + \phi_c]$ $\beta$ = Modulation Index

$am(t) = \sin(\omega_c t + \phi_c) * [1 + M \sin(\omega_m t + \phi_m)]$ M = Modulation Index

$scm(t) = \sin(\omega_c t + \phi_c) * \sin(\omega_m t + \phi_m)$

$exp+(t) = e^{-kt}$ k = Exponential Time Constant

$exp-(t) = 1 - e^{-kt}$ k = Exponential Time Constant

$sinx/x(t) = \frac{\sin(t)}{t}$

$gauss(t) = e^{-t^2}$

$circle(t) = \sqrt{1 - (1 - 2t)^2}$

$linsweep(t) = \sin\left(\omega_{begin}\, t + K\int t\, dt\right)$ K is the slope of the sweep $= \frac{(\omega_{end} - \omega_{begin})}{2}$

$logsweep(T) = \sin\left(\omega_{begin} T * \int_0^T e^{tK} dt\right)$ $K = \frac{\log(\omega_{end}) - \log(\omega_{begin})}{SweepTime}$

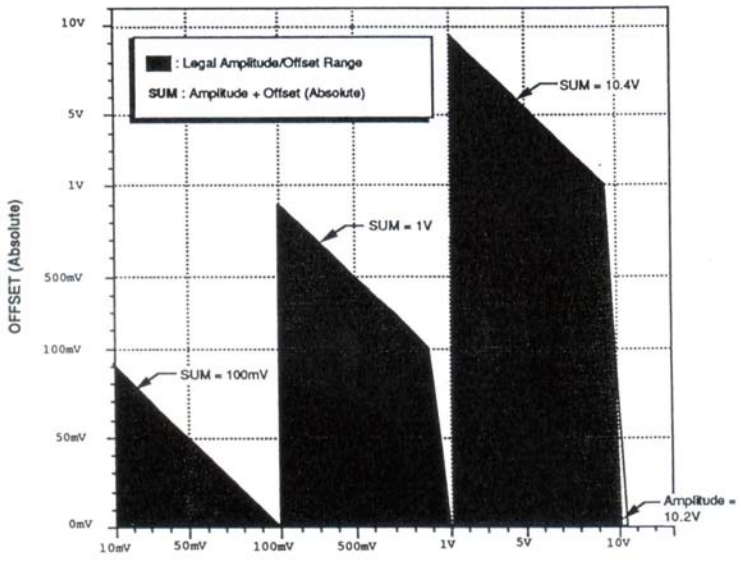
## ERROR CODES

Certain instrument settings are not compatible with one another and will result in error messages. The error codes and messages are as follows:

**ERROR 101**
**AMPL/OFST Range**

Amplitude and/or offset are outside of allowable ranges. Allowable ranges are Illustrated in the graph below:

AMPLITUDE/OFFSET GRAPH




AMPLITUDE

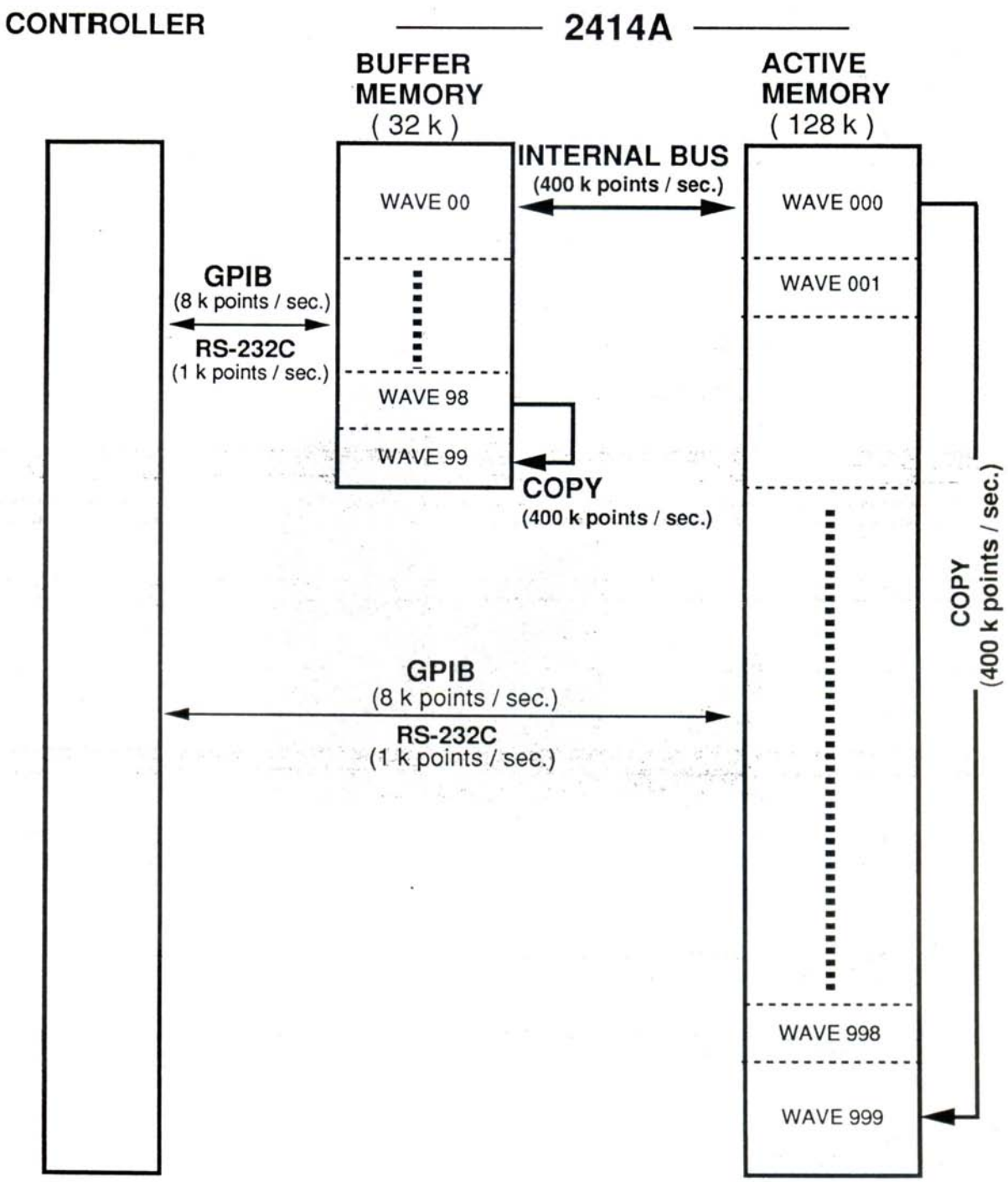
**ERROR 102**
**START/STOP Range**

This message appears in the VIEW menu when the View Left address is not less than the View Right address.

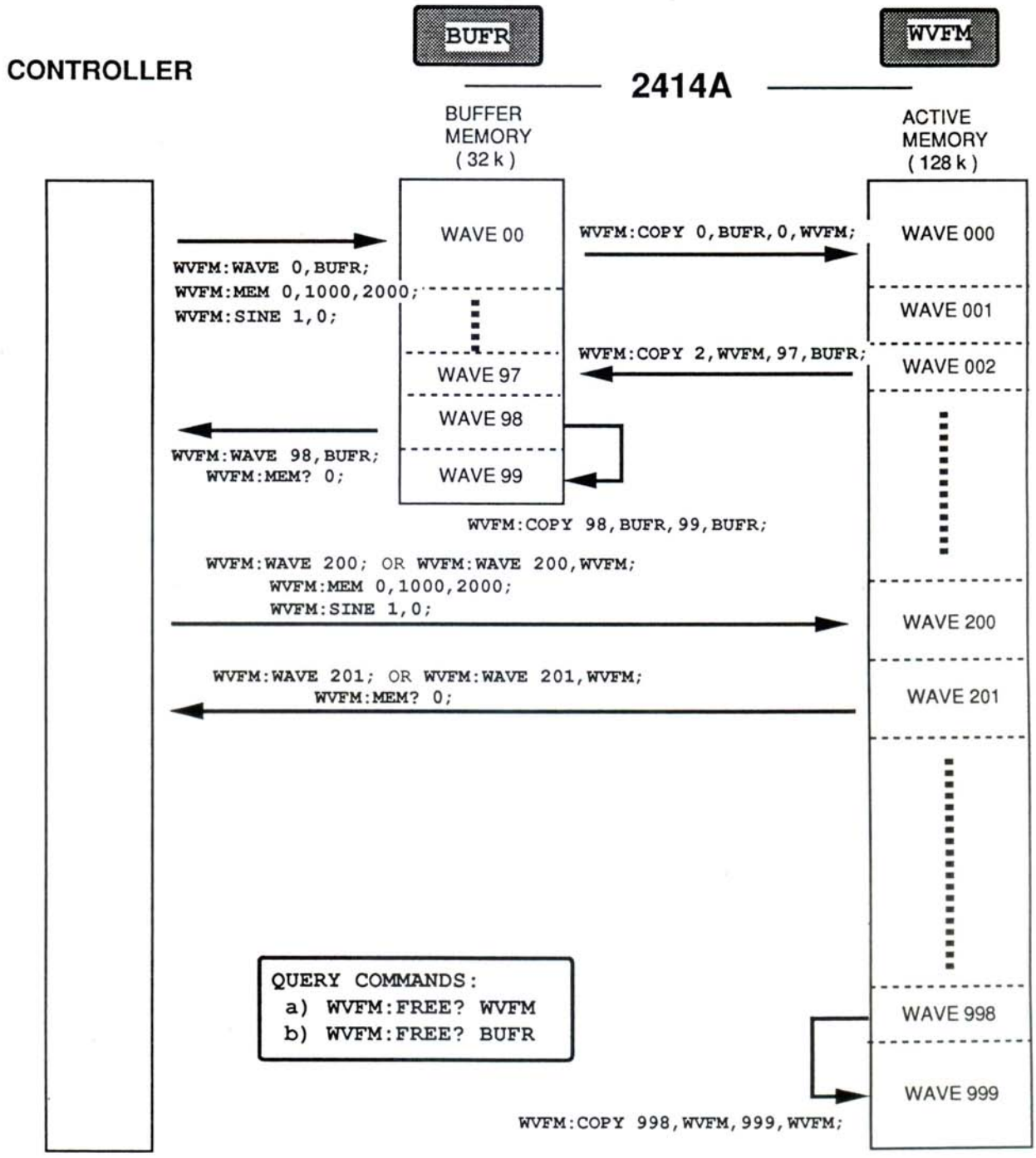
**ERROR 104**
**FUNC/MODE Conflict**

This message appears if Burst Mode and Sequence are both selected.

# WAVEFORM MEMORY ORGANIZATION



# WAVEFORM MEMORY REMOTE COMMANDS



**DSOLink™ Setup For Biomation 4080 DSO**

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

Setup of DSO4080:
[MASTER MENU]
{6 PLOT}
{1 plot mode == SINGLE}
{2 plot output == GPIB}
{3 graticule plot == OFF}
{4 graticule style == SOLID}
{5 plot traces only == ON}
[MASTER MENU]
{7 I/O INTERFACES}
{1 gpib address == 07}
[MENU/TRACE]

Setup of 2414A:
[UTIL]
{GPIB}
{DSO1}
{WAVE == 0}
[FUNC]
{WAV#}
{WAVE == 0}
[ENTER]

dso download:
Press [PLOT] key on DSO4080

# DSOLink™ Setup For Hewlett-Packard 54510B DSO

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

[UTIL]
{HP-IB menu}
{ talk only }
{ time out == 100}
{ initialize == off}
{ plot == Display }
{ device mode == plot}
{ pen Channel 1 == 1}
{ exit }

setup of 2414A:
[UTIL]
{GPIB}
{DSO1}
{WAVE == 0}
[FUNC]
{WAV#}
{WAVE == 0}
[ENTER]

dso download:
press [HARD COPY] on HP 54510B

## DSOLink™ Setup For Hitachi VC6155 DSO

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

Setup of VC6155: Set rear panel dip switches for 9600 baud, 8-data, 1-stop. Press [STORAGE] button, button will flash (green); Adjust Volts/DIV knob until CH1 waveform size is full screen. Press [HOLD], button will turn red.

Setup of 2414A:

[UTIL]
    {R232}
        {DSO}
        {WAVE == 0}
        {TYPE}
            {DSO3}
            [LAST]
        {BAUD}
            {9600}
[FUNC]
    {WAV#}
        {WAVE == 0}
[ENTER]

dso download:
Press [PLOT] key on VC6155

**DSOLink™ Setup For Philips PM3375 DSO**

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

setup of PM3375:
press and hold [MENU] then press [AUTO SET]

```
{APPL}
    {PLOT}
        {PLOT-KEY} ; press until top line == "FUNCTION: plot_digital"
        {PLOT_D}
            {\/ FORMAT /\} ; press until top line == "FORMAT: 1.0"
            {TYPE} ; press until top line == "TYPE: HP7475A"
            {AUTO} ; press until top line == "AUTO: off"
            {RETURN}
        {RETURN}
    {IEEE}
        {T/L} ; press until top line == "MODE: talk_only"
        {RETURN}
[AUTO SET]
```

setup of 2414A:

```
[UTIL]
    {GPIB}
        {DSO1}
        {WAVE == 0}
[FUNC]
    {WAV#}
        {WAVE == 0}
[SETUP]
    {WAV#}
        {WAV000} ; turn knob until "WAV000" appears
        {LEN}
            {LEN == 4093}
            {OK}

[ENTER]
```

for dso download:
press [PLOT] key on pm3375

**DSOLink™ Setup For Tektronix 2232 DSO**

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

Bit switches:

| Switch | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|
| ON =1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |

[SETUP]
{PLOT}
PLOTER TYPE = HPGL
GRAT = OFF Auto
Plot = OFF

2414A
Setup [UTIL]
{R232}
{DSO}
{TYPE}
{DSO2}
[LAST]
{BAUD}
{2.4K}
[LAST]
{PAR}
{NONE}
[LAST]
{BITS}
{8DS1}
[LAST]
[->]
{HAND}
{SW}
[ENTER]

To plot waveform press:
[SETUP]
{PLOT}
{START}
on Tek2232

## DSOLink™ Setup For Tektronix 2440 DSO

[ ] Represents a Designated Hard Key
{ } Represents a Soft Key

[SETUP OUTPUT]
{SETUP}
{MODE}
{DEVICES}
{HPGL PLOTTER}
{SETUP}

| | |
|---|---|
| Settings | OFF |
| Text | OFF |
| Graph | OFF |
| Wvfm | ON |
| Pg Size | US |

2414A
Setup [UTIL]
{GPIB}
{DSO2}
[ENTER]

To plot waveform press:
[SETUP OUTPUT]
{PLOT}
on the Tek2440.

## Warranty

TEGAM, Inc. warrants this product to be free from defects in material and workmanship for a period of one year from date of shipment. During the warranty period, we will at our option, either repair or replace any product that proves to be defective.

To exercise this warranty, contact TEGAM, Inc., Ten Tegam Way, Geneva, Ohio 44041/FAX (440) 466-6110/(440) 466-6100, M-F, 8 a.m.- 5 p.m. ET. You will be given prompt assistance and return instructions. Send the instrument, transportation prepaid, to the indicated service facility. Repairs will be made and the instrument returned, transportation prepaid. Repaired products are warranted for the balance of the original warranty, or at least 90 days, whichever is longer.

## LIMITATION OF WARRANTY

TEGAM, Inc., warranty does not apply to defects resulting from unauthorized modification or misuse of any product or part. This warranty also does not apply to fuses, batteries, or damage from battery leakage.

This warranty is in lieu of all other warranties, expressed or implied, including any implied warranty of merchantability or fitness for a particular use. TEGAM, Inc. shall not be liable for any indirect, special or consequential damages.

## STATEMENT OF CALIBRATION

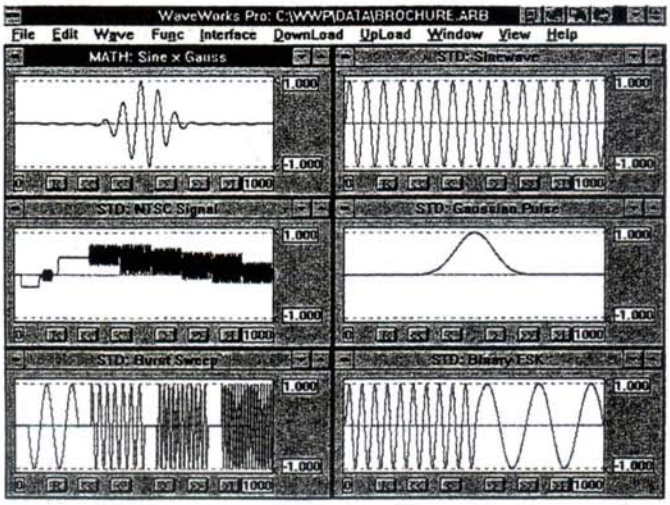
This instrument has been inspected and tested in accordance with specifications published by TEGAM, Inc.

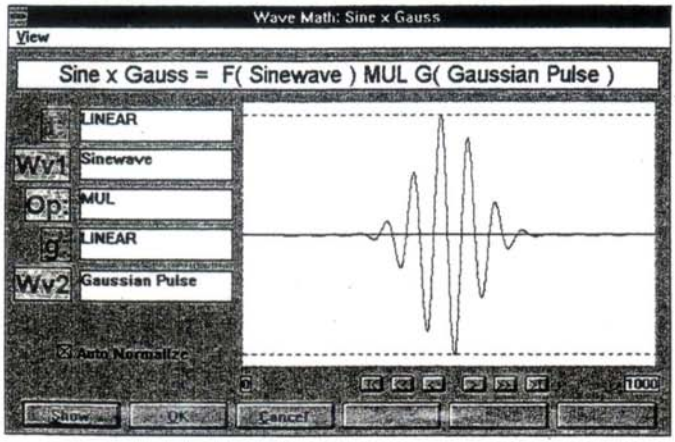
The accuracy and calibration of this instrument are traceable to the National Institute of Standards and Technology through equipment, which is calibrated at planned intervals, by comparison to certified standards maintained in the Laboratories of TEGAM, Inc.

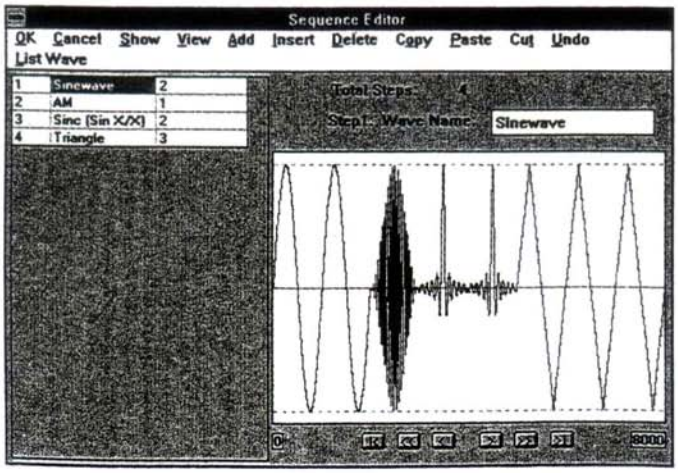
# WaveWorks Pro™

## Waveform Creation Software for Windows™

## The simplest, most intuitive way to make the waveforms you need.




New WaveWorks Pro™ turns your computer screen into a waveform palette. An extensive waveform library with a complete set of design and editing tools. Works with all TEGAM waveform generators. *Now you can solve all your waveform needs like a pro!*

This prolific software is a virtual function generator with unlimited real waveforms and control parameters. A comprehensive array of math operations and transfer functions to design your most demanding waveshape. Synthesis in both the time and frequency domain is provided by FFT and IFFT routines. Import spreadsheet files, accomplish data analysis, facilitate documentation and report writing with ease.

## FEATURES

- **Functions**
  - 30 Standard Waveforms
  - 20 Math Transfer Functions
  - 13 Math Operations
  - Digital Pattern
  - FFT and IFFT
  - Sequence Programming
- **File Import / Export**
  - 7 Formats including Common ASCII Formats (.CSV, .PRN)
- **GPIB and RS-232 Support**
  - Waveform Data Download / Upload
  - Instrument Control Panel
- **Data Import from Popular DSOs**

# WaveWorks Pro™

Waveform Creation Software for Everybody

## STANDARD WAVES

All 30 standard waves with the required parameters are set up in the selected screen for instantaneous use.

| | | | |
|---|---|---|---|
| **Sine** | **Cosine** | **Square** | **Triangle** |
| **DC** | **Ramp** | **Squine** | **Gaussian** |
| **Pulse1** | **Pulse2** | **VHR Pulse** | **SinX/X** |
| **AM** | **FM** | **PWM** | **BFSK** |
| **BPSK** | **Lines** | **NTSC** | **Comb** |
| **FIR(LPF)** | **HAN(SinX/X)** | **Steps** | |
| **Cont. Sweep** | **Step Sweep** | **Burst Sweep** | |
| **Exponential** | **Analog Noise** | **Digital Noise** | |

## WAVEFORM GRAPHIC MATH

### Math Transfer Function

All 20 transfer functions including integration and differentiation are available to modify the waveform in the specified manner.

| | | | |
|---|---|---|---|
| **Null** | **Linear** | **Section** | **Square** |
| **Absolute** | **Cubic** | **Log** | **Square Root** |
| **Exponential** | **Polynomial** | **Integration** | **Band Pass** |
| **Differentiation** | **DC Cut** | **Normalize** | **Rotate** |
| **Mirror** | **I-Phase** | **Q-Phase** | **I/Q Swap** |

### Math Operator

Complex waveforms are readily created by use of the 13 operators. Each of the waveforms in the set are included in developing the final result which is downloaded to the generator memory.

| | | |
|---|---|---|
| **Addition** | **Subtraction** | **Multiplication** |
| **Division** | **Cascade** | **Insert** |
| **Add_Into** | **Convolution** | **FIR Filter** |
| **AM** | **FM** | **QAM** |

## SEQUENCE PROGRAMMING

Sequence offers virtual memory expansion and comprehensive real-world simulations. Any waveform stored in memory may be output in any order in a seamless manner and be repeated up to 1,000,000 times. Test profiles are easily defined by the use of a table to select each waveform in turn and designating the number of times each is looped.

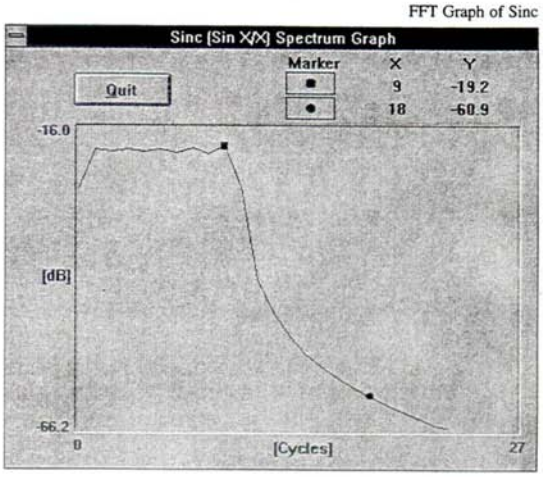
## EDIT

| | | |
|---|---|---|
| **Copy** | **Paste** | **Point** |
| **Vertex** | **Digital Pattern** | **Harmonics (FFT,IFFT)** |
| **Sequence** | | |

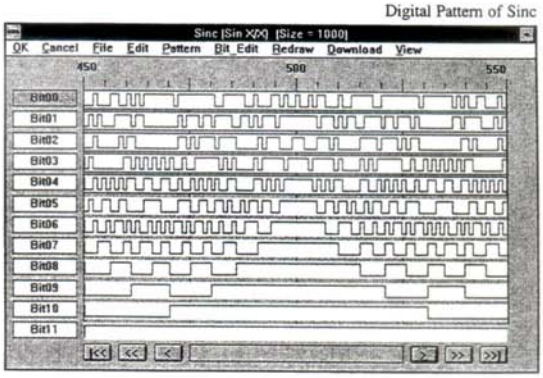
## AUXILIARY FEATURES

Fast Fourier transform offers frequency analysis and graphical presentation for frequency spectrum editing. Recreation of the time-domain waveform is offered using the inverse transform. A complete sample value tabulation of each waveform allows for point-by-point modification. A digital pattern describes the bit-by-bit synthesis of each waveform for editing in this arena.

FFT Graph of Sinc




Digital Pattern of Sinc




TEGAM, Inc. proudly presents its extensive line of waveform generators. The company offers extraordinary customer support providing you waveform creation assistance and application solutions. We can meet all your waveform needs.

# Model2414A
Arbitrary Waveform Generator

## Calibration Verification Procedure

This document provides a procedure to verify that the Model 2414A specifications are within the tolerances listed in the published data sheet. TEGAM, Inc. recommends an annual calibration verification interval.

Test Equipment Required:

Analog Oscilloscope, 100 MHz or greater
High-Stability Reference Frequency Counter, ≤ 10ppm
True-RMS Digital Multimeter, 5-1/2 digits
Distortion Analyzer (Sound Technology 1701 A, or equivalent)
Precision 50Ω load, ±0.1%

Frequency Accuracy:

1) Connect frequency counter to 10 MHz Reference Clock connector on rear panel.

2) Verify that frequency is 10 MHz ± 50 ppm (9,999,500 to 10,000,500 Hz).

**CAUTION**

> *The following procedures reset the instrument to default settings.* Control *setting can be saved using the* Store/Recall *function.* Stored Waveforms *will not be erased if only the* Reset *Current function is used. (Do not use* Reset All.)

Waveform Rise/Fall Time:

1) Select RESET and CURRent and press OK.

2) Amplitude = 10 volts.

**3)** Function = Square.

4) Output On, ENTER.

5) Connect ARB Out to oscilloscope through 50Ω load.

6) Verify rise and fall times are less than 20 ns.

Offset Accuracy:

1) Select RESET and CURRent and press OK.

2) Function = Sine.

3) Amplitude = 1 V.

4) Output On.

5) Connect ARB Out to DMM (No load).

6) Offset = 9.000 V, ENTER.

7) Verify DC offset is between 8.890 V and 9.110 V.

8) Amplitude = 100 mV.

9) Offset = 900 mV, ENTER.

10) Verify DC offset is between 898 mV and 932 mV.

11) Amplitude = 10mV.

12) Offset = 90 mV. ENTER

13) Verify DC offset is between 84.5 mV and 95.5 mV.

Amplitude Accuracy;

1) Select RESET and CURRent and press OK.

2) Function = Sine.

3) Clock = 1 MHz.

4) Output On.

5) Connect ARE Out to DMM (No load).

6) Amplitude = 10.2 V, ENTER.

7) Measure true RMS amplitude on DMM and convert to peak value by multiplying by 1.414.

8) Verify peak value is between 10.078 V and 10.322 V.

9) Amplitude = 900 mV, ENTER.

10) Measure true RMS amplitude on DMM and convert to peak value by multiplying by 1.414.

11) Verify peak value is between 868 mV and 932 mV.

12) Amplitude = 90 mV, ENTER.
13) Measure true RMS amplitude on DMM and convert to peak value by multiplying by 1.414.

14) Verify peak value is between 84.5 mV and 95.5 mV.

Total Harmonic Distortion + Noise:

1) Select RESET and CURRent and press OK.

2) Function = Sine.

3) Amplitude = 9.1 V.

4) Output On.

5) Filter On (From OUTPUT menu).

6) Clock = 10 MHz.

7) Press ENTER.

8) Connect ARB Out to distortion analyzer through 50Ω load.

9) Set distortion analyzer for a 10 kHz signal and 80 kHz measurement bandwidth.

10) Verify THD + noise is less than -65 dB.

For further information, please contact us via phone fax or e-mail:
Phone: 1-800-666-1010

Or 440-466-6100

Fax: 440-466-6110

E-mail: sales@tegam.com

# INDEX

# INDEX


active area 10-42,43
add 1-1, 6-1, 6-9
add sequence (ADDS) 7-1 .
add step (ADDS) 7-4
all key 3-2
AM 1-2, 10-26, 10-31
amplitude (AMPL) 1-2, 3-1, 3-4, 10-25
anchor (ANCH) 6-1, 6-2, 6-4, 6-6, 6-8
arbitrary 6-1
arbitrary block data 10-10,10-17
asynchronous 8-1, 8-2
baud rate (RS-232) 10-5
binary 10-17, 10-45
bits (RS-232) 10-5
block transfer 10-34
buffer memory 1-1,1-2,10-34,10-42
burst 1-2, 3-4, 9-2, 9-3, 10-25
cable (RS-232) 10-3,10-4
chord (CHRD) 6-4
circle 1-2, 10-26, 10-31
clear registers 10-22
clock 1-1, 1-2, 1-3, 3-4, 4-2, 8-1, 8-2
clock select 10-25
commands
  common 10-1, 10-6,10-12
  configuration 10-1
  execution 10-19
  hierarchy 10-20
  instrument specific 10-12
  output 10-2
  path 10-20
  sequence 10-19
  sequence generator 10-2
  system 10-1
  waveform editing 10-2
continuous 1-2, 3-4, 10-25
copy 6-8,10-34
current (CURR) 3-2
DC 1-2, 10-27
decimal numeric data 10-9,10-16, 10-44
defaults 10-40
delay 10-28, 10-32
delete 5-2
delete file (DELF) 7-4
delete step (DELS) 7-4
digital amplitude (DA) 6-6
digital offset (DO) 6-6
digital synthesis 4-1
DSOLink™ 9-4
dump function (DMPF) 6-7
edit waveform 4-1, 6-2, 10-26, 10-42
end block 1-2, 9-1, 9-2
end pulse 9-1, 9-2
errors 10-18
event status register (ESR) 10-7, 10-13
exponential 1-2
falltime 10-28, 10-32
filter 1-2,9-2
FM 1-2, 10-27, 10-31
frequency 3-1
fuse 2-1
gated 1-2, 3-4, 9-2, 10-25
gaussian 1-2, 10-27, 10-31
GPIB 1-1, 1-3, 6-1, 10-1, 10-5 ,10-12 10-40
handshake (RS-232) 10-5
haversine 1-2, 10-27, 10-31
hold 1-2, 1-3, 9-3, 10-23
IEEE-488.2 1-1, 10-6, 10-12
insert function (INSF) 6-6
internal trigger 9-3
length (LEN) 5-2, 10-29, 10-32, 10-43
line 1-1, 4-1, 6-1, 6-2, 10-27
line mode 6-1
linear sweep 10-27, 10-31
log sweep 10-27, 10-32
loop and link 1-1, 1-2, 7-1
low-pass filter 4-3, 9-2
math 1-3, 6-8, 6-12
memory 3-4, 5-1
memory blocks 5-1
memory formats 10-44
mode 3-4
modifying a step (MODS) 7-4
monitor burst (MBST) 9-3
mouse 6-1, 6-4, 6-6
move 6-8
multiple units 8-1, 10-26
multiply 1-1, 6-1, 6-9
new 5-2, 6-10, 6-12, 7-1
noise 1-2
offset (OFST) 1-2, 3-1, 3-4, 10-25
open 7-1, 7-4
output 3-2, 10-25
parity (RS-232) 10-5
parallel 8-1, 10-26
partitioning 5-1

paste (PSTE) 6-8
point edit 6-8
power 1-3, 2-1
protocol 10-5, 10-6, 10-15
program message 10-7, 10-8, 10-15, 10-16
pulse 1-2
random-access-memory (RAM) 4-1
reference 1-2, 1-3, 10-23
reference clock (RCLK) 1-2, 1-3, 10-23
reset 3-2, 10-23, 10-40
resolution 1-2, 4-2
response message 10-8, 10-15
rise/fall time 1-2
RS-232 1-1, 1-3, 6-1**,** 10-3, 10-5, 10-41
return to start (RTS) 1-2, 1-3, 9-3, 10-23
run 1-2, 9-1, 9-2
sample clock (SCLK) 3-4, 10-23, 10-24
samples 3-4
sampling rate 1-2
sawtooth 1-2, 10-28, 10-32
scaling 6-4, 6-6
SCM 1-2, 10-28, 10-32
sequence 7-1
sequence generator 1-1, 1-2, 7-1, 9-2, 10-35, 10-36
sequence number (SEQ#) 7-1, 7-4
series 8-1, 8-2
service request enable (SRE) 10-12, 10-13, 10-14
setup 5-2, 6-10, 7-1, 7-4
shift 3-2 show 6-6
sin x/x 1-2, 10-28, 10-32
sine 1-2, 3-2, 10-28, 10-32
smoothing (SMOO) 6-6
softkeys 3-4
spectral purity 1-2
square 1-2, 10-28, 10-32
standard wave (STDW) 3-2, 5-1, 5-2, 10-29, 10-30, 10-33, 10-42
start (STRT) 6-1, 6-2, 9-1
status 10-13, 10-14
stop 9-1
stored settings 1-3, 10-24
subtract 1-1, 6-1, 6-9 sum 1-3
sum function (SUMF) 6-7
sweep 1-2, 10-27, 10-31, 10-32
sync 3-1, 9-1, 10-26, 10-30, 10-33
sync out 1-2, 9-2
sync trigger (STRG) 8-1
synchronize 8-2
synchronous 8-1, 8-2
syntax 10-6, 10-7, 10-14
synthesized 4-1
synthesizer 4-2
system commands 10-1
toggled 1-2, 3-4, 9-3, 10-25
triangle 1-2, 10-28, 10-32
trigger 1-3, 8-1, 8-2, 10-24, 10-26
trigger generator (TGEN) 1-3, 9-3
trigger in (TGIN) 8-2
triggered 1-2, 3-4, 9-3
utility 8-1, 9-3
vertex 1-1, 4-1, 6-1, 6-4
view 9-1
waveform edit commands (Arbw) 10-26
waveform edit commands (Stdw) 10-30
waveform number (WAV#) 5-1, 6-1, 6-2, 6-10, 6-12, 7-1
WaveWorks Pro™ Software 1-3, 9-5
Z-axis 6-2, 9-2
Z-axis level (ZLVL) 9-2
Z-out 1-2, 6-2, 9-2

TEGAM is a manufacturer of electronic test and measurement equipment for metrology, calibration, and production test. We also provide repair, calibration, and other support services for a wide variety of test and measurement equipment including RF power sensor calibration systems, RF attenuation measurement systems, resistance standards, ratio transformers, arbitrary waveform generators, micro-ohmmeters, LCR meters, handheld temperature calibrators, thermometers, humidity and temperature control devices, and more.

TEGAM also repairs and calibrates test and measurement equipment formerly manufactured by Electro-Scientific Industries (ESI), Gertsch, Keithley Instruments, Lucas Weinschel, and Pragmatic Instruments. A complete list can be viewed on our Product Service Directory at www.tegam.com

For more information about TEGAM and our products, please visit our website at www.tegam.com : or contact one of our customer service representatives at sales@tegam.com or 800-666-1010.

Ten Tegam Way,
Geneva, Ohio 44041

Telephone: (440) 466-6100
Fax: (440) 466-6110
E-mail: sales@tegam.com