User's Manual

DBS 8750

High-Precision Arbitrary Waveform Synthesizer

ANALOGIC®

The World Resource for Precision Signal Technology

Proprietary Statement

The information contained in this publication is derived in part from proprietary and patent data of the Analogic Corporation. This information has been prepared for the express purpose of assisting operating and maintenance personnel in the efficient use of the instrument described herein. Publication of this information does not convey any rights to reproduce it or to use it for any purpose other than in connection with the installation, operation, and maintenance of the equipment described herein.

P/N 82–5106
Revision 2

Warranty

Analogic warrants only to the original purchaser that this product, as purchased from Analogic or an Analogic distributor or dealer, will conform to the written specifications for a period of one year from the date of purchase. If the product fails to conform to these warranties, Analogic, as its sole and exclusive liability hereunder, will repair or replace the product and/or its components within a reasonable period of time if the product is returned to a Tektronix service center within the warranty period. These warranties are made upon the express condition that:

a. The purchaser promptly notify Tektronix in writing of any non-conformity with the above warranty including a detailed explanation of the alleged deficiencies.

b. The product is returned to a Tektronix service center at the buyer's expense after making suitable arrangements for performance of service.

c. When the product is returned for repair, a copy of the original bill of sale or invoice is sent with the product .

d. Analogic will not be liable for any incidental or consequential damages.

e. In the opinion of Analogic upon inspection, the product has not been misused, altered, or damaged due to abnormal handling and/or operation.

f. Repairs to the product and/or its components have not been made by anyone other than Analogic or one of its authorized repair agents.

g. The product has not been modified, altered, or changed in any manner by anyone other than Analogic or one of its authorized repair agents.

THIS WARRANTY EXCLUDES ALL OTHER WARRANTIES, WHETHER EXPRESSED OR IMPLIED, ORAL OR WRITTEN, INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.

No term, condition, understanding or agreement purporting to modify the terms of this warranty shall have any legal effect unless made in writing and signed by an authorized officer of Analogic and the purchaser.

Contents

Tables

Figures

Section 1
Introduction

1.1 PRODUCT DESCRIPTION

The DBS 8750 is a 400 kHz, 16-bit precision, VXI Message-Based Arbitrary Waveform Synthesizer. Using its command set, the DBS 8750 can generate highly-accurate standard and arbitrary waveforms on either of its two differential or single-ended output channels. Combining the superior performance of a dual digital-to-analog converter (DAC) with the speed and processing capability of a digital signal processor (DSP), the DBS 8750 provides precision analog output signals.

1.1.1 Functional Overview

The DBS 8750 (Figure 1-1) generates waveforms using the data placed in the DAC 1 or 2 Waveform Memory (32K words maximum per channel). The DAC Sampling Rate is controlled by either the 400-kHz Internal Sampling Rate Clock or an External Sampling Rate Clock received from the Front Panel CLOCK input or one of the TTL Trigger lines via the VXIbus. The waveform data can be obtained from any of the following sources:

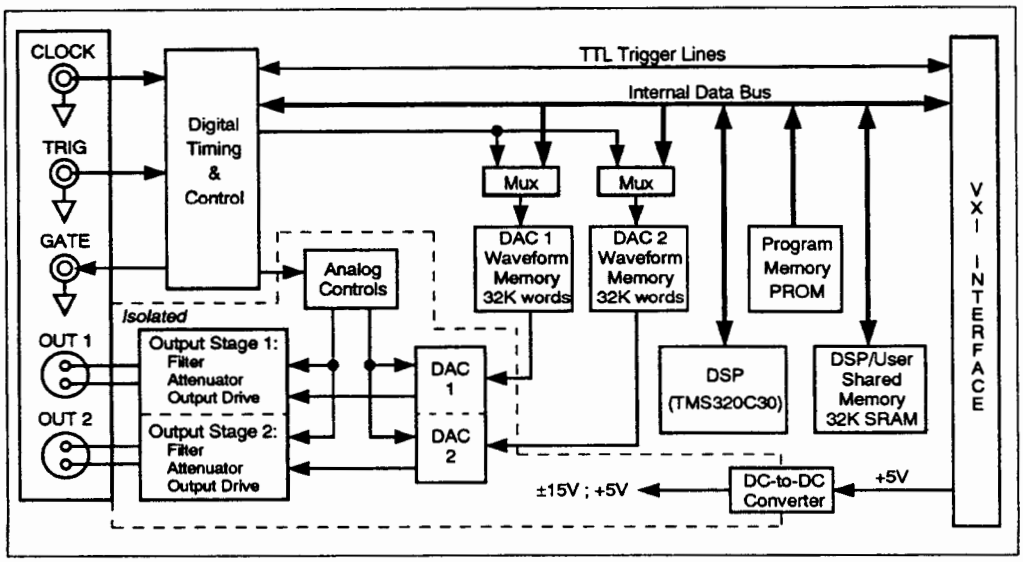
(1) Standard Waveform Library – Sine, Square, Triangle, Noise and DC functions.

(2) User-Defined Waveform Function – Complex waveforms calculated by the DSP according to a prescribed formula and user-specified parameters. Maximum waveform size is 32K points.

(3) DSP/User Memory. The shared DSP/User memory has 17150 words of user-accessible read/write space for saving waveforms and acquiring waveform data from the VXI system.

All control functions, complex waveform calculations and digital signal processing are implemented by the DSP. The DSP serves as an intelligent DMA controller for transferring digital data between the VXIbus and the DSP/User Memory. The DSP also acts as a counter/timing sequencer for cross-synchronization with other VXI instruments.

Figure 1-1. Waveform Synthesizer Block Diagram



On the front panel, an External Trigger input, an External Clock input and a Gate output provide synchronization controls. To cross-synchronize its operation with another VXI instrument, such as a DBS 8700 Digitizer, the DBS 8750 uses the VXIbus TTL Trigger lines to provide a Sampling Rate Clock and/or GATE output, or to receive an External Sampling Rate Clock.

NOTE: The DAC Waveform Memories are not accessible from the VXIbus. The shared DSP/User Memory has 17150 words of user-acccessible read/write space for data transfers to/from VXI memory or from the LOCALbus via connector P2.

After data has been loaded or processed, the DSP writes the final values into the DAC 1 or 2 Waveform Memory. In the write mode the memory is addressed by the DSP, while in the read mode the memory is addressed by a set of address counters. In turn, each DAC memory has its own independent address counter which allows independent waveform generation at each of the two outputs. While operating from pre-loaded commands and after the address counter has reached its last address, the channel loops back and continues to output the same waveform until a new command is entered at which time generation of a new waveform begins.

The analog output section is isolated from the digital section of the card by a 500-volt dc barrier. High-speed pulse transformers couple the data and control signals across the barrier. As each waveform memory outputs 16-bit parallel data words, a parallel-to-serial shift register feeds a serial bit stream over the barrier, while on the floating side, a serial-to-parallel shift register reverses the process. To maintain proper DAC synchronization, the 16-MHz master clock as well as the track-and-hold and load control signals are passed across the barrier using separate transformers.

Under precision timing control, digital data from each DAC memory is converted to analog signals by a highly-accurate 18-bit dual DAC. Each DAC converts data at the sampling rate and provides its own stable reference and two distortion-suppressing output deglitcher amplifiers.

Each DAC output is fed into a 6-pole reconstruction low-pass filter which can be switched into the analog path to eliminate higher frequencies when harmonically clean signals are required. A programmable attenuator feeds the filtered signal to the output amplifier. The output amplifier can be configured for either a single-ended or differential output, with AC or DC coupling, and with 50 or 600 ohms of output impedance. When finer attenuation steps are required, sending the appropriate commands to the DSP produces digitally scaled data within the full 16-bit resolution of the DACs.

A very low noise DC-to-DC converter designed to accommodate the stringent demands of a highly-accurate 16-bit waveform synthesizer provides power to the isolated analog section. Powered by +5 volts from connectors P1 and P2, the DC-to-DC converter provides regulated and filtered +15, -15 and +5 volts for the entire analog output circuitry.

1.1.2 Accessories

- ❑ DVX05-3 3-foot cable/connector assembly
- ❑ DVX05-5 5-foot cable/connector assembly

These assemblies consist of a TWBNC connector attached to either a 3-foot or 5-foot length of twisted shielded-pair cable. The other end of the cable has two cable conductors and a shield available for direct connection to the user's hardware application.

1.2 TECHNICAL SUPPORT

For service, contact your nearest Tektronix Service Center.

For technical support, call 1-800-835-4894 for technical support in the U.S., or contact your nearest Tektronix Office outside the U.S.

Section 2
Preparation & Installation

2.1 OVERVIEW

This section provides instructions for setting up hardware jumpers and installing the instrument in a VXI chassis. After unpacking, carefully inspect the hardware for any damage that may have occurred during shipment. If necessary, contact the carrier to file a claim. Save the shipping carton and packing materials if for any reason you need to return the product for repair or replacement. For technical support or to return this product to the factory, refer to Section 1.2 for instructions.

CAUTION:  **This product contains components which are sensitive to electrostatic discharge (ESD). Be sure to follow proper procedures for handling, storing and transporting ESD-sensitive assemblies.**

2.2 HARDWARE JUMPERS

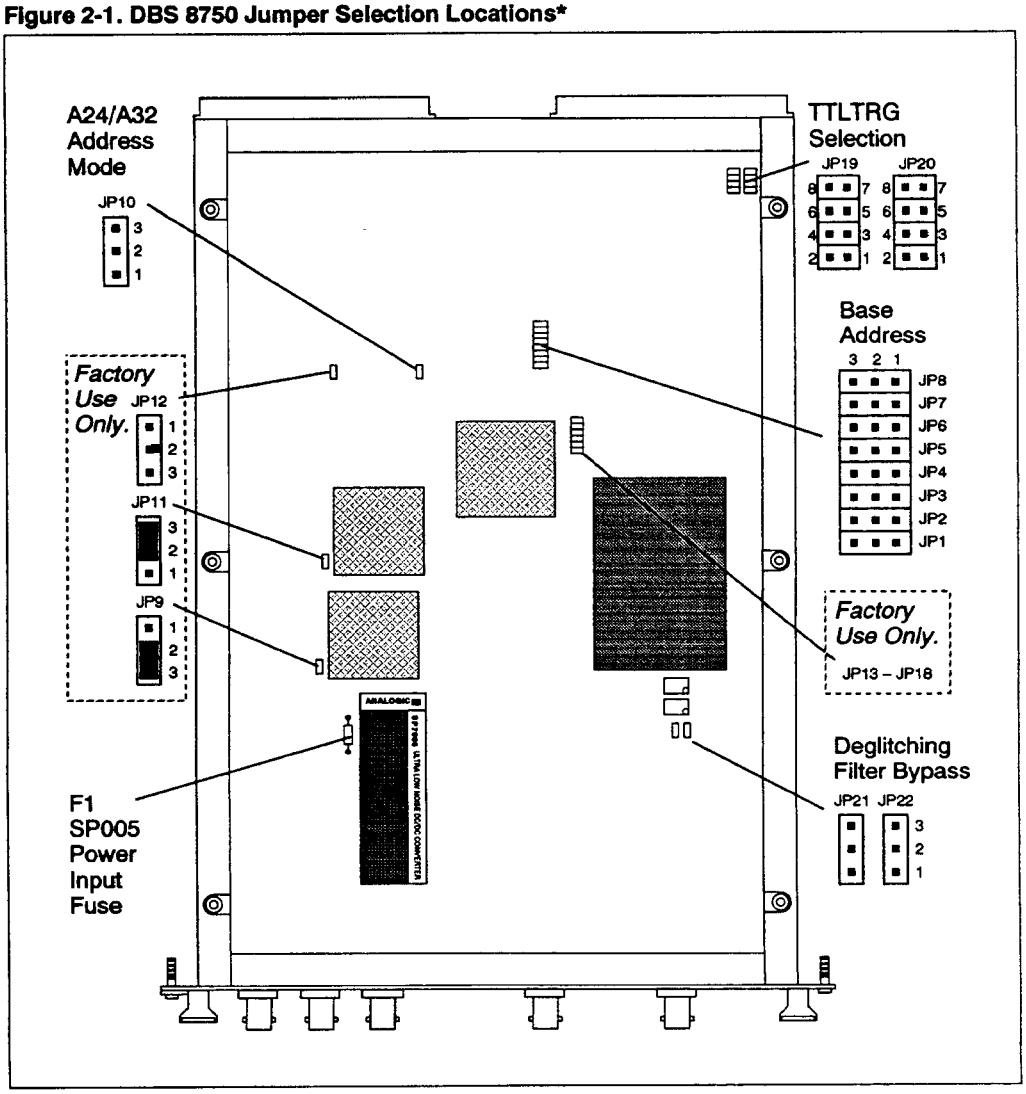
Before installing this instrument in a chassis, check Table 2-1 which lists jumper selections made at the factory before shipment To change any factory selection, refer to Figure 2-1 and the referenced Section.

Table 2-1. Hardware Jumper Factory Setup Summary*

Function	Sect	Factory Selection	Jumpers
VXI Logical Address	2.2.1	132	JP1-1 to JP1-2 JP2-1 to JP2-2 JP3-2 to JP3-3 JP4-1 to JP4-2 JP5-1 to JP5-2 JP6-1 to JP6-2 JP7-1 to JP7-2 JP8-2 to JP8-3
Address Space	2.2.2	A24	JP10-1 to JP10-2
Sampling Rate Clock Output	2.2.3	TTLTRG0	JP19-1 to JP19-2
Deglitching Filter	2.2.4	In Line	JP21-2 to JP21-3 JP22-2 to JP22-3
Factory Setup Only	N/A	N/A	JP9-2 to JP9-3 JP11-2 to JP11-3

* For hardware revision 1 and up only. See Appendix D for revision 0 hardware jumpers.

Figure 2-1. DBS 8750 Jumper Selection Locations*



NOTE: Jumpers JP9 and JP11 – JP18 are used only at the factory. They are installed at the factory in the following arrangement and should not be altered: JP9 and JP11 are always jumpered 2-3; JP12 and JP13 – JP18 are always open.

* For hardware revision 1 and up only. See Appendix D for revision 0 hardware jumpers.

2.2.1 Selecting the Base Address

The Base address of a VXI instrument is determined by the following:

Base Address = (Logical Address x 64) + 49152

The Logical Address of this instrument can be set from 0 to 255d using jumpers JP1 through JP8. However, Logical Address 0 and 1 cannot be used as these addresses are used by various VXI embedded controllers. Table 2-2 list some examples.

NOTE: *Do not use Logical Address 0 or 1. These are reserved for system controllers.*

Table 2-2. Logical Adddress Setup Examples

	VXIbus Address Lines & Jumper Selections							
Logical Address	VA13 JP8	VA12 JP7	VA11 JP6	VA10 JP5	VA9 JP4	VA8 JP3	VA7 JP2	VA6 JP1
255	2-3	2-3	2-3	2-3	2-3	2-3	2-3	2-3
254	2-3	2-3	2-3	2-3	2-3	2-3	2-3	1-2
252	2-3	2-3	2-3	2-3	2-3	2-3	1-2	1-2
248	2-3	2-3	2-3	2-3	2-3	1-2	1-2	1-2
240	2-3	2-3	2-3	2-3	1-2	1-2	1-2	1-2
224	2-3	2-3	2-3	1-2	1-2	1-2	1-2	1-2
192	2-3	2-3	1-2	1-2	1-2	1-2	1-2	1-2
132	2-3	1-2	1-2	1-2	1-2	2-3	1-2	1-2
2	1-2	1-2	1-2	1-2	1-2	1-2	2-3	1-2

2.2.2 Selecting the Address Space

The setting of hardware jumper JP10 enables the instrument to operate in either the A24 or A32 addressing mode. Refer to Table 2-3.

Table 2-3. Address Mode Selections

A24 Space	A32 Space
JP10-1 to JP10-2	JP10-2 to JP10-3

2.2.3 *Selecting TTL Trigger Lines*

The TTL trigger lines can be used for additional triggering and synchronization capabilities. As shown in Figure 2-2, the selected Sampling Rate Clock can be jumper-selected to drive TTLTRG line 0, 1, 2 or 3. The GATE output from the front panel can be jumper-selected to drive TTLTRG 4, 5, 6 or 7. Table 2-4 lists the available jumper selections. Before one of these signals can drive the bus, the TTLTRG line drivers must first be enabled by command OUTPut:TTLTrg ON | OFF.

Also, an External Sampling Rate Clock input can be received on any unused TTLTRG line. The input clock is software selected using the command, SOURce:ROSCillator:SOURce TTLTRG <0-7>.

Figure 2-2. TTLTRG Interface†

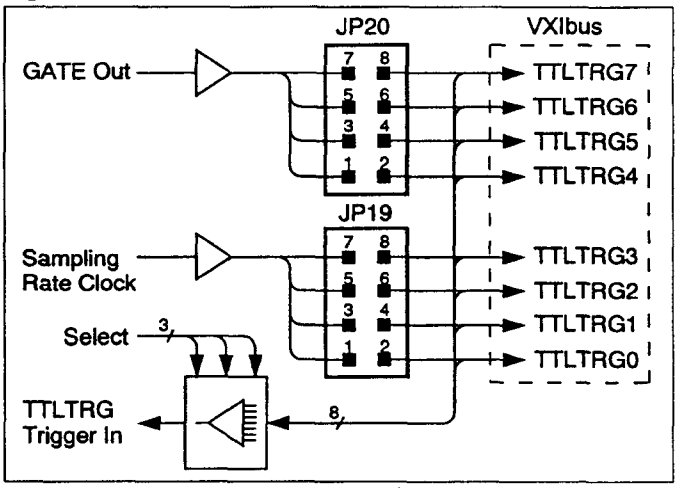


Table 2-4. TTL Trigger Line Output Selections†

Jumper	Setting	Connection
JP19	1 to 2*	Sampling Rate Clock onto TTLTRG0
	3 to 4	Sampling Rate Clock onto TTLTRG1
	5 to 6	Sampling Rate Clock onto TTLTRG2
	7 to 8	Sampling Rate Clock onto TTLTRG3
JP20	1 to 2	GATE output pulse onto TTLTRG4
	3 to 4	GATE output pulse onto TTLTRG5
	5 to 6	GATE output pulse onto TTLTRG6
	7 to 8	GATE output pulse onto TTLTRG7

* Factory Selection

† For hardware revision 1 and up only. See Appendix D for revision 0 hardware jumpers.

2.2.4 *Deglitching Filter Selection**

Each channel of the dual D/A Converter (DAC) is fed to a deglitching filter network where the filter can be connected in line with the DAC output or bypassed (Table 2-5). Bypassing the filter provides improved DC performance. For more information, refer to Section 3.3.9

Table 2-5. Deglitching Filter Jumper Selections

Selection	Channel 1	Channel 2	Application
In Line	JP21-2 to JP21-3	JP22-2 to JP22-3	Sine, Triangle
Bypassed	JP21-1 to JP21-2	JP22-1 to JP22-2	Square, Pulse, Step Functions, DC

2.3 *INSTALLATION*

The 8750 can be installed in any slot except slot 0. Any module used in synchronous operation with the 8750 via the LOCALbus on connector P2 must be installed in the adjacent slot to the right of the 8750.

2.3.1 *Input/Output Control Connections*

The front panel controls and indicators are described in detail in Section 3. However, for installation purposes, the characteristics of the input/output connections are provided here.

CLOCK External clock input (BNC, 50Ω) – accepts a TTL, negative-going pulse (120 ns, min.), maximum frequency of 400 kHz.

TRIG External trigger input (BNC, 50Ω) – accepts a TTL, active-low signal to allow waveform generation.

GATE Gate output (BNC) – a TTL, active-low output during waveform generation, synchronized with the Internal Sampling Rate Clock.

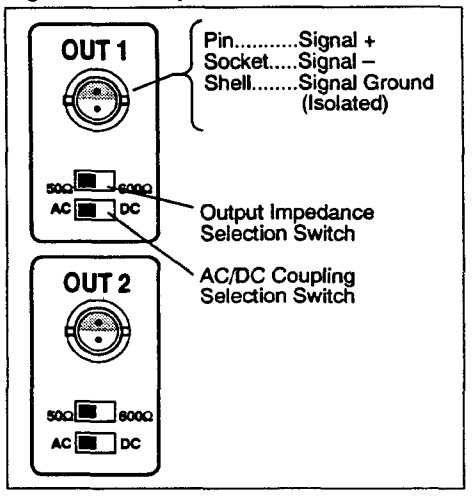
2.3.2 *Analog Outputs*

The analog outputs OUT 1 and OUT 2 (Figure 2-2) are provided on TWBNC connectors. The outputs are switch selectable for 50 or 600 ohms output impedance and AC/DC coupling. The output stage is software selectable for a single-ended or differential configuration.

CAUTION: The output driver current should not exceed 40 mA, max.

* For hardware revision 1 and up only.

Figure 2-3. Output Channel Connectors 1 & 2



2.4 START-UP AND VERIFICATION

2.4.1 Power-Up Status

At power-on, a self-test is executed. The Pass/Fail indicator is red during the test (4.8 seconds) and switches to green when the test passes (see *TST? command in Section 4). Both output channels assume the default parameters listed in Table 2-6 following the self-test.

Table 2-6. OUT 1 & 2 Default Power-On State

Function	State
Output Filters	Bypassed
Attenuation	0.00 dB
Output Drive	Differential
On/Off State	OFF
Waveshape	DC
Frequency	1000.00 Hz
Voltage	0.00 V
Offset Voltage	0.00 V
Memory Buffers	None defined
Oscillator Source	Internal
Trigger Source	Internal
Trigger Sequence Count	0
Trigger Loop Count	0

2.4.2 *Initial Checkout*

The following procedure describes tests that can be performed to verify that the instrument is in good operating condition.

NOTE: For proper operation, the Data Low Register must be read following every command sent to this instrument except for the command, *RST. Normal response is the new line character (x0A) or a status message.

	Action	Response
(1)	Send *RST.	System reset.
(2)	Send *IDN?	Module identification query.
(3)	Read Data Low Register	"Analogic, DBS8750, Rev n, Firmware Revision"
(4)	Send DIAG:ALL.	All diagnostics run. PASS/FAIL is red, switches to green after all tests pass.
(5)	Read Data Low Register	x0A
(6)	Send DIAG?	Diagnostic status query.
(7)	Read Data Low Register	"No errors, self test PASSED"
(8)	*Connect an oscilloscope* probe to OUT1.	
(9)	Send SOUR1:FUNC SIN.	Selects sine function.
(10)	Read Data Low Register.	x0A
(11)	Send SOUR1:VOLT 5	Selects 5-volt amplitude.
(12)	Read Data Low Register.	x0A
(13)	Send OUTP1:DRIV DIFF.	Selects differential output drive.
(14)	Read Data Low Register.	x0A
(15)	Send OUTP1:STAT 1.	1-kHz, 5-volt sine wave on OUT 1.
(16)	Read Data Low Register.	x0A
(17)	Repeat steps (8) through (16) for OUT2.	1-kHz, 5-volt sine wave on OUT 2.
(18)	Send *RST.	System reset. Ready for operation.

Section 3
Operation

3.1 OVERVIEW

This section describes how to operate the DBS 8750 using the DBS 8750 Command Set (Figure 3-1). This command set is used to control the generation of standard and arbitrary waveforms, data transfers to and from memory, triggering, clock selection and diagnostic tests. See Section 4, the DBS 8750 Command Reference, for more information.

NOTES: When programming this instrument it is necessary to read the Data Low Register after sending every command except Reset, *RST. The normal response is the new line character (x0A) or a status message.

The DAC Waveform Memories are not accessible from the VXIbus. The shared DSP/User Memory has 17150 words of user-acccessible read/write space for data transfers to/from VXI memory or from the LOCALbus via connector P2.

If this instrument is to be controlled using a software driver, refer to the Software User's Manual received with the driver for software installation and operating instructions.

Error messages are generated for operational errors that are related to command entry, execution, triggering, data entry, or memory usage. See Appendix C for a list of possible error messages.

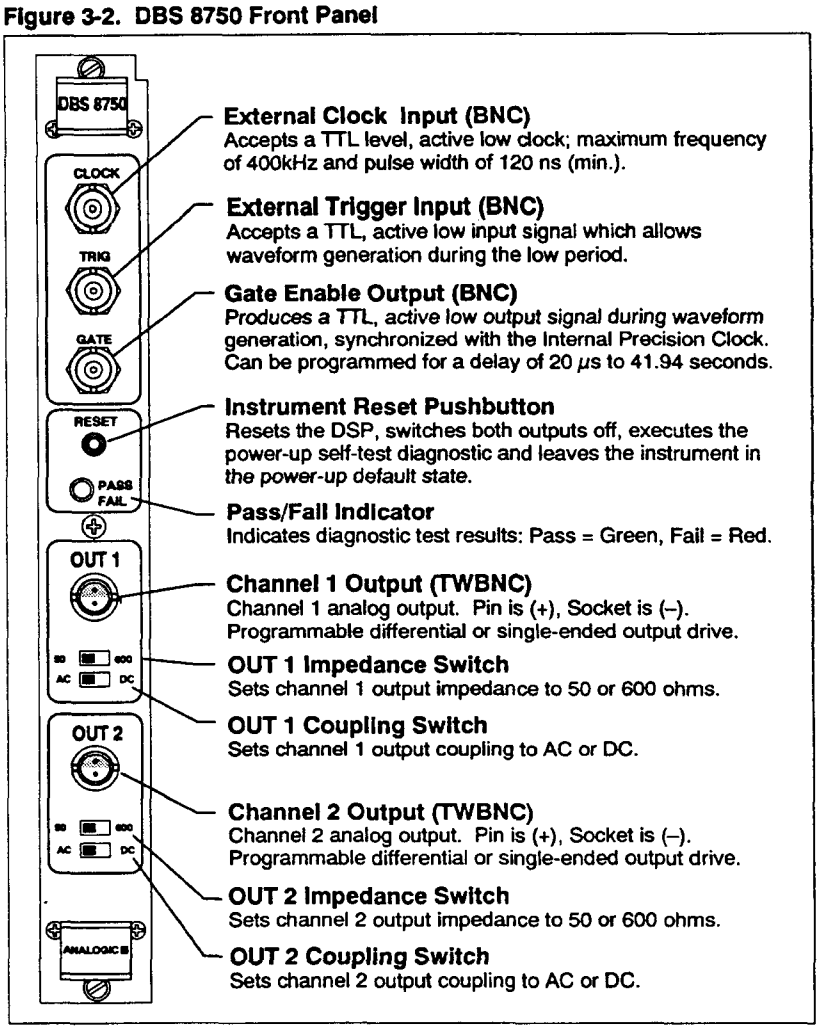
3.2 CONTROLS AND INDICATORS

Figure 3-2 describes the control inputs and indicators on the front panel. After making jumper selections and installing the instrument, all inputs and outputs are operated by software commands. Note that input coupling and input impedance are selected using front panel switches rather than using software commands.

Figure 3-1. DBS 8750 Command Set Quick Reference

```
CALCulate Subsystem ..... CALCulate
                          :MATH <Segment 1> <Segment 2> <Segment 3> ..... <Segment 16>

DIAGnostic Subsystem ..... DIAGnostic
                          :EPRom
                          :SRAM
                          :ANALog
                          :DRAM
                          :REGS
                          :ALL

MEMory Subsystem ..... MEMory
                          :WRITe<sp><VXI Addr><sp><Npts><sp><Buffer Name>
                          :READ<sp><VXI Addr><sp><Npts><sp><Buffer Name>
                          :SAV1(2)<sp><Buffer Name>
                          :ACQuire<sp><Npts><sp><Buffer Name>
                          :CATalog[:ALL]? | BINary?
                          :DELete<sp><Buffer Name> | ALL
                          :FREE[:ALL]? | BINary?
                          :MALLocate<sp><Buffer Name><sp><Npts>

OUTPut Subsystem ..... OUTPut[1](2)
                          :ATTenuation<sp><0 to 90>
                          :FILTer<sp>ON | OFF
                          :STATe<sp>ON | OFF
                          :DRIVe<sp>SINGle-ended | DIFFerential
                          :TRACk<sp>ON | OFF
                          :GAT
                             :DELay<sp><0.00002 to 41.945>
                          :SYNC
                          TTLTrg<sp>ON | OFF

SOURce Subsystem ..... SOURce[1](2)
                          :FREQuency<sp><12.5 to 20000>
                          :VOLTage<sp><–10 to +10>
                             :OFFSet<sp><–10 to +10>
                          :FUNCtion<sp>DC | SINe | SQUare | TRIangle | NOISe | USER | MEMory
                          :ROSCillator
                             :SOURce<sp>INT | EXT | TTLTrg<0-7>
                                   :AUTO SHON | SHOF

TRIGger Subsystem ..... TRIGger[1](2)
                          :SEQuence[1](2)
                             :STARt
                             :STOP
                             :COUNt<sp><numeric>
                          :COUNt<sp><numeric>
                          :SOURce<sp>EXT | INT

                         INITiate[1](2)

                         ABORt[1](2)
```

Figure 3-2. DBS 8750 Front Panel



3.3 GENERATING STANDARD WAVEFORMS

The commands used to generate standard waveforms are found in the SOURce and OUTput subsystems of the DBS 8750 Command Tree. Although the commands that determine waveform characteristics may be written in any order, it is recommended that all characteristics be setup before changing the state of the output to ON.

NOTE: The filter should be ON when generating waveforms. If it is OFF and the internal clock is used to generate square or triangle waves, the acurracy is limited due to spikes introduced from the Track and Hold control circuit (Section 3.3.8).

3.3.1 Selecting the Waveshape

Standard waveshapes are set up by sending the following command:

SOURce[1](2):FUNCtion DC | SINe | SQUare | TRIangle | NOISe

DC ..... DC output level
SINe ..... Sine wave
SQUare ..... Square wave
TRIangle ..... Triangle wave
NOISe ..... Random noise

After all other parameters are set up (frequency, amplitude, DC offset, etc.), the selected output function appears at the output connector when the output channel is switched on by sending OUTPut[1](2):STATe ON.

3.3.2 Selecting the Frequency

The output signal frequency is set up by sending the following command (does not apply to DC and NOISe functions):

SOURce[1](2):FREQuency n,

where n = 12.5 to 20000 Hz for the Sine function,
12.5 to 30000 Hz for the Square and Triangle functions.

3.3.3 Selecting the Amplitude

The output amplitude is set up by sending the following command:

SOURce[1](2):VOLTage n,

where n = +10.000 to –10.000 volts for a single-ended output drive;
+5.000 to –5.000 volts for a differential output drive.

The numeric value "n" represents peak voltage. To prevent clipping, amplitude plus offset should not exceed full scale.

3.3.4 Selecting the DC Offset

The dc output offset is set up by sending the following command:

SOURce[1](2):VOLTage:OFFset n

where n = –10.000 to + 10.000 volts

3.3.5 *Selecting the Drive Mode*

Each channel can be configured for single-ended or differential output drive mode. Figure 3-3 shows a simplified schematic of the output stage. The output coupling and impedance switches are located on the front panel. As you can see, the output drive can be configured for a single-ended or differential output drive depending upon the state of the relay switches. To program the output drive, send the following command:

OUTPut[1](2):DRIVe SINGle-ended | DIFFerential

SINGle-ended .... Single-ended output drive

DIFFerential ..... Differential output drive

NOTE: The Single-Ended output drive has a range of 20 (±10) volts. The Differential output drive has a range of 10 (±5) volts. Maximum output current should not exceed 40 mA.

As shown in Figure 3-4, the actual programmed peak voltage amplitude and voltage offset value are produced at the output connector under open circuit conditions only. When a load is connected, the programmed voltage values are reduced. The amount of reduction depends upon the output impedance of the DBS 8750 and the input impedance of the load. As you can see, applying a load to the output creates a voltage divider with the series output resistance (50 or 600 ohms) of the instrument. The actual voltage level measured at the output depends upon the ratio of the input and output impedances.

Figure 3-3. Simplified Output Stage Schematic

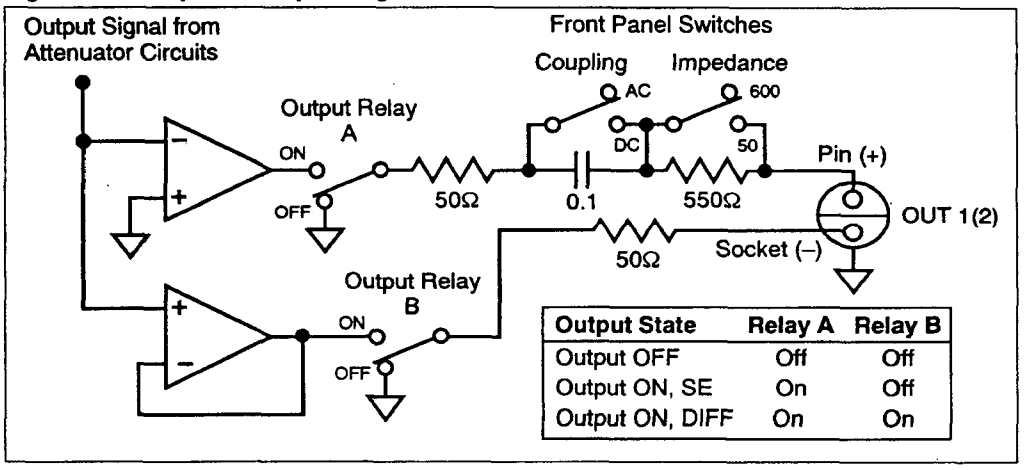
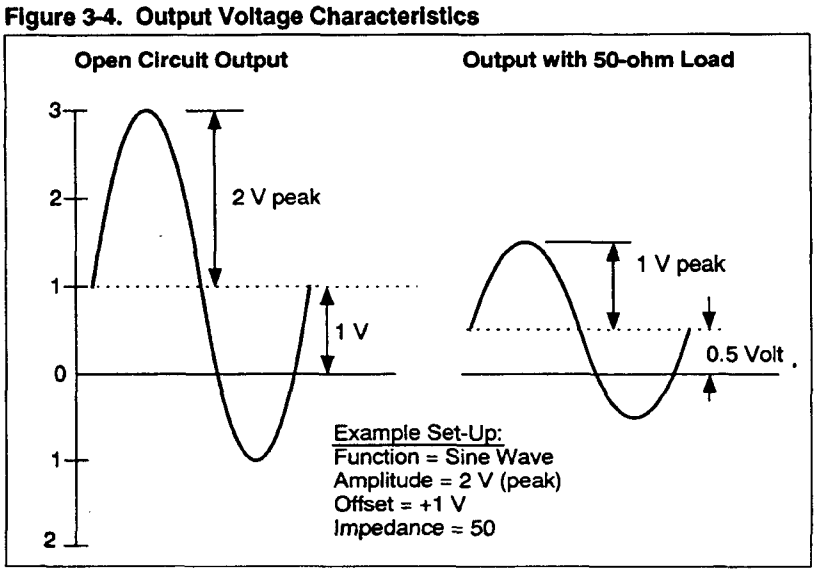


Figure 3-4. Output Voltage Characteristics



3.3.6 *Selecting Attenuation*

The output signal can be attenuated from 0 to 90dB by sending,

OUTPut[1](2):ATTenuation n, where n = 0 to 90

Output attenuation is accomplished by a combination of a switched resistor network and voltage reduction performed by mathematical calculation. The resistor network provides attenuation steps of 0dB, 10dB, 20dB and 30dB. Other settings are completed by mathematical calculations which lower the output voltage an additional amount.

3.3.7 *Selecting the Filter*

An output reconstruction filter is available for low-distortion sine wave generation. The filter is connected in line or bypassed by sending the following command:

OUTPut[1](2):FILTer ON or 1 | OFF or 0

ON or 1 ..... to connect the filter

OFF or 0 ..... to bypass the filter

Normally, the filter should be switched ON. However, the instrument may be used with the filter switched OFF for certain step functions. Always switch the filter ON to generate low distortion SINe waves. See also Section 3.3.8 for use of Track-and-Hold with SINe waves.

3.3.8 Using Track-and-Hold

3.3.8.1 Tracking Mode

The Tracking Mode is selected by using the following command:

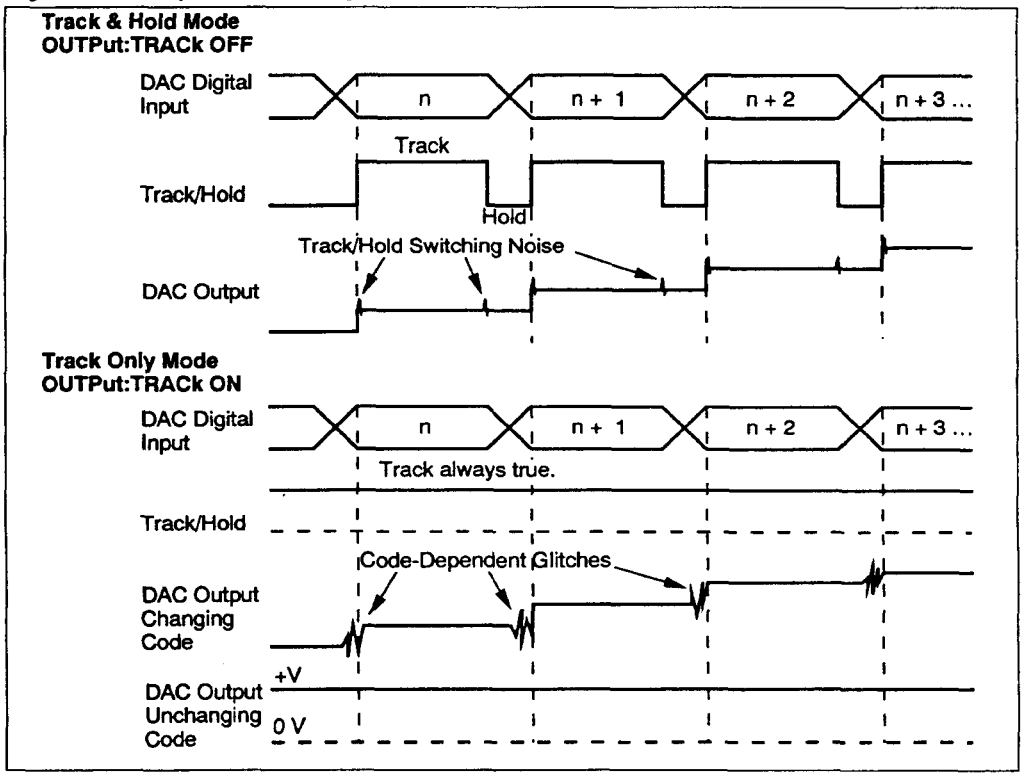
OUTput[1](2):TRACk ON or 1 | OFF or 0

ON or 1 ..... Track Only

OFF or 0 ..... Track and Hold

Figure 3-5 shows the affects of the Tracking Modes on the Digital-to-Analog (DAC) output signal. When generating sine waves, always switch to the Track-and-Hold Mode which reduces the code-dependent glitch energy on the DAC output. For DC levels, switch to the Track Only Mode to prevent Track/Hold switching noise on the DAC output. The Track Only Mode is also useful for reducing square wave settling times. However, glitch energy at major DAC code transitions will be higher.

Figure 3-5. Output vs. Tracking Mode



3.3.8.2 Track-and-Hold Synchronization

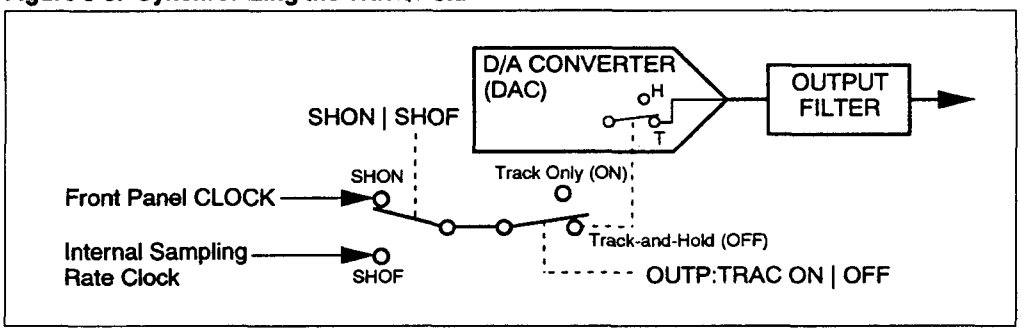
The Track-and-Hold Mode can be synchronized with either the 400-kHz Internal Sampling Rate Clock or an External Sampling Rate Clock via the front panel CLOCK input (Figure 3-6). To synchronize the Track-and-Hold, use the following command:

SOURce:ROSCillator:SOURce:AUTO SHON | SHOF

SHON ... Synchronize Track-and-Hold with Front Panel CLOCK input.

SHOF .... Synchronize Track-and-Hold with Internal Sampling Rate Clock (Default).

Figure 3-6. Synchronizing the Track/Hold

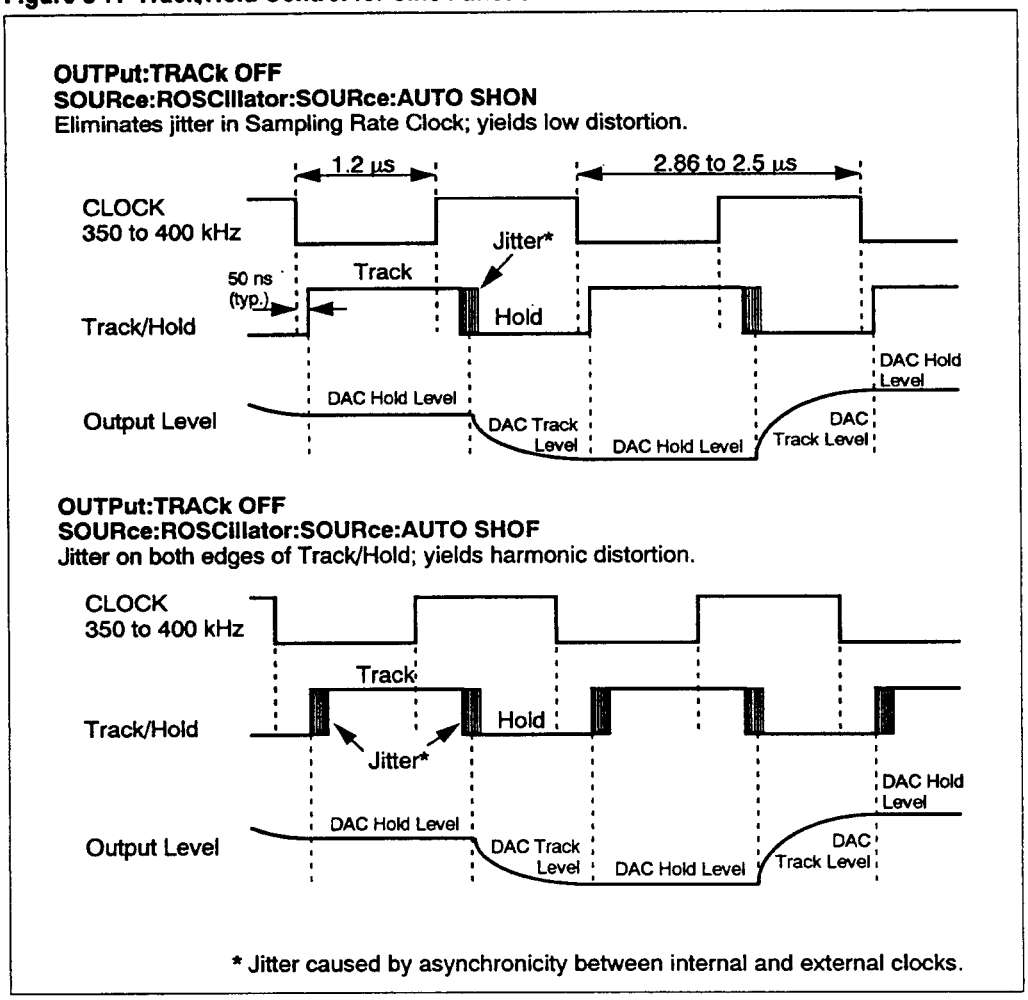


3.3.8.3 Controlling Track-and-Hold with a Front Panel CLOCK

An External Sampling Rate Clock connected to the Front Panel CLOCK input (not the TTLTRG lines) can be used to control the DAC Track-and-Hold Output Amplifiers (Figure 3-7). This feature is especially useful for generating low-distortion Sine waves. For this purpose, the CLOCK input should be a 350 to 400-kHz, negative-going pulse with a pulse width of 1.2us ±50ns, and less than 100 pS of jitter. Refer to Section 3.7 for information on selecting clock sources.

Figure 3-7 shows the affect of clock synchronization on the Track-and-Hold control. In the top half of the diagram, Track-and-Hold is synchronized with the External Sampling Rate Clock from the Front Panel. The bottom half shows synchronization with the Internal Sampling Rate Clock. This diagram clearly shows that low-distortion sine waves can be generated while Track-and-Hold is synchronized with an External Sampling Rate Clock because there is less jitter on the Track-and-Hold control signal.

Figure 3-7. Track/Hold Control for Sine Function



* Jitter caused by asynchronicity between internal and external clocks.

3.3.9 *Using the Deglitching Filter*

By jumper selection, the output signal of each DAC may be passed through a deglitching filter (see Section 2.2.4) to reduce the affect of noise and glitch energy on the output. Bypassing the deglitching filter* provides improved DC output performance and a reduction in square wave settling time.

* For hardware revision 1 and up only.

3.3.10 Switching the Output Channel State to On/Off

As shown previously in Figure 3-3, each of the two analog output channels can be connected or disconnected independently with a relay switch. These relays are switched on/off by sending the following command:

OUTPut[1](2):STATe ON or 1 | OFF or 0

ON or 1 ..... Connects the output

OFF or 0 ..... Disconnects the output

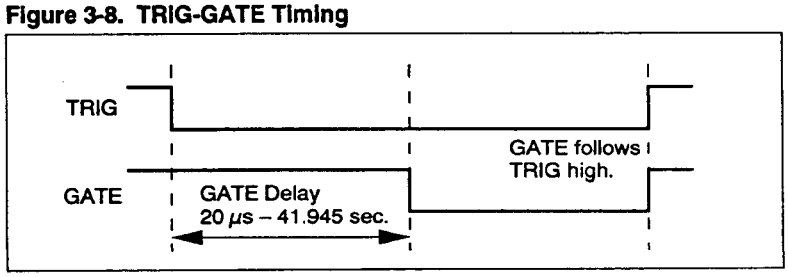
The ON state closes a relay to connect the output amplifier to the front panel OUT 1(2) connector. The OFF state opens the relay and switches the connector output pin to ground through 50 or 600 ohms. During waveform calculations and initialization, you may wish to keep the outputs OFF, and only switch them ON after all other parameters have been specified.

3.3.11 Selecting the GATE Delay

The front panel GATE output produces a TTL, active-low output signal, synchronized with the Sampling Rate Clock, during waveform generation. The GATE output can be programmed for an output delay of 20 µs to 41.945 seconds in 2.5-µs increments (Figure 3-8). This output can be used as an external trigger or start signal for another VXI instrument such as a DBS 8700 Digitizer. By jumper selection (Section 2.2.3), this output can be made available on TTL trigger line 4, 5, 6, or 7 (Section 3.3.13)*. To program the GATE delay, send the following command:

OUTput:GAT:DELay n, where n = 0.00002 to 41.945 secs in 2.5-µs steps

Figure 3-8. TRIG-GATE Timing



* For hardware revision 1 and up only. See Appendix D for revision 0 hardware jumpers.

3.3.12 Synchronizing the Output Channels

The channel 1 and 2 output waveforms can be synchronized with each other, regardless of their current phase relationship, by sending the OUTPut:SYNC command (the [1](2) designation not required). This simultaneously resets the DAC memory address generator for both channels and starts generating the waveforms from their starting addresses. The waveforms stay synchronized providing they have a harmonic relationship from the start.

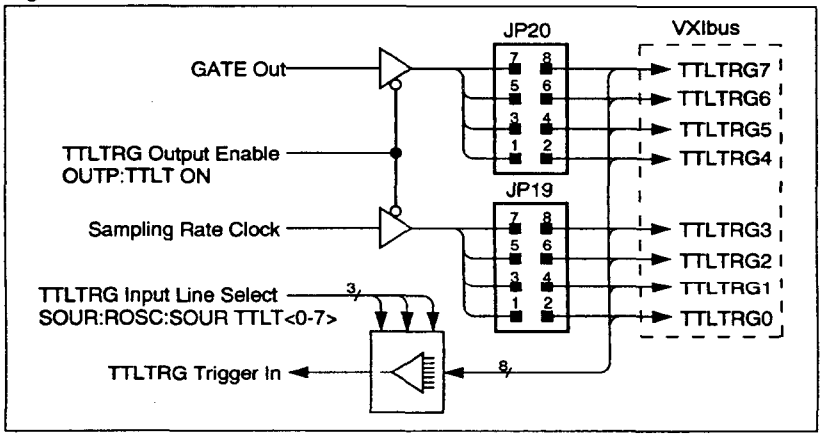
3.3.13 Using the TTL Trigger Lines

The TTL Trigger Lines can be used to synchronize 8750 operation with another instrument. Before using any particular line, a hardware jumper must be installed (Figure 3-9).

An External Sampling Rate Clock that is being driven onto the bus from another source can be received on any TTLTRG line. The input line is selected by the command SOURce:ROSCillator:SOURce TTLTRG<0-7>.

The 8750 can drive two signals onto the VXIbus: the Sampling Rate Clock and the GATE output pulse, the same signal generated from the GATE connector on the front panel. The Sampling Rate Clock can be driven onto TTLTRG 0, 1, 2 or 3. The GATE output pulse can be driven onto TTLTRG 4, 5, 6 or 7. Once the jumpers have been correctly installed, the TTLTRG signals are enabled by sending the OUTPut:TTLTrg ON command. Refer to Section 2.2.3 for jumper selection information.

Figure 3-9. TTLTRG I/O Functions*



* For hardware revision 1 and up only. See Appendix D for revision 0 hardware jumpers.

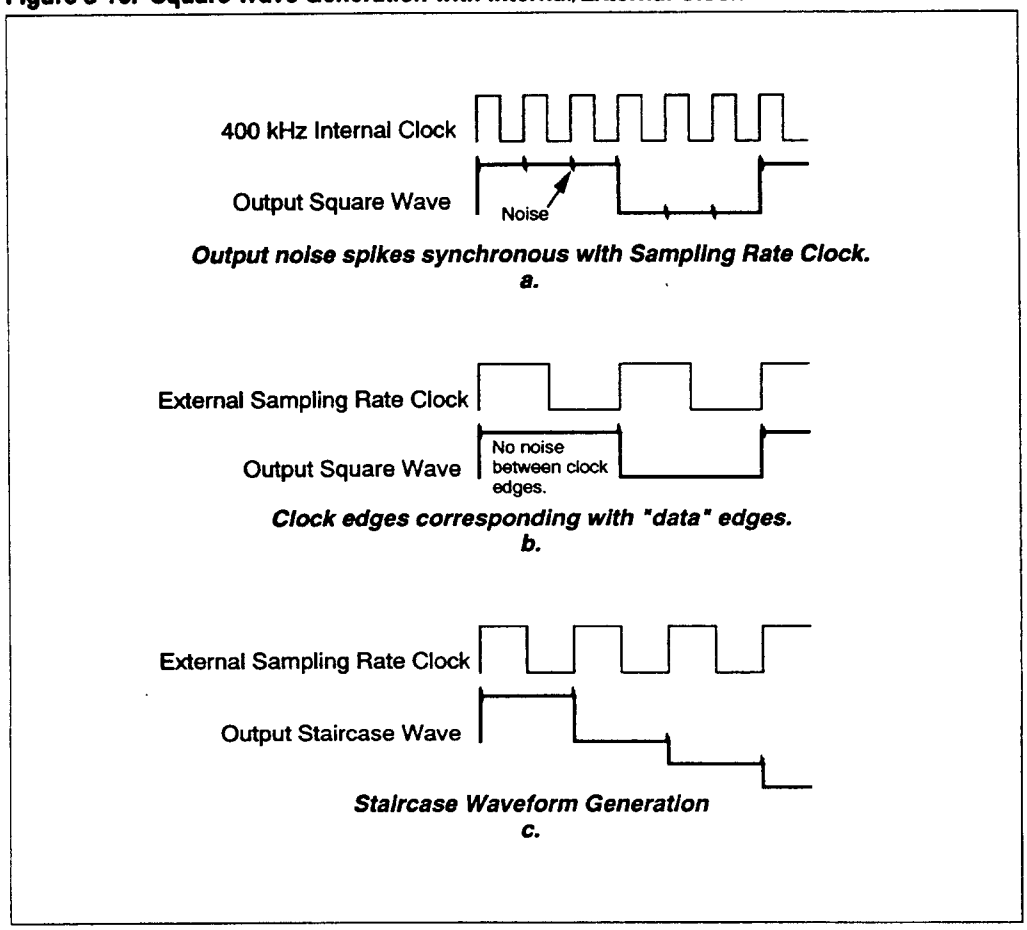
3.3.14 *Generating Clean Square/Staircase Waves*

When generating standard square waves using the 400-kHz Internal Sampling Rate Clock, clock-related noise is evident on the output wave whenever the data point values stay the same across multiple clock edges (Figure 3-10a). This happens because the internal Track-and-Hold circuit of the DAC is switched at every rising clock edge.

One solution to this problem is to switch the Track-and-Hold mode to "Track Only" but this creates a different set of problems. The recommended technique is to use an External Sampling Rate Clock and create the square wave so that each different data point value corresponds with a single rising clock edge (Figure 3-10b). The secret to this technique is to remember that the 8750 always calculates waveforms based on a 400 kHz sampling rate. If you want a square wave to change state on every rising clock edge, you must select a frequency of 200 kHz. This forces the 8750 to create a data pattern consisting of exactly two data points. Now, if a 20 kHz external clock is applied, for example, the output waveform is a 10-kHz square wave (half the external sampling rate) with clock-related noise spikes appearing only at the rising edges of the external clock.

This same technique applies to other DC-accurate applications such as for generating staircase (Figure 3-10c) or pulse waveforms.

Figure 3-10. Square Wave Generation with Internal/External Clock



3.4 GENERATING ARBITRARY WAVEFORMS

Arbitrary waveforms can be generated by either of the two commands SOURce[1](2):FUNCtion USER or SOURce[1](2):FUNCtion MEMory.

3.4.1 SOURce:FUNCtion USER Waveforms

The USER function generates waveforms which have been calculated by the CALCulate[1](2):MATH command. USER-defined waveforms may consist of up to 16 user-defined segments, but the total waveform length must be no more than 32K-words (Refer to Section 3.5 on memory and waveform size).

The format of the CALCulate[1](2):MATH command is as follows:

CALCulate[1](2):MATH<sp><segment 1><sp> ..... <sp><segment 16>

where <segment n> = FOR<sp><duration><sp><voltage>,

a mathematical expression of the waveform segment where

FOR ..... Is a directive which begins the mathematical definition.

<duration> ..... Specifies the period of time (in seconds) during which the voltage function takes place. Do not use commas or suffixes such as "m" (milli), "μ" (micro), "k" (kilo),or "M" (Mega) – these are not recognized by the firmware.

<voltage> ..... Specifies the output voltage with respect to time. It can be expressed as a simple decimal value or a complex mathematical expression using any of the mnemonics and symbols described in Table 3-1.

In general, when writing math expressions, use parentheses to group quantities properly and note that expressions are always evaluated from right to left. Do not exceed 16 FOR statements or 859 characters per command.

Table 3-1. Mathematical Mnemonics and Symbols

Mnemonic or Symbol	Function or Meaning	Mnemonic or Symbol	Function or Meaning
+	Positive value; addition	COS	Cosine
–	Negative value; subtraction	TAN	Tangent
*	Multiply	ASIN	Arc sine
/	Divide	ACOS	Arc cosine
^	Raised to the power	ATAN	Arc tangent
( )	Parentheses	LN	Natural log (base e)
e	2.7182818	LOG	Common log (base 10)
PI	3.1415927	ABS	Absolute value
SIN	Sine	t	Time variable (t=0 at start)

Following a power-on cycle or a system reset, the last USER function calculated is lost making the USER function undefined.

To generate a USER waveform, first define the waveform using the CALCulate[1](2):MATH command and then send the command, SOURce[1](2):FUNCtion USER which transfers the user-defined waveform into Waveform Memory. To change the USER function, send another CALCulate[1](2):MATH command to replace the previously calculated waveform. Example waveforms are shown in Figures 3-11 through 3-15. These examples were performed with a single-ended output, internal clock, no filter and no attenuation.

NOTE: Because of the methods used to calculate waveforms, the actual output may not appear exactly as illustrated.

Figure 3-11. Complex Waveform By Segments

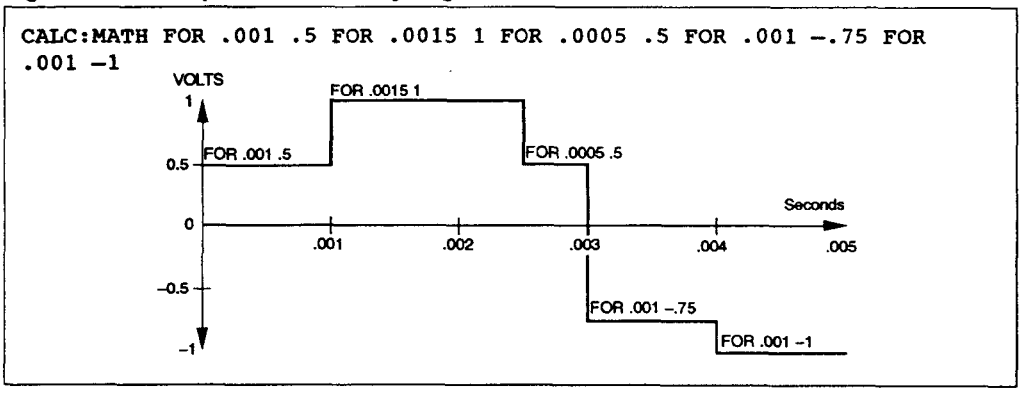


Figure 3-12. 1-kHz Sine Wave Output

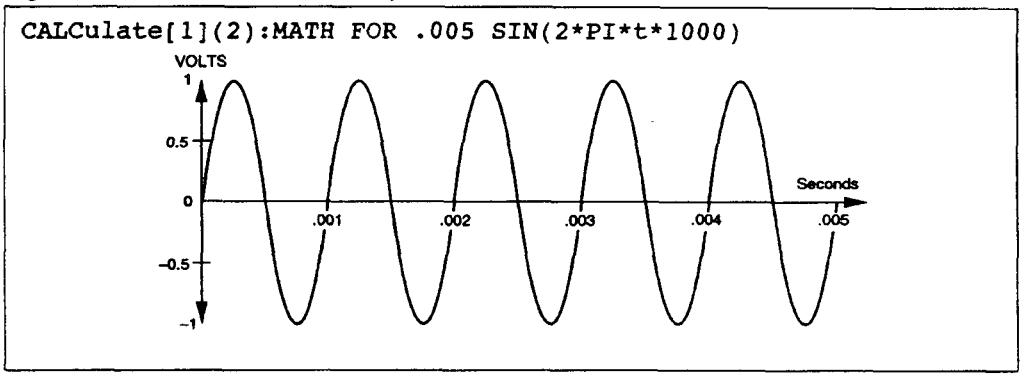


Figure 3-13. Log Function Output

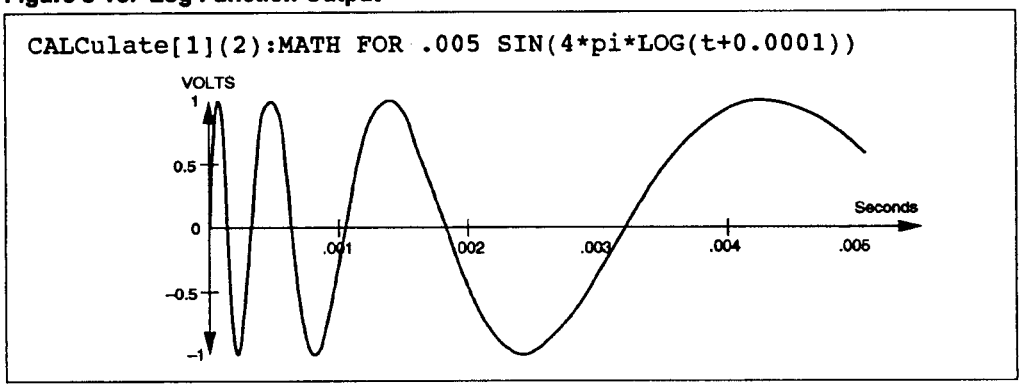


Figure 3-14. Exponential Output Waveform

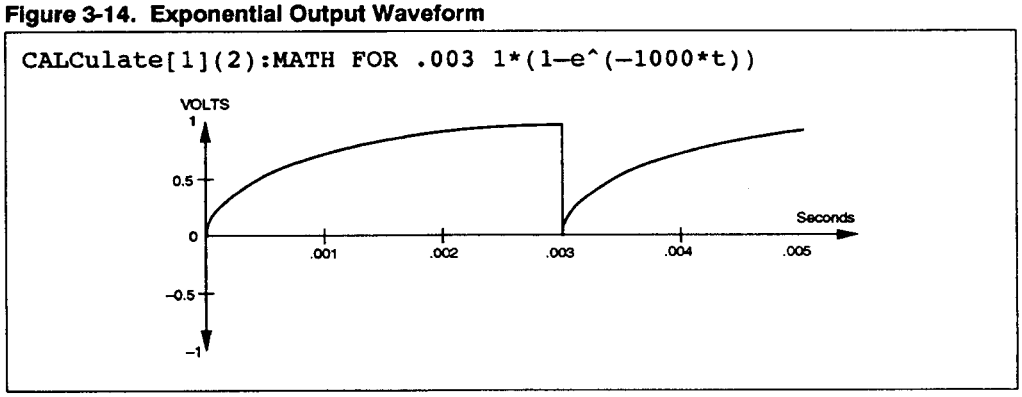
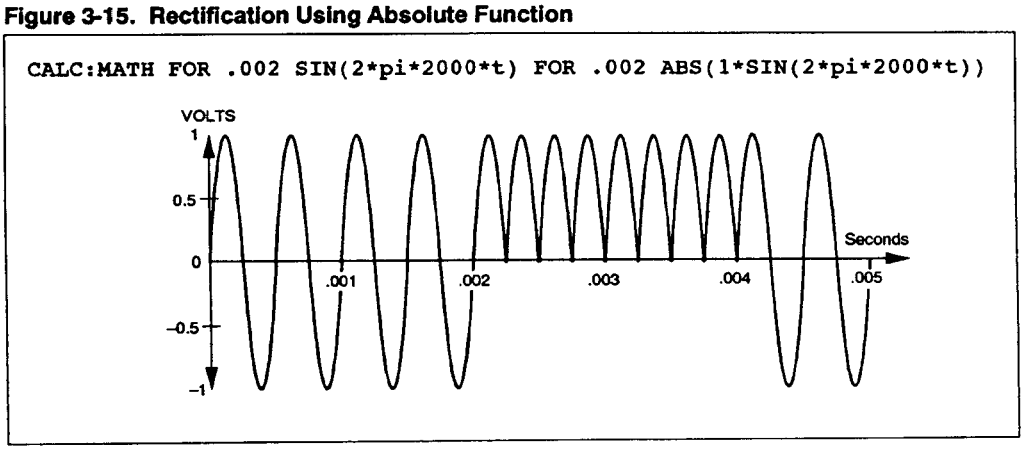


Figure 3-15. Rectification Using Absolute Function



3.4.2 SOURce:FUNCtion MEMory Waveforms

A MEMory waveform is defined by the contents of a memory buffer which was previously created in the 8750 DSP/User Memory. A memory buffer is created whenever waveform data is transferred to the DSP/User Memory from VXI memory or an adjacent Digitizer module via the LOCALbus. To select a MEMory waveform, send

SOURce[1](2):FUNCtion MEMory <Buffer Name>

This transfers the data in <Buffer Name> from DSP/User Memory into DAC Waveform Memory from which the DAC generates the waveform.

If you wish to switch between several user-defined output waveforms, it is best to place them in individual memory buffers and then select the waveforms using the SOURce:FUNCtion MEMory command. This is faster than the USER function because each time you select USER, the DBS 8750 requires time to re-evaluate the CALC statement before placing the results in the DAC Waveform Memory. The DSP/User Memory provides user-accessible read/write space for 17,150 words (See Section 3.5.).

3.5 DSP/USER MEMORY

The DSP/User Memory has a total of 17,150 words of user space available for saving a waveform currently residing in DAC Waveform Memory, or for transferring data from VXI memory or an external source, such as an adjacent Digitizer module, via the LOCALbus.

NOTE: The DSP/User Memory is RAM. All data is lost when power is removed from the DBS 8750. Be sure to save all important data to another memory device before switching power off. After power-on, this memory contains random data. This memory is not initialized during the power-on sequence.

3.5.1 Creating Memory Buffers

Memory buffers are created in the DSP/User Memory whenever any of the following commands are used: MEMory:SAV1(2), MEMory:READ, MEMory:ACQuire, and MEMory:MALLocate.

The creation of each new memory buffer requires 35 words of space for identification and housekeeping. Therefore, when estimating memory space requirements, bear in mind that the size of each buffer is determined by the number of points (words) comprising the waveform plus 35. This means that if three 1000-point waveforms are saved, the amount of actual memory space occupied is 3,105 points [(3 x 1000) + (3 x 35) = 3105].

A maximum of 20 memory buffers may be created. Any attempt to create more than that generates an error message. However, the number of buffers that can be created also depends upon the number of data points (words) being written into each buffer. The number of data points (Npts) in a waveform is determined by dividing the Sampling Rate Clock Frequency by the frequency of the output waveform [Equation 1]. For example, using the 400-kHz Internal Sampling Rate Clock, a 400-Hz sine wave consists of 1,000 points [Equation 2]. At the same rate, a 20 kHz sinewave consists of only 20 points, but a 23.5 Hz sinewave consists of 17021 pts which nearly fills the entire user memory space by itself.

$$\frac{\text{Sampling Rate Clock Frequency}}{\text{Output Function Frequency}} = \text{Npts} \qquad \text{[Equation 1]}$$

$$\frac{400\ \text{kHz}}{400\ \text{Hz}} = 1000\ \text{pts} \qquad \text{[Equation 2]}$$

NOTE: A DC function always generates 100 points and the NOISe function always generates 8192 points.

The user memory space is written contiguously, starting at the Memory Base Address, as each buffer is created. That is, the memory is filled with one buffer after another with no gaps in between. The scenario depicted in Figure 3-16 demonstrates how this memory might be used. In this diagram, the numbered arrows indicate the direction of data transfer; the numbers indicate the sequence in which the commands were performed. First, the user generated (1) a standard 400-kHz (1000 pts.) sine wave on channel 1 and saved it (2) to a buffer named "SavdWave." The user then transferred that buffer (3) to the VXI Memory. Next, 1024 data points were read (4) from the VXI Memory and stored in the buffer named "VxBuffer." After that, the user acquired 2048 data points (5) from a DAS module via the LOCALbus and generated that waveform in the DAC 2 channel (6). Finally, a 2048-word space was allocated (7) in the memory for "NewData."

3.5.2 Saving Waveforms

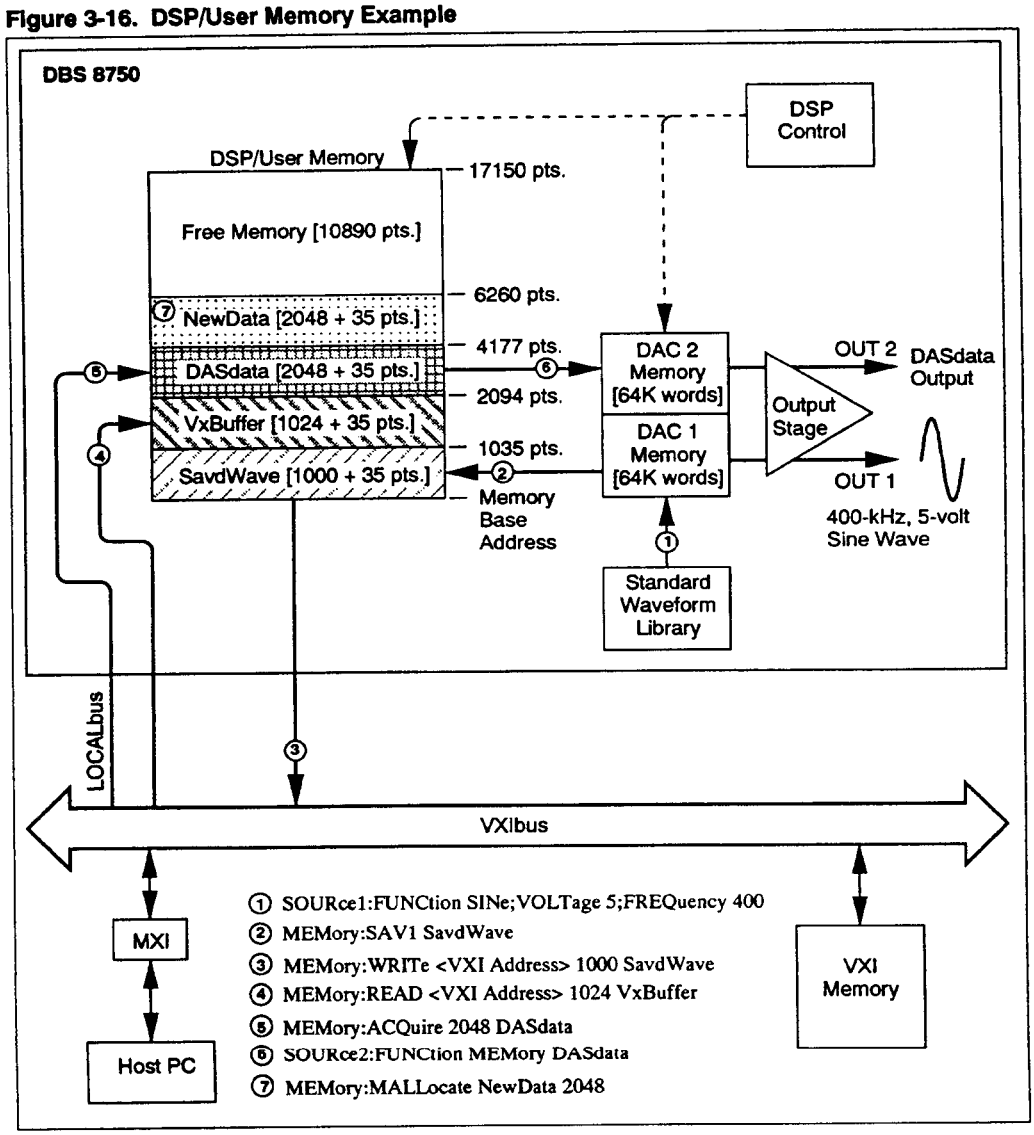
To save a particular waveform, first select it by the command,

SOURce:FUNCtion DC|SINe|SQUare|TRIangle|NOISe|USER|MEMory

This creates the waveform data in the DAC 1/2 Waveform Memory. The waveform data can then be saved using the command,

MEMory:SAV1(2) <Buffer Name>

Figure 3-16. DSP/User Memory Example



3.5.3 *Read/Write VXI Memory*

NOTE: The MEMory:READ and MEMory:ACQuire functions automatically allocate the named buffer if it does not already exist.

The DBS 8750 can read/write VXI Memory data using the MEMory:READ and MEMory:WRITe commands (Refer to Figure 3-14.). A READ transfers data from VXI memory to DSP/User memory. A WRITe transfers data from the DSP/User Memory to VXI Memory.

For example, to read 1024 points of VXI memory data, starting at VXI address 200000h, into a DSP/User Memory buffer named "VDATA," send

MEMory:READ 200000 1024 VDATA

To WRITe 1024 points of DSP/User memory buffer "VDATA" into VXI Memory starting at VXI address 200000h, send

MEMory:WRITe 200000 1024 VDATA

The DBS 8750 is a Bus Master during data transfers.

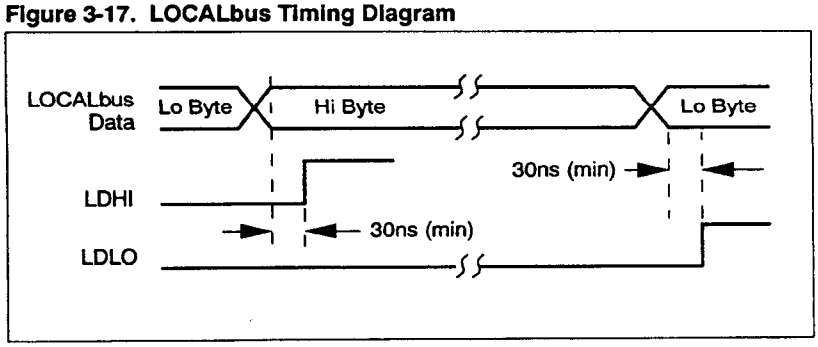
3.5.4 *Acquiring LOCALbus Data*

Waveform data can be acquired via the LOCALbus from an external source such as a DBS 8700 Digitizer. The strobe signals and timing requirements are shown in Figure 3-17. The 16-bit word is received in two bytes using two individual strobe signals: LDHI (P2c17) is the high byte strobe signal (loaded first), and LDLO (P2c18) is the low byte strobe signal.

To acquire 2048 points of waveform data via the LOCALbus and store it in a buffer named DIGDATA, send

MEMory:ACQuire 2048 DIGDATA

Figure 3-17. LOCALbus Timing Diagram



NOTE: This command requires that a complete data transfer occurs in order to terminate successfully. Otherwise, the command will "hang" waiting for the transfer to complete.

The data is stored in the specified memory buffer as 16-bit words. The maximum word rate is 300 kHz. As long as the timing requirements are met, this port can receive data from any custom device without burdening the VXIbus. See Appendix A for signal connection pin assignments.

3.5.5 *Allocating Memory*

The MEMory:MALLocate command can be used to allocate a block of DSP/User memory. This command creates a buffer of a specified point size. After allocation, any waveform data may be written into this space.

To allocate a block of DSP/User Memory, send

MEMory:MALLocate<sp><Buffer Name><sp><Npts>

<Buffer Name> .. Name of buffer, 31 characters max., excluding whitespace, colons, and semicolons.

<Npts> ..... Number of points – A positive decimal integer for the number of 16-bit words to be written to memory (maximum is amount of free memory minus 35 words overhead). No default value. Failure to specify causes a syntax error.

For example, sending MEMory:MALLocate NewWave 1000 would create a buffer named "NewWave" starting at the next free location in DSP/User Memory and allocate 1000 + 35 points of memory space for this buffer.

In response to this command, the DBS 8750 returns either the A24 base address offset (in ASCII) of the buffer, or the error message; "-225, Out of memory; cannot allocate memory for buffer."

When specifying buffer size, keep in mind that the buffer must be loaded into contiguous memory. Thus, if you have been adding and deleting buffers, this command may fail, even if the total amount of free memory is sufficient.

Data may be written to the allocated memory in the A24, D16 space, but since only 16-bit transfers are allowed, only even addresses are valid.

This command can also be used to re-allocate an existing buffer. However, if a sufficient amount of contiguous memory space is not available, the command may fail. Note that the DBS 8750 frees the existing buffer memory before attempting to reallocate the buffer. If resizing fails, the

buffer re-allocates memory to the original size and data in the existing buffer may be corrupted. If failure occurs, the error message "-225, Out of memory; cannot resize the buffer" is generated.

NOTE: Buffers which are allocated by this command are not initialized by the DBS 8750.

The DBS 8750 assumes that the whole allocated buffer area is used for the waveform.

The DBS 8750 does not provide any bounds checking while writing to a user buffer. Writing to memory which has not been allocated, or writing more points than have been allocated may cause unpredictable results, including "hanging" the VXI bus.

3.5.6 Memory Status

A status of the DSP/User Memory can be obtained using the MEMory:CATalog? and MEMory:FREE? query commands.

The MEMory:CATalog? command responds with a continuous data string in the following format:

<filled memory>,<free memory>,<buffer name>,<type>,<size>,
<buffer name>,<type>,<size>,<buffer name>,<type>,<size> ... etc.

If there are no existing buffers, the response to MEM:CAT? is 0,17150.

Conversely, the MEM:FREE? response is <free memory>,<filled memory>; or 17150, 0.

If the memory is filled as shown in Figure 3-14, MEM:CAT? responds with the following which is formatted in a column for the purpose of this presentation. Formatting is left up to the user.

6260,10890,
SAVDWAVE,BIN,1000,
VXBUFFER,BIN,1024,
DASDATA,BIN,2048,
NEWDATA,BIN,2048

And the MEM:FREE? response is,

10890,6260.

3.5.7 Deleting Buffers

Buffers can be deleted from DSP/User Memory all at once or one at a time by name using the MEMory:DELete command.

To delete the first buffer in the Figure 3-14 example, send

MEM:DEL SavdWave

To delete all buffers, send

MEM:DEL ALL

3.6 TRIGGERING

The DBS 8750 can be triggered internally by the software, or externally via the front panel TRIG input or one of the TTL Trigger lines. Use the following command for trigger control:

TRIGger:SOURce INT | EXT | TTLtrg<0-7>

INT ..... Internal Software Command

EXT ..... Front Panel TRIG Input

TTLtrg<0-7> ..... TTL Trigger line 0-7

3.6.1 Internal Trigger

To select internal triggering, send TRIGger:SOURce INT. Since this is the power-on default setting, in a sense the DBS 8750 is always triggerred. With the internal trigger mode, waveform generation begins immediately after it is defined, assuming that the output is switched on.

To use this command effectively,

(1) Send TRIG:SOUR EXT.

(2) Define the waveform.

(3) Send TRIG:SOUR INT to start the waveform.

3.6.2 External Trigger

The TRIG input (via the front panel connector or a TTLTRG line) accepts a TTL, active low signal to enable waveform generation. An uncertainty of 60ns to 120ns may be introduced due to internal synchronization.

By sending TRIGger:SOURce EXT, the TRIG input signal can be used to control waveform generation.

3.6.3 Trigger Sequences

A Trigger Sequence is a set of command functions which are executed under control of the Internal/External Trigger signal and the Trigger Sequence Commands (Table 3-2). Trigger Sequences allow you combine various types of standard and/or user-defined wave shapes to generate complex output waveforms. The defined Trigger Sequence can be generated as a single burst of a defined number of cycles or it can be generated continuously in a Free Running mode.

The following three examples describe an internally-triggerred complex wave burst, an externally-triggerred sine wave burst and a free running complex wave.

NOTE: In order for Trigger Sequences to run correctly, the DBS 8750 must be equipped with firmware revision 2.0 or higher. To determine hardware/software revision levels, refer to the *IDN? command in Section 4.

3.6.3.1 Programming Guidelines

When creating Trigger Sequences, bear in mind the following guidelines:

(1) Only two trigger sequences are allowed per channel. When sending the commands, SEQuence1 must always precede SEQuence2.

(2) The beginning and end of a sequence must be defined by a TRIG[1](2):SEQ[1](2):STARt command and a TRIG[1](2):SEQ[1](2):STOP command, respectively.

(3) Since the defined sequence is created in 32K-word DAC Waveform Memory, the aggregate number of points generated by the trigger sequence(s) must be less than or equal to 32K-points.

(4) As a general pratice, always strive to fill waveform memory with as many waveform cycles as possible (TRIG:SEQ:COUNt <Highest Possible Value>) and set the repeat count as low as possible (TRIG:COUNt <Lowest Possible Value>). This generates the minimum number of interrupts to the DSP and ensures better performance.

(5) Always send the INITiate command last after the complete sequence has been defined.

(6) OUTPut commands may be included in a trigger sequence, but are executed only once following the INITiate command.

(7) When using an External Trigger signal, it must be held low for the entire duration of the Trigger Sequence output including all repeat loops.

(8) To generate a waveform in the free running mode, set the repeat count to zero. In other words, include the command TRIG:COUN 0.

Table 3-2. Trigger Sequence Control Commands

Command	Function
TRIGger[1](2):SEQuence[1](2):STARt	Defines the beginning of a Trigger Sequence. SEQ1 must always precede SEQ2.
TRIGger[1](2):SEQuence[1](2):STOP	Defines the end of a Trigger Sequence.
TRIGger[1](2):SEQuence[1](2):COUNt	Sets the sequence counter which controls the number of times to repeat the commands of the sequence. This statement must be sent between the STARt and STOP commands. This count determines the number of waveform cycles that are to be written in waveform memory. This value should always be as high as possible. However, maximum COUNt is limited by waveform memory size. The total number of points generated by the trigger sequences must be less than or equal to 32K words, the maximum number of points that each DAC Waveform Memory can hold.
TRIGger:COUNt	Sets the repeat counter which controls the number of times to repeat all trigger sequences. Actually, this specifies the number of times the program reads through waveform memory.
INITiate[1](2)	Arms the Trigger Sequence for execution. If this command is sent while in the Internal Trigger Mode, the trigger sequence is executed immediately. If in the External Trigger Mode, the sequence is executed when a positive TTL level is present on the TRIG input.
ABORt[1](2)	Stops execution of the Trigger Sequence. Trigger Sequence information is saved and the instrument is returned to its previous state.

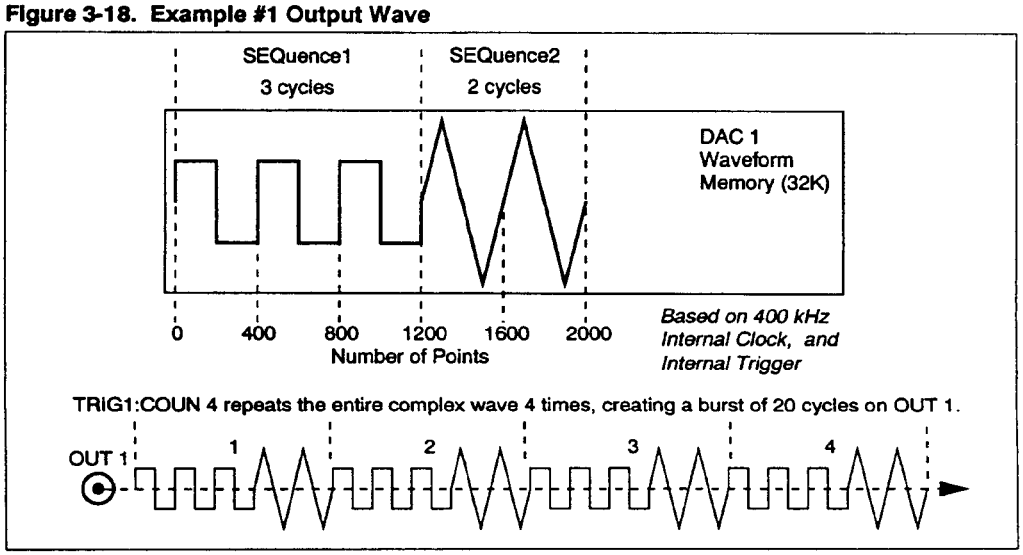
3.6.3.2 *Example #1: Complex Wave Burst – Internal Trigger*

This Trigger Sequence creates three cycles of a 1-kHz square wave followed by two cycles of a 1-kHz triangle wave and repeats the composite waveform three times. The Internal Trigger is selected by a reset default setting.

```
*RST ..... Reset the instrument.
TRIG1:SEQ1:STARt ..... Start of SEQuence1 for OUT 1.
  SOUR1:FUNC SQU ..... Select square wave.
  SOUR1:VOLT 5;FREQ 1000 ..... Select 5-volt amplitude; 1 kHz.
  OUTP1:DRIV SING ..... Select single-ended output.
  OUTP1:FILT OFF;STAT ON ..... Select filter off; output ON.
  TRIG1:SEQ1:COUN 3 ..... Set SEQuence1 Counter to 3.
TRIG1:SEQ1:STOP ..... End of SEQuence1.
TRIG1:SEQ2:STARt ..... Start of SEQuence2 for OUT 1.
  SOUR1:FUNC TRI ..... Select triangle wave.
  SOUR1:VOLT 10;FREQ 1000 ..... Select 10-volt amplitude; 1 kHz.
  TRIG1:SEQ2:COUN 2 ..... Set SEQuence2 Counter to 2.
TRIG1:SEQ1:STOP ..... End of SEQuence2.
TRIG1:COUN 4 ..... Set Repeat Counter to 4.
INIT1 ..... Arm trigger sequence for OUT1.
```

NOTE: Because of the methods used to calculate waveforms, the actual output may not appear exactly as illustrated.

Figure 3-18. Example #1 Output Wave



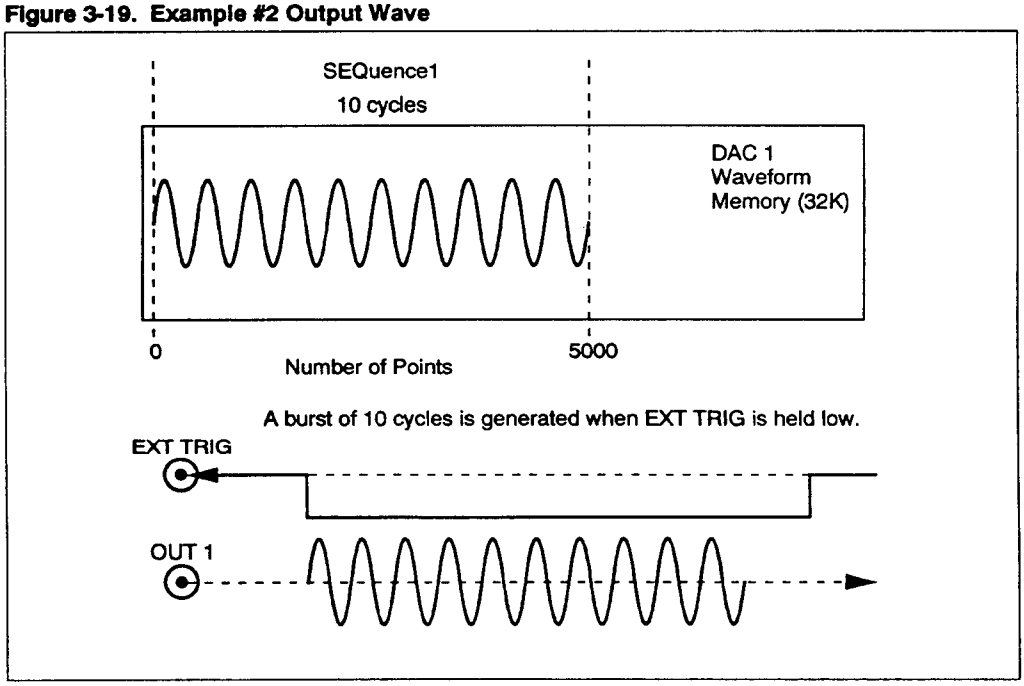
3.6.3.3 Example #2: Sine Wave Burst – External Trigger

The following generates 10 cycles of a 800-Hz Sine Wave, triggerred externally. Note that the EXT TRIG input signal is held high for the entire output burst.

```
*RST ..... Reset the instrument.
TRIG1:SEQ1:STARt ..... Start of SEQuence1 for OUT 1.
  SOUR1:FUNC SIN ..... Select sine wave.
  SOUR1:VOLT 5.75 ..... Select 5.75-volt amplitude.
  SOUR1:FREQ 1000 ..... Select 1 kHz.
  OUTP1:FILT ON;STAT ON ..... Select filter OFF; output ON.
  TRIG1:SEQ1:COUN 10 ..... Set SEQuence1 Counter to 10.
  TRIG:SOUR EXT ..... Select External Trigger from front panel.
TRIG1:SEQ1:STOP ..... End of SEQuence1.
INIT1 ..... Initiate trigger sequence.
```

NOTE: Because of the methods used to calculate waveforms, the actual output may not appear exactly as illustrated.

Figure 3-19. Example #2 Output Wave



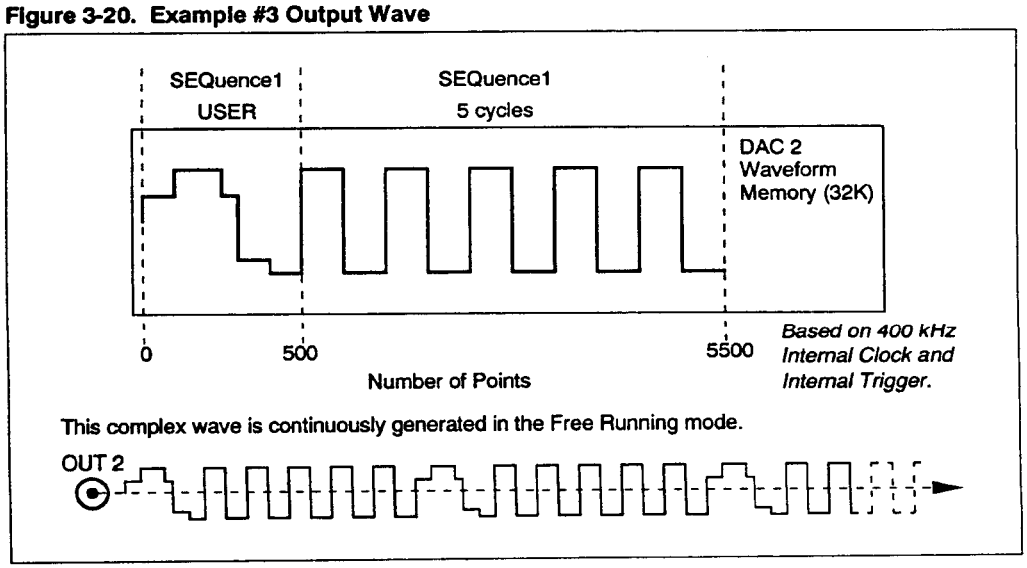
3.6.3.4 *Example #3: Free Running Complex Wave*

The following sequence continuously generates 1 cycle of a pre-defined USER function followed by 5 cycles of a 400-Hz, 1-volt square wave on channel 2. Notice that the Free Running Mode is set by the TRIG2:COUNt 0 command.

```
*RST ..... Reset the instrument.
TRIG2:SEQ1:STAR ..... Start of SEQuence1 for OUT 2.
  SOUR2:FUNC USER ..... Select user-defined function.
  OUTP2:FILT OFF;STAT ON ... Select filter off; output ON.
  TRIG2:SEQ1:COUN 1 ..... Set SEQuence1 Counter to 1.
TRIG2:SEQ1:STOP ..... End of SEQuence1.
TRIG2:SEQ2:STAR ..... Start of SEQuence2 for OUT2.
  SOUR2:FUNC SQU ..... Select square wave.
  SOUR2:VOLT 1;FREQ 400 ..... Select 1-volt amplitude; 400 Hz.
  TRIG2:SEQ2:COUN 5 ..... Set SEQuence1 Counter to 5.
TRIG2:SEQ2:STOP ..... End of SEQuence2.
TRIG2:COUN 0 ..... Select Free Running Mode.
INIT2 ..... Arm trigger sequence.
```

NOTE: Because of the methods used to calculate waveforms, the actual output may not appear exactly as illustrated.

Figure 3-20. Example #3 Output Wave



3.7 CLOCK SOURCES

The Digital-to-Analog Converter (DAC) is updated with new data by the Sampling Rate Clock. The clock is selected with the following command:

SOURce:ROSCillator:SOURce INT | EXT | TTLTrg<0-7>

INT ..... 400 kHz Internal Sampling Rate Clock

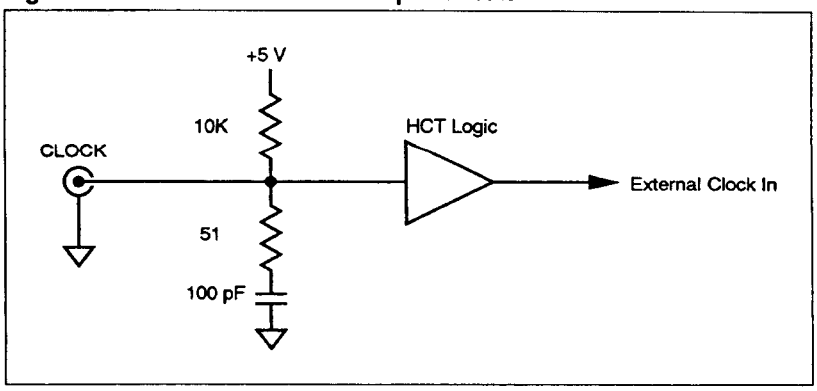
EXT ..... External Sampling Rate Clock; Front Panel CLOCK BNC (Figure 3-21): Accepts a TTL, active low clock, 120 ns minimum pulse width, 400 kHz maximum frequency. For generating Sine waves, the CLOCK input should be a 350 to 400-kHz, negative-going pulse with a pulse width of 1.2us ±50ns. For low-distortion, the clock signal should have less than 100 ps of jitter. If jitter exceeds 100 pS, the output DAC noise and distortion are degraded. The CLOCK input can be used to control Track-and-Hold. See Section 3.3.8.3.

TTLTrg<0-7> .. External Sampling Rate Clock from TTLTRG 0-7.

NOTE: To generate low-distortion sine waves, set the Front Panel CLOCK input frequency to 350 to 400-kHz. For other functions, there are no External Clock frequency limits.

Do not use an External Sampling Rate Clock for setting up the output functions. Perform all output setup using the Internal Sampling Rate Clock and then switch to the External Sampling Rate Clock for the final operation.

Figure 3-21. Front Panel CLOCK Input Circuit



3.8 DIAGNOSTIC TESTS

The DIAGnostic Subsystem contains commands that perform specific diagnostic tests on the DBS 8750. For all diagnostic tests, the front panel PASS/FAIL LED is red while the test is running and switches to green when the test passes. Use the query command DIAG? to obtain the status following any test.

DIAGnostic

:EPRom ..... EPROM Checksum Test.

:SRAM ..... Static RAM (SRAM) Test.

:ANALog ..... DAC Output Test.

:DRAM ..... DAC Waveform Memory Test.

:REGS ..... Register Test.

:ALL ..... All Diagnostic Tests.

3.8.1 EPROM Checksum Test

Sending DIAGnostic:EPRom calculates a checksum and compares it with the correct checksum stored in the EPROM. The test passes if the calculated checksum is equal to the stored checksum.

3.8.2 SRAM Test

Sending DIAGnostic:SRAM checks the results of the SRAM diagnostic test which is initiated by any one of three methods: (1) cycle the power off and on, (2) press the front panel RESET push-button, or (3) send the *RST command.

This command returns the contents of a memory diagnostic register. A return of 0xFA051793 means that the SRAM test passed.

3.8.3 DAC Output Test

Sending DIAGnostic:ANALog initiates a quick functional check of the DAC outputs. Various voltages are compared to one another to verify that both channels are operating. Both outputs are switched OFF by this test.

NOTE: This test disconnects the outputs and destroys any programmed waveforms on either channel.

3.8.4 DAC Waveform Memory Test

Sending DIAGnostic:DRAM initiates a routine that stores a series of patterns throughout the DAC 1/2 Waveform Memory then reads the stored patterns to verify proper operation. SRAM is not tested during this routine.

NOTE: This test disconnects the outputs and destroys any programmed waveforms on either channel.

3.8.5 Register Test

Sending DIAGnostic:REGS initiates a routine that verifies proper operation of the DAC Control, Offset, and VXI Control registers.

NOTE: This test disconnects the outputs and destroys any programmed waveforms on either channel.

3.8.6 Running All Diagnostic Tests

Sending DIAGnostic:ALL performs all diagnostic tests in the following order:

EPROM Checksum Test

DAC Register Test

Offset Register Test

VXI Control Register Test

SRAM Test

DAC Wavfom Memory Test

DAC Output Test

See also the *TST? command in Section 4.

Section 4
DBS 8750 Command Reference

4.1 INTRODUCTION

This section describes the DBS 8750 Command Set. A command index is provided on pages 4-4 and 4-5. The command descriptions are grouped by subsystems and each command and the subsystem to which it belongs is referenced in the header of each page. The commands are defined in terms of function, syntax, parameter selections and related commands.

4.2 COMMAND SYNTAX

The commands are organized in a hierarchical structure called a Command Tree (Figure 4-1). Associated commands are organized into groups, each under a common subsystem. The command mnemonics are called keywords. Indentation of the keywords shows the hierarchy within the command tree. A colon (:) is used to separate one command tree level from another such as in SOURce:ROSCillator:SOURce TTLTrg<0-7>.

NOTE: When programming this instrument it is necessary to read the Data Low Register after sending every command except Reset, *RST. The normal response is the new line character (x0A) or a status message.

4.2.1 Abbreviated Commands

Most commands use upper and lower case letters. Upper case letters indicate the abbreviated form of the command. For example, if a keyword is written as SOURce, then SOUR and SOURCE are acceptable forms. Other forms will generate an error. Although commands & queries are shown in the examples as uppercase characters, the DBS 8750 is not case-sensitive.

4.2.2 Command Concatenation

A semi-colon is used to concatenate commands within a common subsystem. Figure 4-2 shows and example using the three separate commands on the left and writes them in the concatenated format on the right. Concatenated commands are more efficient because they require fewer read-back commands, and therefore, fewer bus cycles.

Figure 4-1. DBS 8750 Command Tree

```
CALCulate Subsystem ..... CALCulate
                            :MATH <Segment 1> <Segment 2> <Segment 3> ..... <Segment 16>
DIAGnostic Subsystem ..... DIAGnostic
                            :EPRom
                            :SRAM
                            :ANALog
                            :DRAM
                            :REGS
                            :ALL
MEMory Subsystem ..... MEMory
                            :WRITe<sp><VXI Addr><sp><Npts><sp><Buffer Name>
                            :READ<sp><VXI Addr><sp><Npts><sp><Buffer Name>
                            :SAV1(2)<sp><Buffer Name>
                            :ACQuire<sp><Npts><sp><Buffer Name>
                            :CATalog[:ALL]? | BINary?
                            :DELete<sp><Buffer Name> | ALL
                            :FREE[:ALL]? | BINary?
                            :MALLocate<sp><Buffer Name><sp><Npts>
OUTPut Subsystem ..... OUTPut[1](2)
                            :ATTenuation<sp><0 to 90>
                            :FILTer<sp>ON | OFF
                            :STATe<sp>ON | OFF
                            :DRIVe<sp>SINGle-ended | DIFFerential
                            :TRACk<sp>ON | OFF
                            :GAT
                               :DELay<sp><0.00002 to 41.945>
                            :SYNC
                            TTLTrg<sp>ON | OFF
SOURce Subsystem ..... SOURce[1](2)
                            :FREQuency<sp><12.5 to 20000>
                            :VOLTage<sp><–10 to +10>
                               :OFFSet<sp><–10 to +10>
                            :FUNCtion<sp>DC | SINe | SQUare | TRIangle | NOISe | USER |
                            MEMory
                            :ROSCillator
                               :SOURce<sp>INT | EXT | TTLTrg<0-7>
                                     :AUTO SHON | SHOF
TRIGger Subsystem ..... TRIGger[1](2)
                            :SEQuence[1](2)
                               :STARt
                               :STOP
                               :COUNt<sp><numeric>
                            :COUNt<sp><numeric>
                            :SOURce<sp>EXT | INT
                          INITiate[1](2)
                          ABORt[1](2)
```

Figure 4-2. Concatenating Commands

Separate Commands	Concatenated Format
SOURce:FUNCtion SINe *<Read Data Low>* SOURce:VOLTage 10 *<Read Data Low>* SOURce:FREQuency 400 *<Read Data Low>*	SOURce:FUNCtion SINe;VOLTage 10;FREQuency 400 *<Read Data Low>* **NOTE: Concatenated commands require fewer read-back commands. Therefore, fewer bus cycles are needed.**

4.2.3 Command Parameters

- Numeric – Some commands use variable parameters which include a range of values, such as output attenuation, gate delay, frequency, amplitude, and offset voltage. Numeric variables can be expressed using signs, decimal points and scientific notation. Numeric parameter values must be preceded by a space (<sp>).
- Boolean – Boolean parameters are a single binary condition that is either true or false such as ON, OFF, 1 or 0.
- Discrete – Discrete parameters use mnemonics to represent each valid setting. For example, the command OUTPut[1]:DRIVe <mnemonic> selects the output drive configuration using the mnemonics "SINGle-ended" and "DIFFerential."

4.2.4 Optional Keywords and Parameters

Command keywords and parameter values/selections that are shown within brackets ([ ]) are optional. For parameters, the brackets indicate default values/selections. The brackets are not part of the command and if you don't specify a value/selection in the command, the default value/selection is used. Any alternate value is shown in parentheses immediately following the bracketed value. For example, OUTPut[1](2):STATe ON | OFF.

4.2.5 Output Channel Selection

The "[1](2)" designation following the first keyword of a command is used to select the output channel. For example, OUTPut2:STATe ON switches channel 2 ON, whereas OUTPut1:FILTer ON or OUTPut:FILTer ON switches the channel 1 output filter ON.

4.2.6 Query Commands

Unless noted otherwise, parameter settings are queried by adding a question mark to the command which set the parameter. For example, sending SOURce1:VOLTage 10 sets the amplitude of channel 1 to 10 volts. Sending the command SOURce1:VOLTage? reads back the current value.

4.4 COMMAND INDEX

SOURce Subsystem

Commands used for setting function, frequency, amplitude and offset voltage.

TRIGger Subsystem

Contains used for controlling trigger sequences.

System Commands/Queries

VXI sytem-level commands.

CALCulate:MATH

Description: Calculates a USER output function. USER functions are selected using the SOURce:FUNC USER command.

Syntax: CALCulate[1](2):MATH<sp>FOR<sp><duration><sp><voltage><sp> FOR<sp><duration><sp><voltage><sp>FOR<sp><duration><sp><voltage> ..... [16 FOR statements, maximum]

FOR ..... A directive which begins the mathematical definition of a segment.

<duration> ..... The time period (seconds) during which the voltage function takes place.

<voltage> ..... Output voltage with respect to time. Expressed as a simple decimal value or a complex mathematical expression using the mnemonics and symbols in Table 4-1.

Notes:

1. When specifying duration and voltage, DO NOT use commas or suffixes such as "m" (milli), "μ" (micro), "k" (kilo), or "M" (Mega) - these are not recognized by the firmware.
2. Expressions are always evaluated from right to left and should fit on one line. If an expression does not fit, perform some precalculations. Use parentheses to group quantities properly.
3. CALCulate commands may consist of up to 16 segments and 859 characters, maximum. Each segment is defined by a "FOR<sp><duration><voltage>" statement.
4. Total USER function output must be equal to or less than 32K points which is the maximum capacity of the DAC 1 or 2 Waveform Memory.

Examples:

1. To calculate a 1-KHz, 5-volt sine wave on channel 2:

```
CALC2:MATH FOR 0.001 5*SIN(2*PI*t*1000)
SOUR2:FUNC USER
```

2. To calculate a 1-KHz, 5-volt sine wave on channel 1:

```
CALC1:MATH FOR 0.001 5*SIN(2*PI*t*1000)
SOUR1:FUNC:USER
```

Related Commands: SOUR:FUNC USER

Reset Condition: All programmed CALC:MATH functions are erased.

Table 4-1. CALC:MATH Command Mnemonics & Symbols

Mnemonic or Symbol	Function or Meaning
+	Positive value; addition
–	Negative value; subtraction
*	Multiply
/	Divide
^	Raised to the power
( )	Parentheses
e	2.7182818
PI	3.1415927
SIN	Sine
COS	Cosine
TAN	Tangent
ASIN	Arc sine
ACOS	Arc cosine
ATAN	Arc tangent
LN	Natural log (base e)
LOG	Common log (base 10)
ABS	Absolute value
t	Time variable (t=0 at start)

DIAGnostic:EPRom

Description: Calculates an EPROM checksum and compares it with the correct checksum stored in EPROM. If the numbers match, the test passes and the green LED is switched on.

Syntax: DIAGnostic:EPRom

Notes: Command has no effect on signal generation.

Example: DIAG:EPR

Reset Condition: This diagnostic is included in the self-test diagnostics which are invoked by a reset command.

DIAGnostic:SRAM

Description: Reports the status of the SRAM (DSP/User Memory) Diagnostic Test following the last reset or power-on sequence.

This command does not perform a SRAM diagnostic test in which all system memory is over-written.

Syntax: DIAGnostic:SRAM

Notes: To run the SRAM diagnostic test, do one of three actions:

(a) Cycle the power off and then on again.

(b) Press the front panel reset push-button.

(c) Send the *RST (Reset) command.

The SRAM diagnostic test does not test the DAC Waveform Memories.

This command does not effect signal generation.

Example: DIAG:SRAM

Reset Condition: This diagnostic is included in the self-test diagnostics which are invoked by a reset command.

DIAGnostic:ANALog

Description: Provides a quick functional test of the DAC outputs. Various output voltages are compared to one another in order to verify that both channels are generating waveforms. The front panel LED is red while the test is running, and switches to green if the test passes.

Syntax: DIAGnostic:ANALog

Note: Running DIAGnostic:ANALog disconnects the outputs and destroys any programmed waveform on both channels.

Example: DIAG:ANALOG

Reset Condition: This diagnostic is included in the self-test diagnostics which are invoked by a reset command.

DIAGnostic:DRAM

Description: Stores a series of patterns throughout the Waveform Memory and reads the stored patterns to verify that the DAC RAM is working properly. SRAM is not tested. The front panel LED is red while the test runs and switches to green if the test passes.

Syntax: DIAGnostic:DRAM

Note: Running DIAGnostic:DRAM disconnects the outputs and destroys any programmed waveform on both channels.

Example: DIAG:DRAM

Reset Condition: This diagnostic is included in the self-test diagnostics which are invoked by a reset command.

DIAGnostic:REGS

Description: Checks for proper operation of the DAC Control, Offset, and VME Control registers. The front panel LED is red while the tests run, and switches to green if all 3 tests pass.

Syntax: DIAGnostic:REGS

Note: Running DIAGnostic:REGS disconnects the outputs and destroys any programmed waveform on both channels.

Example: DIAG:REGS

Reset Condition: This diagnostic is included in the self-test diagnostics which are invoked by a reset command.

DIAGnostic:ALL

Description: The DIAGnostic:ALL command performs the following, in order: EPROM Checksum Test, DAC Register Test, Offset Register Test, VXI Control Register Test, SRAM Status Register verification, DAC RAM Test, and Analog output Test. The front panel LED is red while the tests run, and switches to green if all tests pass.

Syntax: DIAGnostic:ALL

Notes:
1. Running DIAGnostic:ALL disconnects the outputs and destroys any programmed waveform on both channels.
2. The diagnostic halts upon detecting an error, after which no further testing is performed. Thus, an EPROM checksum error prevents the register, memory, and analog tests from running.

Example: DIAG:ALL

Reset Condition: This command invokes the same self-test diagnostic routine which is invoked by a reset command.

DIAGnostic?

Description: Generates an ASCII error message.

Syntax: DIAGnostic?

Response: One of the following messages may be returned:

0, "No Error; Self-test PASSED"
-330, "Self-test failed; Static RAM failed self-test"
-330, "Self-test failed; DAC RAM 0 memory error"
-330, "Self-test failed; DAC RAM 1 memory error"
-330, "Self-test failed; DAC control register error"
-330, "Self-test failed; DAC rate register error"
-330, "Self-test failed; DAC 0 address register error"
-330, "Self-test failed; DAC 1 address register error"
-330, "Self-test failed; VME offset register error"
-330, "Self-test failed; VME control register error"
-330, "Self-test failed; EPROM memory error"
-330, "Self-test failed; DAC Analog output error"

Notes:
1. Running DIAGnostic? doesn't affect signal generation.
2. If no errors are detected, the following message is returned: 0, "No Error; Self-test PASSED".

Example: DIAG?

Reset Condition: Reflects the status of the last diagnostic test performed.

MEMory:WRITe

Description: Initiates a DMA write cycle to transfer a specified number of data points (words) from a buffer in DSP/User Memory to VXI memory.

Syntax: MEMory:WRITe<sp><VXI Addr><sp><Npts><sp><Buffer Name>

<VXI Addr> ..... VXI starting memory address

<Npts> ..... number of data points in the buffer

<Buffer Name>..... name of memory buffer (32 characters, max.)

Example: To transfer 1024 words from "VDATA" to VXI memory, starting at VXI memory address 200000h:

MEM:WRITe 200000 1024 VDATA

Related Commands: All other MEMory commands.

Reset Condition: All buffers in SRAM are deleted.

MEMory:READ

Description: Initiates a DMA read cycle to transfer a block of data from VXI memory to a specified DSP/User Memory buffer.

Syntax: MEMory:READ<sp><VXI Addr><sp><Npts><sp><Buffer Name>

<VXI Addr> ..... Starting address of the VXI memory block to be read.

<Npts> ..... Number of data points (bytes) to be read. <Npts> must not exceed the amount of free memory (see MEM:FREE?) minus 35 points for buffer overhead.

<Buffer Name> ... Name of memory buffer (32 characters, max.)

Notes:

1. If the specified buffer does not exist, a new one is created. If the buffer already exists, the previous waveform data is lost. If enough memory cannot be allocated for the user buffer, an error message is generated and the buffer is not created.
2. If the buffer exists and a subsequent read requires a larger buffer, the program attempts to re-size the buffer. If this fails, an error message is generated and enough memory is re-allocated for the original buffer. However, the data in the original buffer may be corrupted.
3. All DBS 8750 buffers are BINary. Up to 20 user buffers may be created.

4. The DBS 8750 must allocate contiguous blocks of memory. If you have been creating and deleting buffers, a MEM:READ may fail even though the total amount of free memory is sufficient.

Example: To read 1024 words from VXI memory, starting at address 200000h, and store them in buffer "VDATA":

MEM:READ 200000 1024 VDATA

Related Commands: Other MEMory commands, SOUR:FUNC:MEM

Reset Condition: All buffers in SRAM are deleted.

MEMory:SAV

Description: Transfers the waveform data that currently resides in DACWaveform Memory to a specified buffer in DSP/User Memory.

Syntax: MEMory:SAV1(2)<sp><Buffer Name>

Notes:

1. If the specified buffer does not exist, a new one is created. If the buffer already exists, the previous waveform data is lost. If enough memory cannot be allocated for the user buffer, an error message is generated and the buffer is not created.
2. If the buffer exists and a subsequent read requires a larger buffer, the program attempts to re-size the buffer. If this fails, an error message is generated and enough memory is re-allocated for the original buffer. However, the data in the original buffer may be corrupted.
3. All DBS 8750 buffers are BINary. Up to 20 user buffers may be created.
4. The DBS 8750 must allocate contiguous blocks of memory. If you have been creating and deleting buffers, a MEM:READ may fail even though the total amount of free memory is sufficient.
5. The number of waveform points must not exceed the amount of free memory (see MEM:FREE?) minus 35 points for buffer overhead. However, the DBS 8750 must allocate contiguous blocks of memory. If you have been creating and deleting buffers, a MEM:READ may fail even though the total amount of free memory is sufficient.

Example: To save a waveform in DAC #2 memory to buffer VDATA1
MEM:SAV2 VDATA1

Related commands: SOUR:FUNC:MEM, other MEMory commands.

Reset Condition: All buffers in SRAM are deleted.

MEMory:ACQuire

Description: Transfers a block of data from the LOCALbus and stores it in a specified DSP/User Memory buffer.

Syntax: MEMory:ACQuire<sp><Npts><sp><Buffer Name>

<Npts> ..... number of data points (bytes) to be stored. <Npts> must not exceed the amount of free memory (see MEM:FREE?) minus 35 points for buffer overhead.

<Buffer Name> ..... name of DBS 8750 buffer to receive data.

Notes:

1. If the specified buffer does not exist, a new one is created. If the buffer already exists, the previous waveform data is lost. If enough memory cannot be allocated for the user buffer, an error message is generated and the buffer is not created.
2. If the buffer exists and a subsequent read requires a larger buffer, the program attempts to re-size the buffer. If this fails, an error message is generated and enough memory is re-allocated for the original buffer. However, the data in the original buffer may be corrupted.
3. All DBS 8750 buffers are BINary. Up to 20 user buffers may be created.
4. The DBS 8750 must allocate contiguous blocks of memory. If you have been creating and deleting buffers, a MEM:READ may fail even though the total amount of free memory is sufficient.

Example: To save 1024 data words from the LOCALbus in buffer "VDATA":

MEM:ACQ 1024 VDATA

Related Commands: Other MEMory commands, SOUR:FUNC:MEM.

Reset Condition: All buffers in SRAM are deleted.

MEMory:CATalog

Description: Returns a memory status in the following format:

<filled memory>,<free memory>,<buffer name>,<type>,<size>,
<buffer name>,<type>,<size><buffer name>,<type>,<size> ... etc.

Syntax: MEMory:CATalog[:ALL]? or MEMory:CATalog:BINary?

Note: MEMory:CATalog[:ALL]? returns information about all memory buffers, including used and available memory and a list of buffers and buffer sizes.

Example: With no existing buffers.: Command: MEM:CAT?
Response: 0,17150

With three 1000-point buffers: Command: MEMory:CATalog?
Response: 3105,14045
BUFFER1,BIN,1000
BUFFER2,BIN,1000
BUFFER3,BIN,1000

Related Commands: Other MEMory commands.

Reset Condition: All existing buffers are deleted.

MEMory:DELete

Description: Removes DSP/USER Memory buffers.

Syntax: MEMory:DELete<sp><mnemonic>

Mnemonic Parameters: ALL ..... All buffers
<Buffer Name> ..... Only named buffer

Example: To delete buffer VDATA and then all buffers:

```
MEM:DEL VDATA
MEM:DEL ALL
```

Related Commands: All MEMory commands.

Reset Condition: All buffers are deleted.

MEMory:FREE?

Description: Returns the amount of available and used DSP/User Memory in the following format:

<free memory>,<filled memory>

Syntax: MEMory:FREE:[ALL]? or MEM:FREE:BINary?

Notes:
1. All DBS 8750 buffers are BINARY.
2. MEM:FREE? and MEM:CAT? return the number of free and used words (16 bits) not bytes.

Example: MEM:FREE:ALL?

Related Commands: Other MEMory commands.

Reset Condition: All free memory is available.

MEMory:MALLocate

Description: Allocates a block of DSP/User Memory by creating a buffer of a specified point size. The command reponse is either the buffer's offset (in hexidecimal) from the VXI A24 base address, or the error message "-225, Out of memory; cannot allocate memory for buffer."

Syntax: MEMory:MALLocate<sp><Buffer Name><sp><Npts>

<Buffer Name> ..... Up to 31 characters, excluding whitespace, colons, and semicolons.

<Npts> ..... A positive decimal integer representing the number of data points allocated to the buffer. <Npts> must not exceed the amount of free memory (see MEM:FREE?) minus 35 points for buffer overhead.

Notes:

1. The DBS 8750 must allocate contiguous blocks of memory. If you have been creating and deleting buffers, this command may fail even though the total amount of free memory is sufficient.
2. This command can be used to re-size an existing named buffer. If a sufficient amount of contiguous memory space is not available, the command may fail and generate error message "-225, Out of memory; cannot resize the buffer".
3. The 8750 frees existing buffer memory before attempting to reallocate the buffer. If resizing fails, the buffer re-allocates memory to the original size, and data in the existing buffer may be corrupted.
4. Since only 16-bit transfers are allowed, only EVEN addresses are valid.
5. The DBS 8750 is a bus master during data transfers.
6. Buffers defined by this command are not initialized by the 8750. The 8750 assumes that the whole allocated buffer area is used for the waveform. Thus, if the buffer is used to generate an output before it has been completely initialized, the waveform may have unpredictable values.
7. There is no bounds checking while writing the user buffer. Writing to unallocated memory, or writing more points than have been allocated may cause unpredictable results, including "hanging" the VME bus.

Example: A DBS 8750 using offset 0x480000 allocates 1000 wave points beginning at VXI A24 address 0x4D6A50.

MEM:MALL NEW_WAVEFORM 1000

Response: 0x00056A50

OUTPut:ATTenuation

Description: Sets output signal attenuation.

Syntax: OUTPut[1](2):ATTenuation <sp><numeric> (Query: OUTP:ATT?)

Numeric Parameters: 0 to 90

Note: Output attenuation is accomplished by a combination of a switched resistor network and voltage reduction performed by mathematical calculation. The resistor network provides attenuation steps of 0dB, 10dB, 20dB and 30dB. Other settings are completed by mathematical calculations which lower the output voltage an additional amount.

Example: To set –20dB attenuation on channel 2: OUTP2:ATT 20

Related Commands: None.

Reset Condition: Zero (0) attenuation for both channels.

OUTPut:FILTer

Description: Connects or disconnects the output filter for channel 1 or 2.

Syntax: OUTPut[1](2):FILTer<sp><boolean> (Query: OUTP:FILT?)

Boolean Parameters: ON or 1 ..... connects the 6-pole filter

OFF or 0 ..... disconnects the 6-pole filter

Note: Set output filter "ON" to generate low distortion SINE waves.

Example: To set the channel 2 output filter "ON": OUTP2:FILT ON

Related Commands: OUTPut[1](2):FILT 1, OUTPut[1](2):FILT 0.

Reset Condition: Filter OFF.

OUTPut:STATe

Description: Switches outputs OUT 1 and OUT 2 on or off.

Syntax: OUTPut[1](2):STATe<sp><boolean> (Query: OUTP?)

Boolean Parameters: ON or 1 ..... Connects output amplifiers to the output jacks

OFF or 0 ..... (power-on state) Disconnects output amplifiers and switches output to ground through a 50 or 600 ohm resistor.

Example: To set the channel 2 output filter "ON": OUTP2:STAT ON

Related Commands: OUTP ON, OUTP OFF, OUTP 1, OUTP 0.

Reset Condition: OUTPut[1](2):STAT OFF.

OUTPut:DRIVe

Description: Sets the output stage to either a single-ended or differential output drive.

Syntax: OUTPut[1](2):DRIVe<sp><mnemonic> (Query: OUTP:DRIV?)

Mnemonic Parameters: SINGle-ended .... single-ended drive configuration

DIFFerential ..... differential drive configuration

Note:

Example: To set the channel 2 output to a differential drive: OUTP2:DRIV DIFF

Related Commands: None.

Reset Condition: OUTPut[1](2):DIFFerential.

OUTPut:TRACk

Description: Selects Tracking Mode.

Syntax: OUTPut:TRACk<sp><boolean> (Query: OUTP:TRAC?)

Boolean Parameters: ON or 1 ..... Track Only

OFF or 0 ..... Track-and-Hold

Notes:

1. In the Track-and-Hold mode (OUTPut:TRACk OFF), noise spikes appear on the output. The output filter can be switched on to reduce the effect. If the output filter is not used, configure the deglitching filter* to be in line (jumper selection required).
2. Use the Track Only Mode to generate DC and square waves.
3. Code-dependent glitch energy is increased in the Track Only Mode.
4. Using the deglitching filter* increases square wave settling times. Disconnect the deglitching filter* to reduce settling times.

Example: To select Track Only Mode: OUTP:TRACk ON

Related Commands: None.

Reset Condition: OUTPut:TRACk OFF.

OUTPut:GAT:DELay

Description: Sets the amount of time to delay the GATE output following an input trigger signal.

Syntax: OUTPut:GAT:DELay<sp><numeric> (Query: OUTP:GAT:DELay?)

Numeric Parameters: 0.00002 to 41.945 seconds in steps of 2.5 us

Example: To set the output gate delay to 100 usec: OUTP:GAT:DEL 0.0001

Related Commands: None.

Reset Condition: Delay is set to zero (0).

* For hardware revision 1 and up only.

OUTPut:SYNC

Description: Re-starts waveform generation on outputs 1 and 2 at the same time regardless of their current phase relationship. The waveforms stay synchronized only if they have a harmonic relationship from the start.

Syntax: OUTPut:SYNC

Parameters: None

Notes: 1. Sine waves and square waves will have a 0 degree phase difference when OUTP:SYNC is executed.

2. Triangle waves will have a 90 degree phase difference with sine & square waves.

Example: To set the outputs to run synchronously: OUTP:SYNC

Related Commands: None.

Reset Condition: Outputs are synchronized.

OUTPut:TTLTrg

Description: Enables the TTLTRG lines to drive the selected output onto the bus. TTLTRG 0-3 can drive the Sampling Rate Clock and TTLTRG 4-7 can drive the GATE*.

Syntax: OUTPut:TTLTrg<sp><boolean>

Boolean Parameters: ON or 1 ..... Enable TTLTRG

OFF or 0 ..... Disable TTLTRG

Notes: The TTLTRG line is selected by installing a hardware jumper. See Section 2.2.3.

Example: To enable the TTLTRG lines, send OUTP:TTLT ON.

Related Commands: None.

Reset Condition: OUTPut:TTLTrg OFF.

* For hardware revision 1 and up only. See Appendix D for revision 0 hardware TTLTRG I/O functions.

SOURce:FREQuency

Description: Sets the frequency of standard output waveforms.

Syntax: SOURce[1](2):FREQuency<sp><Numeric> (Query: SOUR:FREQ?)

Numeric Parameters: 12.5 Hz to 20000 Hz (using internal clock)

Note: Using an external clock, the low frequency limit for output waves is defined by the external clock.

Example: To set frequency to 15.12 kHz for channel 2: SOUR2:FREQ 15120

Related Commands: All OUTPut commands.

Reset Condition: Frequency is set to 1000 Hz for both channels.

SOURce:VOLTage

Description: Sets voltage amplitude for standard output waveforms.

Syntax: SOURce[1](2):VOLTage<sp><numeric> (Query: SOUR:VOLT?)

Numeric Parameters: –10V to +10V

Example: To set the channel 2 output voltage to 9.981 Volts: SOUR2:VOLT 9.981

Related Commands: All OUTPut commands.

Reset Condition: Both output channels are set to 0 Volts.

SOURce:VOLTage:OFFSet

Description: Sets DC offset for standard output waveforms.

Syntax: SOURce[1](2):VOLTage:OFFSet<sp><numeric>

(Query: SOUR:VOLT:OFFS?)

Numeric Parameters: –10V to +10V

Note: Output voltage plus offset cannot exceed ±10 Volts.

Example: To set the channel 2 output offset to 1.234 Volts:

SOUR2:VOLT:OFFS 1.234

Related Commands: All OUTPut commands.

Reset Condition: Offset is 0 Volts for both channels.

SOURce:FUNCtion DC

Description: Selects the DC volts output function.

Syntax: SOURce[1](2):FUNCtion<sp>DC (Query: SOUR:FUNC?)

Notes: 1. The voltage range for a DC waveform is ±10 volts (single-ended) or ±5 volts (differential).

2. To generate a clean DC volt level, switch the Tracking Mode to Track Only (OUTPut[1](2):TRACk ON) and switch the output filter ON (OUTPut[1](2):FILTer ON).

Example: To set the DC function for channel 2:

SOUR2:FUNC DC

OUTP2:FILT ON

Related Commands: All OUTPut commands.

Reset Condition: Sets DC function to 0 Volts for both channels.

SOURce:FUNCtion SINe

Description: Selects the Sine Wave output function.

Syntax: SOURce[1](2):FUNCtion<sp>SINe (Query: SOUR:FUNC?)

Notes: 1. Sine wave generation begins only after frequency, amplitude and function are defined.

2. For low-distortion sine wave generation, the output filter must be "ON".

3. Voltage range = ±10 Volts.

Example: To set a 9.99 KHz, 4.777-volt sine wave for channel 2:

```
SOUR2:VOLT 4.777
SOUR2:FREQ 9990
SOUR2:FUNC SIN
OUTP2:FILT ON
```

Related Commands: All OUTPut commands.

Reset Condition: Sets DC function to 0 Volts for both channels.

SOURce:FUNCtion SQUare

Description: Selects the Square Wave output function.

Syntax: SOURce[1](2):FUNCtion<sp>SQUare (Query: SOUR:FUNC?)

Notes: 1. When using the Internal Sampling Rate Clock, a ±5-mV, 1.5-us spike appears on the unfiltered output every 2.5 usec. This limits the amplitude accuracy of the generated square wave. For better performance, use an External Sampling Rate Clock.

2. Wave generation begins only after frequency, amplitude and function are defined.

3. Voltage range: ±10 Volts.

Example: To generate a 9.99 kHz, 4.777-volt square wave on channel 2.

```
SOUR2:VOLT 4.777
SOUR2:FREQ 9990
SOUR2:FUNC SQU
```

Related Commands: OUTPut commands.

Reset Condition: Sets DC function of both channels to 0 Volts.

SOURce:FUNCtion TRIangle

Description: Selects the Triangle output function.

Syntax: SOURce[1](2):FUNCtion<sp>TRIangle (Query: SOUR:FUNC?)

Notes: 1. Wave generation begins only after frequency, amplitude and function are defined.

3. Voltage range: ±10 Volts.

Example: To generate a 9.99 kHz, 4.777-volt triangle wave on channel 2:

```
SOUR2:VOLT 4.777
SOUR2:FREQ 9990
SOUR2:FUNC TRI
```

Related Commands: OUTPut commands.

Reset Condition: Sets DC function of both channels to 0 Volts.

SOURce:FUNCtion NOISe

Description: Selects the Noise output function.

Syntax: SOURce[1](2):FUNCtion<sp>NOISe (Query: SOUR:FUNC?)

Notes: Maximum noise amplitude: ±10 Volts.

Example: To generate ±5.00-volt noise for channel 2:

```
SOUR2:FUNC NOIS
SOUR2:VOLT 5.0
```

Related Commands: OUTPut commands.

Reset Condition: Sets DC function of both channels to 0 Volts.

SOURce:FUNCtion USER

Description: Selects the USER output function. The USER output function is defined by the CALC:MATH command. If the mathematical expression was already defined and entered using the CALC:MATH command, the actual calculation is executed only after entering the SOURce:FUNCtion USER command.

Syntax: SOURce[1](2):FUNCtion<sp>USER (Query: SOUR:FUNC?)

Example: To generate a 1-kHz, 2-volt sine wave on channel 1:

```
CALC:MATH FOR 0.001 2*SIN(2*PI*t*1000)
SOUR:FUNC USER
SOUR:OUTP:FILT ON
SOUR:OUTP:STAT ON
```

Related Commands: OUTPut commands, CALC:MATH <expression>.

Reset Condition: Erases FUNCtion<sp>USER and CALC:MATH information.

SOURce:FUNCtion MEMory

Description: Selects the MEMory output function. The MEMory output function is determined by the selected memory buffer.

Syntax: SOURce[1](2):FUNCtion<sp>MEMory<sp><Buffer Name>

(Query: SOUR:FUNC?)

Example: To generate a waveform on channel 1 using memory buffer <VDATA>:

```
SOUR:FUNC MEM VDATA
```

Related Commands: OUTPut commands, MEMory commands.

Reset Condition: Erases FUNCtion:MEMory and all memory buffers.

NOTE: Do not use an External Sampling Rate Clock for setting up the output functions. Perform all output setup using the Internal Sampling Rate Clock and then switch to the External Sampling Rate Clock for the final operation.

SOURce:ROSCillator:SOURce

Description: Selects the source of the Sampling Rate Clock to control the update rate for both channels. Channels cannot be clocked separately.

Syntax: SOURce:ROSCillator:SOURce<sp>INT|EXT|TTLTrg<0-7>

(Query: SOUR:ROSC:SOUR?)

Mnemonic Parameters:

INT	Selects the 400-kHz Internal Sampling Rate Clock
EXT	Selects the Front Panel CLOCK input
TTLTrg<0-7>	Selects one of eight TTL Trigger lines

Notes:

1. The Internal Sampling Rate Clock is always 400 kHz. The output filter is optimized for sine wave generation at this internal rate.
2. In general, there is no low-frequency limit for an external clock. The external clock should be a TTL-compatible, negative-going pulse, 120 ns pulse width, 400 kHz maximum frequency.
3. The Front Panel CLOCK input can also be used to control the DAC internal Track-and-Hold Output Amplifiers. Use a negative-going clock pulse, 350 to 400 kHz, 1.2 µs ± 50 ns wide. Refer to the SOURce:ROSCillator:SOUR:AUTO SHON command.
4. An External Sampling Rate Clock is recommended when accurate square or staircase waveforms need to be generated. See Section 3.3.14 for details.

Example: SOUR:ROSC:SOUR INT ... Selects the Internal Sampling Rate Clock
SOUR:ROSC:SOUR EXT ... Selects the Front Panel CLOCK input
SOUR:ROSC:SOUR TTLT4 ... Selects TTL Trigger Line #4

Related Commands: SOURce:ROSCillator:SOUR:AUTO

Reset Condition: Sends SOURce:ROSCillator:SOURce INT.

SOURce:ROSCillator:SOURce:AUTO

Description: Synchronizes the Track-and-Hold to either the Front Panel CLOCK input or the Internal Sampling Rate Clock.

Syntax: SOURce:ROSCillator:SOURce:AUTO<sp><mnemonic>

(Query: SOUR:ROSC:SOUR AUTO?)

Mnemonic Parameters:

SHON	Synchronize Track-and-Hold to Front Panel CLOCK
SHOF	Synchronize Track-and-Hold to the Internal Sampling Rate Clock

Notes:

1. The CLOCK input requires a negative-going pulse, 1.2µs ± 50ns wide, 350 to 400 kHz.
2. To generate a sine wave signal synchronously using an External Sampling Rate Clock, use SOUR:ROSC:SOUR:AUTO SHON.

Example: To synchronize Track-and-Hold with the Front Panel CLOCK input:

SOUR:ROSC:SOUR:AUTO SHON

Related Commands: SOURce:ROSC:SOURce EXT

Reset Condition: Sends SOURce:ROSCillator:SOURce:AUTO SHOF.

NOTE: In order for Trigger Sequences to run correctly, the DBS 8750 must be equipped with firmware revision 2.0 or higher. To determine hardware/software revision levels, refer to the *IDN? command in Section 4.

TRIGger:SEQuence:STARt

Description: Defines the beginning of a Trigger Sequence*. SEQ1 must precede SEQ2.

Syntax: The TRIGger[1](2):SEQuence[1](2):STARt (Query: None)

Notes: 1. To program each Trigger Sequence, send the following set of commands:

TRIGger[1](2):SEQuence[1](2):STARt

•
•
•

<sequence definition statements>

•
•
•

TRIGger[1](2):SEQuence[1](2):STOP.

2. The designation "[1](2)" following the keyword, TRIGger, is used to select the channel being programmed.
3. Two TRIGger sequences are allowed per channel. The designation "[1](2)" following the keyword, SEQuence, is used to delineate sequence #1 from sequence #2.
4. Always start a trigger sequence by sending the command "INITiate". The sequence can be stopped anytime with the "ABORt" command. Analog output control commands (OUTPut commands) can be included as part of trigger sequences, but are executed only once, following INITiate.
5. Sending an ABORt does not effect the output settings.

Example: TRIGger1:SEQuence1:STARt
TRIG2:SEQ1:STAR

Related Commands: INITiate, ABORt, SOURce commands.

Reset Condition: Erases all TRIGger SEQuences.

NOTE: In order for Trigger Sequences to run correctly, the DBS 8750 must be equipped with firmware revision 2.0 or higher. To determine hardware/software revision levels, refer to the *IDN? command in Section 4.

TRIGger:SEQuence:STOP

Description: Defines the end of a Trigger Sequence.

Syntax: TRIGger[1](2):SEQuence[1](2):STOP (Query: None.)

Notes: 1. To program each Trigger Sequence, send the following set of commands:

TRIGGger:SEQuence[1](2):STARt

•
•
•

<sequence definition statements>

•
•
•

TRIGger:SEQuence[1](2):STOP.

2. The designation "[1](2)" following the keyword, TRIGger, is used to select the channel being programmed.

3. Two TRIGger sequences are allowed per channel. The designation "[1](2)" following the keyword, SEQuence, is used to delineate sequence #1 from sequence #2.

4. Always start a trigger sequence by sending the command "INITiate". The sequence can be stopped anytime with the "ABORt" command.

Example: TRIGger2:SEQuence1:STOP
TRIG1:SEQ2:STOP

Related Commands: INITiate, ABORt.

Reset Condition: Erases all TRIGger SEQuences.

NOTE: In order for Trigger Sequences to run correctly, the DBS 8750 must be equipped with firmware revision 2.0 or higher. To determine hardware/software revision levels, refer to the *IDN? command in Section 4.

TRIGger:SEQuence:COUNt

Description: Specifies how many times to repeat the conditions of the specified Trigger Sequence. This specifies the number of waveform cycles that are to be written in Waveform Memory.

Syntax: TRIGger[1](2):SEQuence[1](2):COUNt<sp><numeric>

Numeric Parameter: <numeric> ..... Number of times to repeat the trigger sequence(s).

Notes:
1. The "COUNt" must be a decimal integer.
2. The designation "[1](2)" following the keyword, TRIGger, is used to select the channel being programmed.
3. Always start a trigger sequence by sending the command "INITiate". It can be stopped with the "ABORt" command if TRIG:COUN = 0.
4. This command must be sent between the STARt and STOP commands.
5. The total number of points generated by the trigger sequence(s) must be less than or equal to 32K words, the maximum number of points that each DAC Waveform Memory can hold.

Example: The following Trigger Sequence continuously generates 6 cycles of a 5.0-kHz sine wave followed by 10 cycles of a 10-kHz sine wave.

```
TRIG1:SEQ1:STAR
SOUR1:FUNC SIN;VOLT 8.5;FREQ 5000
OUTP1:FILT ON;STAT ON
TRIG1:SEQ1:COUN 6
TRIG1:SEQ1:STOP
TRIG1:SEQ2:STAR
SOUR1:FUNC SIN;VOLT 5.0;FREQ 10000
TRIG1:SEQ2:COUN 10
TRIG1:SEQ2:STOP
INIT1
```

Related Commands: INITiate, ABORt

Reset Condition: Erases all TRIGger:COUNt.

NOTE: In order for Trigger Sequences to run correctly, the DBS 8750 must be equipped with firmware revision 2.0 or higher. To determine hardware/software revision levels, refer to the *IDN? command in Section 4.

TRIGger:COUNt

Description: Specifies how many times to repeat the Trigger Sequence(s) before it returns to the previously programmed waveform. This determines the number of times the program loops through Waveform Memory.

Syntax: TRIGger[1](2):COUNt<sp><numeric>

Numeric Parameter: <numeric> ..... Number of times to repeat all trigger sequences, where 0 = Free Running Mode.

Notes:
1. Always start a trigger sequence by sending the command "INITiate". It can be stopped with the "ABORt" command if COUN = 0.
2. The designation "[1](2)" following the keyword, TRIGger, is used to select the channel being programmed.
3. If "TRIGger:COUNT" is '0' or omitted, the trigger sequences are repeated continuously. If the number of counts does not equal "0", the trigger sequences will execute the specified number of times after "INIT", then the channel will return to the waveform being generated before the "INITiate" command.

Example: The following list of commands generates 6 cycles of a 5.0-kHz, 8.5-volt Sine wave and 10 cycles of a 10-kHz, 5-volt Sine wave. The entire ouput pattern is repeated15 times and then the output returns to 0 Volts.

```
SOUR1:FUNC DC;VOLT 0
OUTP1:STAT ON;FILT ON
TRIG1:SEQ1:STAR
SOUR1:VOLT 8.5;FREQ 5000
TRIG1:SEQ1:COUN 6;STOP
TRIG1:SEQ2:STAR
SOUR1:FUNC SIN;VOLT 5.0;FREQ 10000
TRIG1:SEQ2:COUN 10
TRIG1:SEQ2:STOP
TRIG1:COUNT 15
INIT1
```

Related Commands: INITiate, ABORt

Reset Condition: Erases all TRIGger:COUNt.

TRIGger:SOURce EXT

Description: Selects the Front Panel TRIG input. The TRIG input signal is synchronized with the internal 16-MHz clock and may be used to initiate trigger sequence operation.

Syntax: TRIGger[1](2):SOURce<sp>EXT

Example: To set the external trigger mode: TRIG:SOUR EXT

Related Commands: None

Reset Condition: Sets trigger mode to internal.

TRIGger:SOUR INT

Description: Selects the internal trigger mode.

Syntax: TRIGger[1](2):SOURce<sp>INT (Query: TRIG:SOUR?)

Note: The Internal Trigger mode initiates trigger sequence operation using the Internal Precision Clock.

Example: To set the internal trigger mode: TRIG:SOUR:INT

Related Commands: None.

Reset Condition: Sets trigger mode to internal.

INITiate

Description: Arms a trigger sequence.

Syntax: INITiate[1](2)

Notes:
1. INITiate does not affect any other DBS 8750 setting.
2. The designation "[1](2)" following the keyword, INITiate, is used to select the channel being programmed.
3. If ABORt stops the TRIGger sequence, the next INITiate uses analog output control conditions set by the most recent OUTPut commands.
4. Once INITiate activates the TRIGger SEQuence, subsequent commands are not accepted (except ABORt and *RST) until a terminating condition occurs (ABORT received, or TRIG:COUNt expires).

Example: Initiating a trigger sequence: INIT

Related Commands: TRIGger:SEQuence, ABORt

Reset Condition: Erases all pre-programmed TRIGger SEQuences.

ABORt

Description: ABORt stops execution of the trigger sequence, saves trigger sequence information, and returns to generating the waveform which was set before the sequence started. To restart sequence, send INITiate again. ABORt affects only the Trigger Subsystem.

Syntax: ABORt[1](2)

Notes:
1. After ABORt is executed, the output control conditions (output state, filter, drive, attenuation, etc.) stay as set by the TRIGger sequence commands.
2. The designation "[1](2)" following the keyword, ABORt, is used to select the channel being programmed.
3. ABORt does not affect any other setting in the DBS 8750.

Example: Aborting a trigger sequence: ABOR

Related Commands: TRIGger:SEQuence, INITiate

Reset Condition: Erases all pre programmed TRIGger SEQuences

*CLS

Description: Clear Status – clears all registers in the instrument.

Syntax: *CLS

Related Commands: None

Reset Condition: No effect.

*ESE

Description: Event Status Enable – sets the DBS 8750 Event Status Enable register to a number between 0 and 255. *ESE? returns the previously set value. The event status register of the DBS 8750 has no operational meaning for the DBS 8750.

Syntax: *ESE<0-255>

Related Commands: None

Reset Condition: The Event Status Enable register is '0'.

*ESR?

Description: Event Status Register Query – returns the value of the Event Status register. This register has no operational meaning to the DBS 8750.

Syntax: *ESR

Reset Condition: The Event status register is set to 129 (Power on | Operation Complete).

Related Commands: *CLS sets the Event Status register to 0.

*IDN?

Description: Identification Query – returns the manufacturer, model number, and revision levels of the DBS 8750 in the following format:

Analogic, DBS8750, <Hardware Rev>, <Fimware Rev>

Syntax: *IDN

Related Commands: None

*OPC

Description: Operation Complete – The *OPC command has no effect on the DBS 8750, since the instrument is strictly a sequential system. *OPC? always returns '1'.

Syntax: *OPC

Related Commands: *WAI

Reset Condition: Operation complete is '1'.

*RST

Description: Reset – executes the equivalent of a power-on or front panel reset. All diagnostic tests are executed and the instrument returns to it's quiescent state.

Syntax: *RST

Related Commands: None

Note: Command is ignored during waveform calculation.

*SRE

Description: Service Request Enable – sets the service request enable register to the value in the command. *SRE? returns the value previously set. The service request enable register in the DBS 8750 is of no operational significance.

Syntax: *SRE<0-255>

Related Commands: None.

Reset Condition: SRE register is '0'.

*STB?

Description: Status Byte – returns the value of the instrument status byte. The status byte has no operational meaning to the DBS 8750.

Syntax: *STB?

Related Commands: None

Reset Condition: The status byte is '0'.

*TST?

Description: Self-Test – executes all diagnostic tests, except the SRAM diagnostic, and returns a single ASCII character indicating self-test results.

Returned Characters and their meaning:

0 ..... No Diagnostic Errors
1 ..... SRAM self-test failed.
2 ..... DAC RAM bank 0 self-test failed.
3 ..... DAC RAM bank 1 self-test failed.
4 ..... DAC control register self-test failed.
5 ..... DAC rate register self-test failed.
6 ..... DAC 1 address register self-test failed.
7 ..... DAC 2 address register self-test failed.
8 ..... VXI offset register self-test failed.
9 ..... VXI control register self-test failed.
: (colon) ..... EPROM checksum self-test failed.
; (semi-colon) ..... DAC output self-test failed.

Syntax: *TST?

Related Commands: None

*WAI

Description: The *WAI command has no operational effect on the DBS 8750.

Syntax: *WAI

Related Commands: *OPC, *OPC?

Appendix A
Specifications

Unless otherwise noted, all specifications apply to an instrument configured with the factory jumper selections at 25 °C.

Analog Outputs

Number of Channels	2
Output Voltage	± 10V Single-Ended; ± 5V Differential (Relative to analog ground.)
Resolution	16 Bit
Offset (programmable)	+10V to –10V (in 1 LSB steps, min.)
Attenuation (programmable)	Coarse: 0 dB to -30 dB in –10 dB step Fine: ±0.003%, FSR
Gain (filter bypass)	1 ± 0.5%
Differential Non-Linearity	±0.003% of FSR, max.
Output Drive (programmable)	Single-Ended or Differential
Output Impedance	50 Ohm or 600 Ohm (switch-selectable)
Output Coupling	DC or AC (switch-selectable)
Maximum Load Current	40 mA

Standard Waveform Outputs

Sine Wave Frequency	12.5 Hz to 20 kHz
Square/Triangle Wave Frequency	12.5 Hz to 30 kHz
Peak Distortion (PD)	–96 dB @ 1.0kHz; –93 dB @ 20kHz [PD represents the ratio between the highest spurious frequency component below the Nyquist rate and the test signal.]
Total Harmonic Distortion (THD)	–94 dB @ 1.0kHz; –90 dB @ 20kHz [THD represents the ratio between the rms sum of all harmonics up to the 20th harmonic and the rms value of the test signal.]
Settling Time	16 us (to 1 LSB after 10V step), output filter bypassed, deglitching filter in line.
Offset Voltage	±20 mV, typ.

Filters

Filter Type	6-pole Butterworth
Modes of Operation	ON or bypass (programmable)
Pass Band Frequency	25 kHz (3dB point)
Flatness in Pass Band	±0.1dB to 10 kHz; ±0.25dB to 20 kHz, max.
Attenuation in Stop Band	–90 dB
Settling Time	250µs to 16-bit accuracy, max. overshoot >2.5V (For a 20-volt step function with Filter ON)

Stability(0 C to +50 C)

Required Warm-up Time	15 minutes (for ultimate specifications)
Offset Tempco	±23 µV/°C, typ.; ±75 µV/°C, max.
Gain Tempco	±40 ppm FSR/ °C Max.
Differential Non-Linearity Tempco	±0.3 ppm FSR/°C Max

Memory

DAC Waveform Memory	32K words/channel (maximum waveform size)
DSP/User Memory	32K longwords, SRAM (17150 words read/write accessible to user)

Triggering and Synchronization

Front Panel CLOCK Input	TTL level, active low, 400 kHz maximum frequency, minimum pulse width 120 ns
Front Panel TRIG input	TTL level, active low level
Front Panel GATE Output	TTL level, active low during waveform generation, programmable delay of 20 µs to 41.94 sec.
TTLTRG Input Sampling Rate Clock	TTLTRG0-7, software selected
TTLTRG Output Sampling Clock	TTLTRG0-3, jumper selected, software enabled
TTLTRG GATE Output	TTLTRG4-7, jumper selected, software enabled
Precision Clock Accuracy	16 MHz ± 0.005%

DSP Processor

TMS320C30 @ 33 MHz

Front Panel LED Indicator

Self-Test Pass/Fail	Green = Pass; Red = Fail

Power Required

+ 5V Supply	+4.75V (min.), +5.25V (max.)
Power Consumption	20W, max.

Environmental and Mechanical

Rated Performance Temperature	0 °C to 50 °C
Storage Temperature	25 °C to +75 °C
Relative Humidity	0 to 85% non-condensing up to +40 °C
Recommended Forced Air Cooling	10 cubic feet per minute
Dimensions	"C" size VXI

VXIbus Compliance

Addressing Capabilities	Master: A244/A32; Slave: A16/A24
Data Transfer Capabilities	Master: D16; Slave: D16
Interrupt Levels	1 to 7, programmable
VXI Configuration Registers	ID Register, Device Type Register, Control/Status Register, and Offset Register
Communication Registers	Protocol Register, Response Register, and Data Low Register

Appendix B
VXIbus Connections

Figure B-1. Connector P1

	A	B	C
1	VDB0	VBBSYL	VDB8
2	VDB1	VBCLRL	VDB9
3	VDB2	VACFAIL	VDB10
4	VDB3	VBG0INL	VDB11
5	VDB4	VBG0OUTL	VDB12
6	VDB5	VBG1INL	VDB13
7	VDB6	VBG1OUTL	VDB14
8	VDB7	VBG2INL	VDB15
9	GND	VBG2OUTL	GND
10	VSYSCLK	VBG3INL	VSYSFAIL
11	GND	VBG3OUTL	VBERRL
12	VDSL1	VBR0L	VSYSRSTL
13	VDSL0	VBR1L	VLWORDL
14	VWRITEL	VBR2L	VAM5
15	GND	VBR3L	VA23
16	VDTACKL	VAM0	VA22
17	GND	VAM1	VA21
18	VASL	VAM2	VA20
19	GND	VAM3	VA19
20	VIACKL	GND	VA18
21	VIACKINL	NOT USED	VA17
22	VIACKOUT	NOT USED	VA16
23	VAM4	GND	VA15
24	VA7	VIRQL7	VA14
25	VA6	VIRQL6	VA13
26	VA5	VIRQL5	VA12
27	VA4	VIRQL4	VA11
28	VA3	VIRQL3	VA10
29	VA2	VIRQL2	VA9
30	VA1	VIRQL1	VA8
31	NOT USED	NOT USED	NOT USED
32	+ 5VD	+ 5VD	+ 5VD

Figure B-2. Connector P2

	A	B	C
1	NOT USED	+ 5VD	NOT USED
2	NOT USED	GND	NOT USED
3	NOT USED	NOT USED	GND
4	GND	NOT USED	NOT USED
5	NOT USED	NOT USED	LBUSC0
6	NOT USED	NOT USED	LBUSC1
7	NOT USED	NOT USED	GND
8	NOT USED	NOT USED	LBUSC2
9	NOT USED	NOT USED	LBUSC3
10	GND	NOT USED	GND
11	NOT USED	NOT USED	LBUSC4
12	NOT USED	GND	LBUSC5
13	NOT USED	+ 5VD	NOT USED
14	NOT USED	NOT USED	LBUSC6
15	NOT USED	NOT USED	LBUSC7
16	GND	NOT USED	GND
17	NOT USED	NOT USED	LDHI
18	NOT USED	NOT USED	LDLO
19	NOT USED	NOT USED	NOT USED
20	NOT USED	NOT USED	NOT USED
21	NOT USED	NOT USED	NOT USED
22	GND	GND	GND
23	TTLTRG0	NOT USED	TTLTRG1
24	TTLTRG2	NOT USED	TTLTRG3
25	NOT USED	NOT USED	GND
26	TTLTRG4	NOT USED	TTLTRG5
27	TTLTRG6	NOT USED	TTLTRG7
28	GND	NOT USED	GND
29	NOT USED	NOT USED	NOT USED
30	MODID	NOT USED	GND
31	GND	GND	NOT USED
32	NOT USED	+ 5VD	NOT USED

Appendix C
Error Messages

OPERATIONAL ERROR MESSAGES

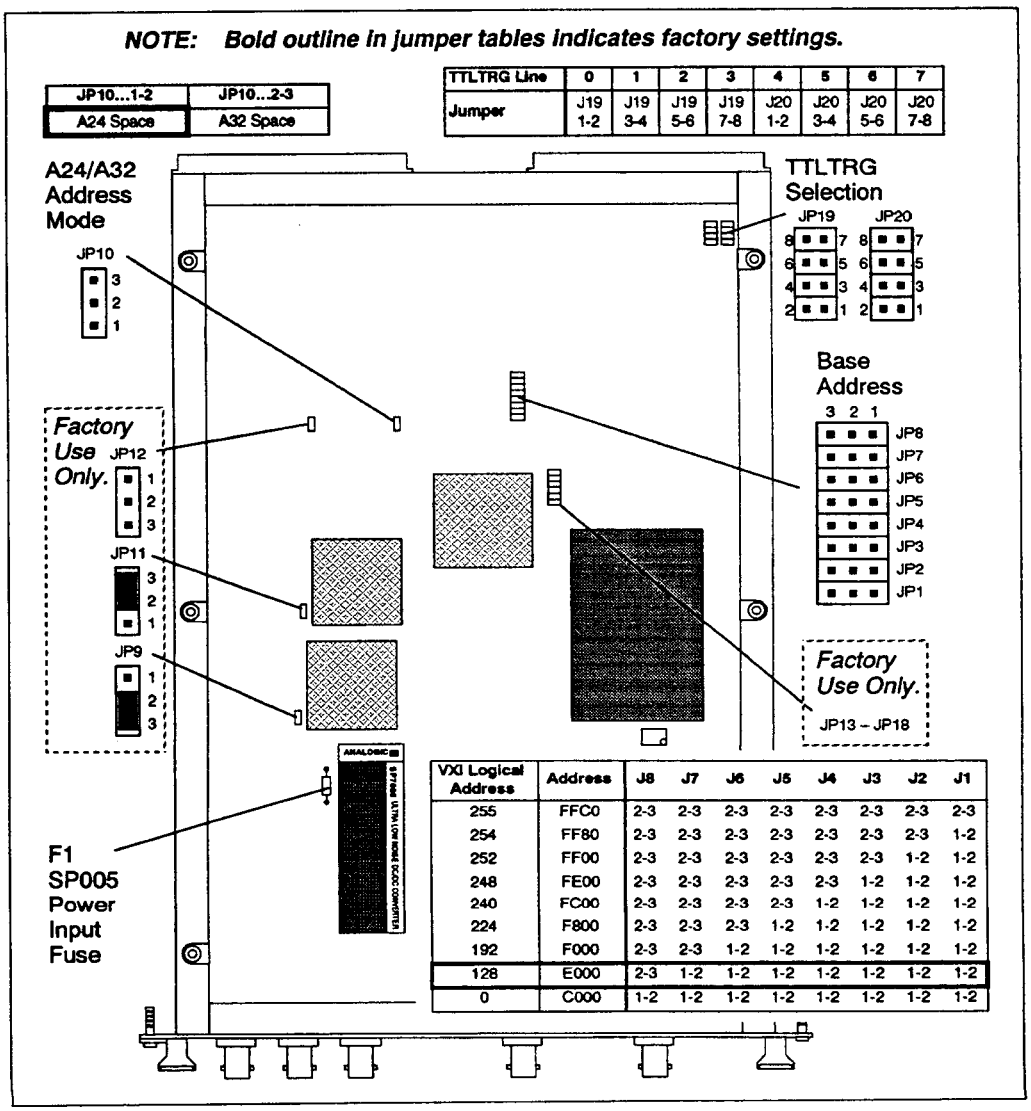
0, "No error"
–100, "Command error; command not implemented"
–100, "Command error; missing right parentheses"
–100, "Command error; missing left parentheses"
–100, "Command error; expecting waveform directive FOR"
–100, "Command error; incomplete waveform specification"
–100, "Command error; duration field must evaluate to a constant"
–102, "Syntax error"
–200, "Execution error; DMA controller is already in use"
–200, "Execution error; no buffers defined"
–200, "Execution error; buffer not found"
–200, "Execution error; user buffer too small for requested transfer size"
–210, "Trigger error; trigger system is currently active"
–210, "Trigger error; invalid trigger sequence definition"
–210, "Trigger error; no more trigger sequences available"
–210, "Trigger error; no user function defined"
–210, "Trigger error; trigger sequence too long"
–210, "Trigger error; no trigger sequences defined"
–210, "Trigger error; user functions are invalid in trigger sequences"
–222, "Data out of range"
–225, "Out of memory; cannot allocate memory for buffer"
–225, "Out of memory; cannot resize the buffer"
–225, "Out of memory; not enough waveform memory for requested signal"
–255, "Directory Full"
–310, "System error; VME BUSERR – DMA Transfer Aborted"

DIAGNOSTIC TEST ERROR MESSAGES

After a diagnostic failure or at any time, the diagnostic message status can be obtained by sending the query command, DIAGnostic?

0, "No error; Self-test PASSED"
–330, "Self-test failed; Static RAM failed self-test"
–330, "Self-test failed; DAC RAM 0 memory error"
–330, "Self-test failed; DAC RAM 1 memory error"
–330, "Self-test failed; DAC control register error"
–330, "Self-test failed; DAC rate register error"
–330, "Self-test failed; DAC 0 address register error"
–330, "Self-test failed; DAC 1 address register error"
–330, "Self-test failed; VME offset register error"
–330, "Self-test failed; VME control register error"
–330, "Self-test failed; EPROM memory error"
–330, "Self-test failed; DAC analog output error"

Appendix D
Revision 0 Hardware



Selecting TTL Trigger Lines

The TTL trigger lines can be used for additional triggering and synchronization capabilities. The selected Sampling Rate Clock can be jumper-selected to drive one of the eight TTLTRG lines. Before the selected clock signal can drive the bus, the TTLTRG line drivers must first be enabled by command OUTPut:TTLTrg ON | OFF.

Also, an External Sampling Rate Clock input can be received on any unused TTLTRG line. The input clock is software selected using the command, SOURce:ROSCillator:SOURce TTLTRG <0-7>.

TTLTRG Interface – Revision 0 Hardware