User Manual

VX4792
Arbitrary Waveform Generator

070-8959-05

This document applies to instruments serial number J330101 and above.

Printed in Japan.

Sony/Tektronix Corporation, P.O.Box 5209, Tokyo Int'l, Tokyo 100–31 Japan

Tektronix, Inc., P.O. Box 1000, Wilsonville, OR 97070–1000

WARRANTY

Tektronix warrants that this product will be free from defects in materials and workmanship for a period of three (3) years from the date of shipment. If any such product proves defective during this warranty period, Tektronix, at its option, either will repair the defective product without charge for parts and labor, or will provide a replacement in exchange for the defective product.

In order to obtain service under this warranty, Customer must notify Tektronix of the defect before the expiration of the warranty period and make suitable arrangements for the performance of service. Customer shall be responsible for packaging and shipping the defective product to the service center designated by Tektronix, with shipping charges prepaid. Tektronix shall pay for the return of the product to Customer if the shipment is to a location within the country in which the Tektronix service center is located. Customer shall be responsible for paying all shipping charges, duties, taxes, and any other charges for products returned to any other locations.

This warranty shall not apply to any defect, failure or damage caused by improper use or improper or inadequate maintenance and care. Tektronix shall not be obligated to furnish service under this warranty a) to repair damage resulting from attempts by personnel other than Tektronix representatives to install, repair or service the product; b) to repair damage resulting from improper use or connection to incompatible equipment; or c) to service a product that has been modified or integrated with other products when the effect of such modification or integration increases the time or difficulty of servicing the product.

THIS WARRANTY IS GIVEN BY TEKTRONIX WITH RESPECT TO THIS PRODUCT IN LIEU OF ANY OTHER WARRANTIES, EXPRESSED OR IMPLIED. TEKTRONIX AND ITS VENDORS DISCLAIM ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. TEKTRONIX' RESPONSIBILITY TO REPAIR OR REPLACE DEFECTIVE PRODUCTS IS THE SOLE AND EXCLUSIVE REMEDY PROVIDED TO THE CUSTOMER FOR BREACH OF THIS WARRANTY. TEKTRONIX AND ITS VENDORS WILL NOT BE LIABLE FOR ANY INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES IRRESPECTIVE OF WHETHER TEKTRONIX OR THE VENDOR HAS ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Table of Contents

List of Figures iv
List of Tables vii
General Safety Summary ix
Preface xiii

Getting Started

Product Description 1–1
Accessories 1–2
Applications Support 1–2
Configuration and Installation 1–3
Setting the Logical Address 1–4
Preparing the VXIbus Mainframe 1–6
VX4792 Installation 1–7
VX4792 Removal 1–9
Functional Check 1–11
Initial Setup 1–11
Functional Check Procedure 1–11

Operating Basics

Functional Overview 2–1
Connectors and Indicators 2–2
Generating Waveforms 2–3
Operating Modes 2–5
Sequence Control 2–8
Waveform Memory 2–10
Data Length 2–11
Setting Clock Source and Frequency 2–12
Setting Amplitude and Offset 2–13
Setting Output Filter 2–14
Time Delay Due to Filters 2–15
Setting Trigger Parameters for an External Trigger 2–15
Waveform Timing 2–16
External AM Operation 2–17
Setting Sync Signal 2–18
Equations 2–18
Diagnostics and Calibration 2–29
Instrument I/O 2–33
Programming Examples 2–39
Preliminary Information 2–39
Instrument Control and Basic Waveform Generation 2–40
Synchronous Operation 2–41
Defining Equations 2–43
Compiling Equations 2–44
Transferring Waveforms 2–45

Syntax and Commands

Command Syntax 3–1
Command Notation 3–1
Program and Response Messages 3–1
Command and Query Structure 3–2
Character Encoding 3–2
Syntactic Delimiters 3–3
White Space 3–3
Special Characters 3–3
Arguments 3–4
Header 3–6
Concatenating Commands 3–8
Query Responses 3–9
Other General Command Conventions 3–10
Syntax Diagrams 3–10

Command Groups 3–13
Commands Grouped by Function 3–13
Command Quick Reference 3–14
Command Summaries 3–15

Command Descriptions 3–21

Status and Events

Status and Event Reporting 4–1
Registers 4–1
Queues 4–5
Processing Sequence 4–6

Appendices

Appendix A: ASCII & GPIB Code Chart A–1

Appendix B: Messages B–1

Appendix C: Default Settings C–1

Appendix D: Sample Waveform Library D–1
Sine Wave Signal D–3
Triangle Wave Signal D–4
Ramp Signal D–5
Square Wave Signal D–6
Gaussian Pulse D–7
Lorentz Pulse D–8
Sin(x)/x Pulse D–9
Squared Sine Pulse D–10
Rising, Falling Exponential Pulse D–11
Nyquist Pulse D–12
Linear Frequency Sweep D–13
Logarithmic Frequency Sweep D–14
Amplitude Modulated Signal D–15
Frequency Modulated Signal D–16
Damped Sine Wave D–17
Pulse Width Modulated Signal D–18

NRZ Pseudo-Random (PRBS) Signal D–19
Electromagnetic Disk Signal 1 . . . . . D–20
Electromagnetic Disk Signal 2 . . . . . D–21
NTSC Directory D–22

Appendix E: Specifications E–1
Introduction E–1
Performance Conditions E–1

Appendix F: Performance Verification F–1
Levels of Testing F–1
Test Interval F–2
Conventions F–2
Prerequisites F–3
Equipment Required F–3
Performance Check Disk Waveform Files F–4
System Setup F–8
Operating Mode Checks F–11
External AM Operation Checks F–22
Clock Frequency and Amplitude Checks F–25
Amplitude and Offset Accuracy Checks F–28
Pulse Response Check F–30
SYNC OUTPUT and MARKER OUTPUT Amplitude Checks F–33
External Trigger Level Accuracy Check F–35
External CLOCK INPUT Check F–37
Synchronous Operation Check F–39

Appendix G: Functions G–1
Differentiation G–1
Integration G–3
Random (rnd) Function G–4

Glossary and Index

List of Figures

Figure 1–1: Setting the Logical Address 1–5
Figure 1–2: Module Retainer Screws and Ejector Mechanism 1–8
Figure 2–1: Front Panel Connectors and Indicators 2–3
Figure 2–2: External Trigger Signal and Waveform Output in Triggered Mode 2–5
Figure 2–3: External Gate Signal and Waveform Output in Gated Mode 2–6
Figure 2–4: Waveform Output in Burst Mode 2–6
Figure 2–5: Sequence Output in Waveform Mode 2–7
Figure 2–6: Sequence Output in Autostep Mode 2–8
Figure 2–7: Relationship Between Sequence Memory and Waveform Memory 2–9
Figure 2–8: Waveform Memory Organization 2–10
Figure 2–9: Triangular Wave Resolution and Number of Data Points 2–11
Figure 2–10: Waveform Points and Sync Output in Continuous Mode 2–12
Figure 2–11: Clock and Waveform Points 2–13
Figure 2–12: Amplitude and Offset Setting 2–13
Figure 2–13: Representative Filter Characteristics 2–14
Figure 2–14: Slope and Level Controls 2–15
Figure 2–15: Trigger Input, Sync Output, and Waveform Timing . . 2–16
Figure 2–16: External AM Operation (Waveform Multiplication) . . 2–17
Figure 2–17: Example of Equation File Data and Resulting Waveform 2–19
Figure 2–18: Trigonometric Function Waveform Expressed with Variable x 2–22
Figure 2–19: Trigonometric Function Waveform Expressed with Variable t 2–22
Figure 2–20: Equation Using Exp(. . . . 2–23
Figure 2–21: Equation Using Log(. . . . 2–23
Figure 2–22: Equation Using Ln(. . . . 2–24
Figure 2–23: Equation Using Sqrt(. . . . 2–24
Figure 2–24: Equation Using Abs(. . . . 2–25
Figure 2–25: Equation Using Int(. . . . 2–25
Figure 2–26: Equation Using Round(. . . . 2–26
Figure 2–27: Equation Using Norm(. . . . 2–26

Figure 2–28: Equation Using Min(and Max(. . . . 2–27
Figure 2–29: Equation Using Rnd(. . . . 2–27
Figure 2–30: Waveform Before Calculating Differentiation 2–28
Figure 2–31: Waveform After Differentiation Using Diff(. 2–28
Figure 2–32: Waveform After Integration Using Integ(. 2–29
Figure 2–33: VXIbus Connectors P1 and P2 2–33
Figure 3–1: Command and Query Structure Flowchart 3–2
Figure 3–2: Typical Syntax Diagrams 3–10
Figure 4–1: The Standard Event Status Register (SESR) 4–1
Figure 4–2: The Status Byte Register (SBR) 4–3
Figure 4–3: The Device Event Status Enable Register (DESER) 4–4
Figure 4–4: The Event Status Enable Register (ESER) 4–4
Figure 4–5: The Service Request Enable Register (SRER) 4–5
Figure 4–6: Status and Event Handling Process Overview 4–7
Figure D–1: Sine Wave Signal D–3
Figure D–2: Triangle Wave Signal D–4
Figure D–3: Ramp Signal D–5
Figure D–4: Square Wave Signal D–6
Figure D–5: Gaussian Pulse D–7
Figure D–6: Lorentz Pulse D–8
Figure D–7: Sin(x)/x Pulse D–9
Figure D–8: Squared Sine Pulse D–10
Figure D–9: Rising, Falling Exponential Pulse D–11
Figure D–10: Nyquist Pulse Waveform D–12
Figure D–11: Linear Frequency Sweep Signal D–13
Figure D–12: Logarithmic Frequency Sweep Signal D–14
Figure D–13: Amplitude Modulated Signal D–15
Figure D–14: Frequency Modulated Signal D–16
Figure D–15: Damped Sine Wave D–17
Figure D–16: Pulse Width Modulated Signal D–18
Figure D–17: NRZ Pseudo-random Signal D–19
Figure D–18: Pseudo-random Signals Generated with Shift Register D–19
Figure D–19: NRZ Pseudo-random Signal D–19
Figure D–20: Electromagnetic Disk Signal 1 Waveform D–20
Figure D–21: Electromagnetic Disk Signal 2 Waveform D–21
Figure D–22: NTSC Directory Signals D–23
Figure F–1: Setting Switches on the Resource Manager F–9
Figure F–2: Basic System Configuration F–10

Figure F–3: Cont Mode Initial Test Setup F–11
Figure F–4: Triggered Mode Initial Test Setup F–13
Figure F–5: Burst Mode Initial Test Setup F–15
Figure F–6: Gated Mode Initial Test Setup F–17
Figure F–7: Waveform Output Signal with Positive Gate Signal F–18
Figure F–8: Waveform Output Signal with Negative Gate Signal . . . F–19
Figure F–9: Waveform Advance Mode Initial Test Setup F–20
Figure F–10: Autostep Mode Initial Test Setup F–21
Figure F–11: External AM Operation Initial Test Setup F–23
Figure F–12: Clock Frequency Accuracy Initial Test Setup F–25
Figure F–13: Clock Amplitude Initial Test Setup F–27
Figure F–14: Gain Accuracy Initial Test Setup F–29
Figure F–15: Pulse Response Initial Test Setup F–31
Figure F–16: Sync Output and
Marker Output Amplitude Initial Test Setup F–33
Figure F–17: External Trigger Level Accuracy Initial Test Setup . . . F–35
Figure F–18: External CLOCK INPUT Initial Test Setup F–37
Figure F–19: Synchronous Operation Initial Test Setup F–40
Figure F–20: Skew Between Modules F–42
Figure G–1: Equation Differentiation G–2
Figure G–2: Equation Integration G–3

List of Tables

Table 1–1: VX4792 Arbitrary Waveform Generator Installation Record 1–8
Table 2–1: Types of Waveform Files 2–4
Table 2–2: An Example Sequence File (XXX.SEQ) 2–8
Table 2–3: Time Delay Caused By Filters 2–15
Table 2–4: Output Signal Amplitude Relative to External Modulation Signal 2–17
Table 2–5: Components for Assembling Equations 2–20
Table 2–6: Diagnostic and Calibration Commands 2–30
Table 2–7: Left Slot P1 Pinout 2–34
Table 2–8: Left Slot P2 Pinout 2–35
Table 2–9: Right Slot P1 Pinout 2–36
Table 2–10: Right Slot P2 Pinout 2–37
Table 2–11: Trigger Requirements for Synchronous Output 2–41
Table 3–1: BNF Symbols and Meanings 3–1
Table 3–2: Decimal Numeric Notation 3–4
Table 3–3: Header in Query Responses 3–9
Table 3–4: Function Groups in the Command Set 3–13
Table 3–5: Calibration and Diagnostic Commands 3–15
Table 3–6: Memory Commands 3–15
Table 3–7: Mode Commands 3–16
Table 3–8: Output Commands 3–17
Table 3–9: Setup Commands 3–17
Table 3–10: Status and Event Commands 3–18
Table 3–11: Synchronization Commands 3–18
Table 3–12: Synchronization Commands 3–19
Table 3–13: Waveform Commands 3–20
Table 4–1: SESR Bit Functions 4–2
Table 4–2: SBR Bit Functions 4–3
Table B–1: Normal Condition B–1
Table B–2: Command Errors (CME Bit:5) B–2
Table B–3: Execution Errors (EXE Bit:4) B–3
Table B–4: Internal Errors (DDE Bit:3) B–5
Table B–5: System Event B–5
Table B–6: Warnings (EXE Bit:4) B–6

Table B–7: Internal Warnings (DDE Bit:3) **B–6**
Table B–8: Device-Specific Messages **B–6**
Table C–1: Factory Initialized Settings **C–1**
Table D–1: Waveform Samples **D–1**
Table D–2: NTSC Directory **D–2**
Table E–1: Electrical Characteristics **E–2**
Table E–2: Environmental Characteristics **E–7**
Table E–3: Mechanical Characteristics **E–7**
Table F–1: Test Equipment Requirements **F–3**
Table F–2: VX4792 Performance Check Disk Waveform File Summary **F–4**
Table F–3: VX4792 Performance Check Disk Command File Summary **F–5**
Table F–4: VX4792 Device Characteristics Parameters **F–10**
Table F–5: Clock Frequency Settings **F–26**
Table F–6: Pulse Response Characteristics for 1 V Amplitude **F–32**

General Safety Summary

Review the following safety precautions to avoid injury and prevent damage to this product or any products connected to it.

Injury Precautions

Use Proper Power Cord

To avoid fire hazard, use only the power cord specified for the mainframe.

Avoid Electric Overload

To avoid electric shock or fire hazard, do not apply a voltage to a terminal that is outside the range specified for the mainframe.

Ground the Product

This product is grounded through the grounding conductor of the power cord of the mainframe. To avoid electric shock, the grounding conductor must be connected to earth ground. Before making connections to the input or output terminals of the product, ensure that the product is properly grounded.

Do Not Operate Without Covers

To avoid electric shock or fire hazard, do not operate this product with covers or panels removed.

Use Proper Fuse

To avoid fire hazard, use only the fuse type and rating specified for this product.

Do Not Operate in Wet/Damp Conditions

To avoid electric shock, do not operate this product in wet or damp conditions.

Do Not Operate in Explosive Atmosphere

To avoid injury or fire hazard, do not operate this product in an explosive atmosphere.

Product Damage Precautions

Use Proper Power Source

Do not operate this product from a power source that applies more than the voltage specified.

Use Proper Voltage Setting

Before applying power to the mainframe, ensure that the line selector is in the proper position for the power source being used.

Provide Proper Ventilation To prevent product overheating, provide proper ventilation.

Do Not Operate With Suspected Failures If you suspect there is damage to this product, have it inspected by qualified service personnel.

Safety Terms and Symbols

Terms in This Manual These terms may appear in this manual:

WARNING. *Warning statements identify conditions or practices that could result in injury or loss of life.*

CAUTION. *Caution statements identify conditions or practices that could result in damage to this product or other property.*

Terms on the Product These terms may appear on the product:

DANGER indicates an injury hazard immediately accessible as you read the marking.

WARNING indicates an injury hazard not immediately accessible as you read the marking.

CAUTION indicates a hazard to property including the product.

Symbols on the Product The following symbols may appear on the product:

Certifications and Compliances

CSA Certified Power Cords

CSA Certification includes the products and power cords appropriate for use in the North America power network. All other power cords supplied are approved for the country of use.

Safety Certification of Plug-in or VXI Modules

For modules (plug-in or VXI) that are safety certified by Underwriters Laboratories, UL Listing applies only when the module is installed in a UL Listed product. CSA Certification applies only when the module is installed in a CSA Certified product.

Compliances

Consult the product specifications for IEC Installation Category, Pollution Degree, and Safety Class.

Preface

This manual provides the information necessary to install, configure, and operate the VX4792 Arbitrary Waveform Generator.

Manual Organization

This manual consists of the following chapters and appendices:

- *Getting Started* contains a basic product description and information on how to configure and install the waveform generator.
- *Operating Basics* contains a functional overview, instrument I/O information, and descriptions of basic applications.
- *Syntax and Commands* contains syntactic rules, a list of commands by functional group, and full command descriptions in alphabetical order.
- *Status and Events* describes the waveform generator status and event reporting structure.
- The *Appendices* contain an ASCII and GPIB code chart, a list of error messages, descriptions of default settings, a summary of the waveform library disk, detailed specifications, a performance verification procedure, and information about some of the functions used to assemble equations.
- *Glossary/Index* contains a glossary of both instrument-specific and general VXI terms, and a full index.

Notational Conventions

The following notational conventions apply throughout this manual:

Active Signal States

A tilde (~) following a signal mnemonic (as in ACFAIL~) indicates that the signal is active when in its low state.

A signal mnemonic without a tilde indicates that the signal is active when in its high state.

Number Base

Unless otherwise noted, all numbers are assumed to be decimal (base 10).

Numbers expressed in other bases appear with that base attached as a subscript. For example, binary (0110_2), octal (341_8), and hexadecimal ($97FF_{16}$) numbers may appear in this manual.

Related Publications

The following documents provide valuable information on subjects related to the VX4792 Arbitrary Waveform Generator:

- *VXIbus System Specification, Version 1.3, July 14, 1989*
- *VMEbus Specification Manual, Revision C.1, October, 1985*
- *ANSI/IEEE Standard 1014–1987, IEEE Standard for a Versatile Backplane Bus: VMEbus*

Getting Started

Product Description

The VX4792 Arbitrary Waveform Generator is a double-wide, C-size module. Major features include:

- Custom waveforms of up to 250 MS/s (4 ns/point of resolution)
- 12 bits of amplitude resolution
- 256 K words of waveform memory
- Single channel output
- A wide variety of trigger sources and modes
- Output disable
- Four selectable output filters

The waveform memory of the waveform generator is organized as waveform segments. You can set segments to any size (with a step size of 8) from 64 to 256 K 12-bit words. You can also combine sequences of waveform segments into custom waveforms.

The waveform generator has two marker channels. You can individually program each marker to generate marker signals that correspond to any waveform data point.

The waveform generator accepts triggers from VXIbus ECLTRG lines, software, or the front-panel BNC connector.

Software codes and command formats generally conform to IEEE 488.2 and VXIbus System Specification Revision 1.3.

Accessories

The VX4792 Arbitrary Waveform Generator is shipped with the following standard accessories:

- VX4792 User Manual
(Tektronix part number 070-8959-XX)
- VX4792 Performance Check Disk, 3.5 inch
(Tektronix part number 063-1766-XX)
- Sample Waveform Library Disk, 3.5 inch
(Tektronix part number 063-2198-XX)
- Plug & Play Disk, 3.5 inch
(Tektronix part number 063-1527-01)

Applications Support

Contact the following sources for applications support.

VX4792 Arbitrary Waveform Generator

In North America, contact your local sales office or call:

(800) 835-9433, extension 2400

between 8:00 a.m. and 5:00 p.m., Pacific Standard Time.

Outside of North America, call your local Tektronix sales office or distributor.

Microsoft Windows

Contact your Microsoft support center.

Configuration and Installation

The *Configuration and Installation* section explains how to set the logical address, prepare the VXIbus mainframe, and install the VX4792 Arbitrary Waveform Generator into a mainframe. You must complete each of these tasks before putting the waveform generator into service.

The VX4792 Arbitrary Waveform Generator requires two slots in a VXIbus mainframe. This means that you can install up to six modules in a Tektronix VX1410 C-size, 13 slot mainframe.

Within this section you will find instructions to perform the following tasks:

- *Setting the Logical Address* (page 1–4) explains how to set the logical address for operation within your system. You must perform this task before installing the waveform generator into a VXIbus mainframe.
- *Preparing the VXIbus Mainframe* (page 1–6) describes specifications and configuration information you need to consider when selecting a VXIbus mainframe for the waveform generator.
- *VX4792 Installation* (page 1–7) describes how to install the waveform generator into a Tektronix VXIbus mainframe.
- *VX4792 Removal* (page 1–9) describes how to eject the waveform generator from a Tektronix VXIbus mainframe.

Setting the Logical Address

Every module within a VXIbus system must have a unique logical address; no two modules can have the same address. On the VX4792 Arbitrary Waveform Generator, you select the logical address value using a switch located beneath the right cover. The waveform generator has two user-selectable address modes: Dynamic Auto Configuration and static.

Dynamic Auto Configuration Addressing

With Dynamic Auto Configuration address mode (hexadecimal FF_{16} or 255 decimal), the system resource manager automatically sets the address to the next available value within your system. For example, if you have devices set to addresses 01 and 02 already in your system, the resource manager will automatically assign address 03 to the waveform generator at power-on.

Static Addressing

Static address mode sets the address to a fixed value between 0 and 255. This mode ensures that the waveform generator address remains fixed for compatibility with systems that require a specific address value.

Remember that each device within your system must have a unique address to avoid communication problems.

Address Switch Setup

The logical address switch (Figure 1–1) is located beneath the right cover. The 8-section switch has the following characteristics:

- An OPEN switch selects binary 1
- A CLOSED switch selects binary 0
- Switch 1 sets least-significant digit, switch 8 sets most-significant digit

Use the following procedure to set the logical address:

⚠ ***CAUTION.*** *Many components within this instrument are extremely susceptible to static-discharge damage. Remove the instrument covers only in a static-free environment. Observe standard handling precautions for static-sensitive devices. Always wear a grounded wrist strap when handling the waveform generator.*

1. Use a screwdriver with a size T-10 Torx® tip to remove six screws (Figure 1–1), then lift off the right cover.
2. Set the switch address mode and binary value as desired:
   - Set switches 1 through 8 to OPEN for Dynamic Auto Configuration
   - Set switches for a static address between 0 and 255 (decimal)
3. Slide the right cover in place, then install the six screws.

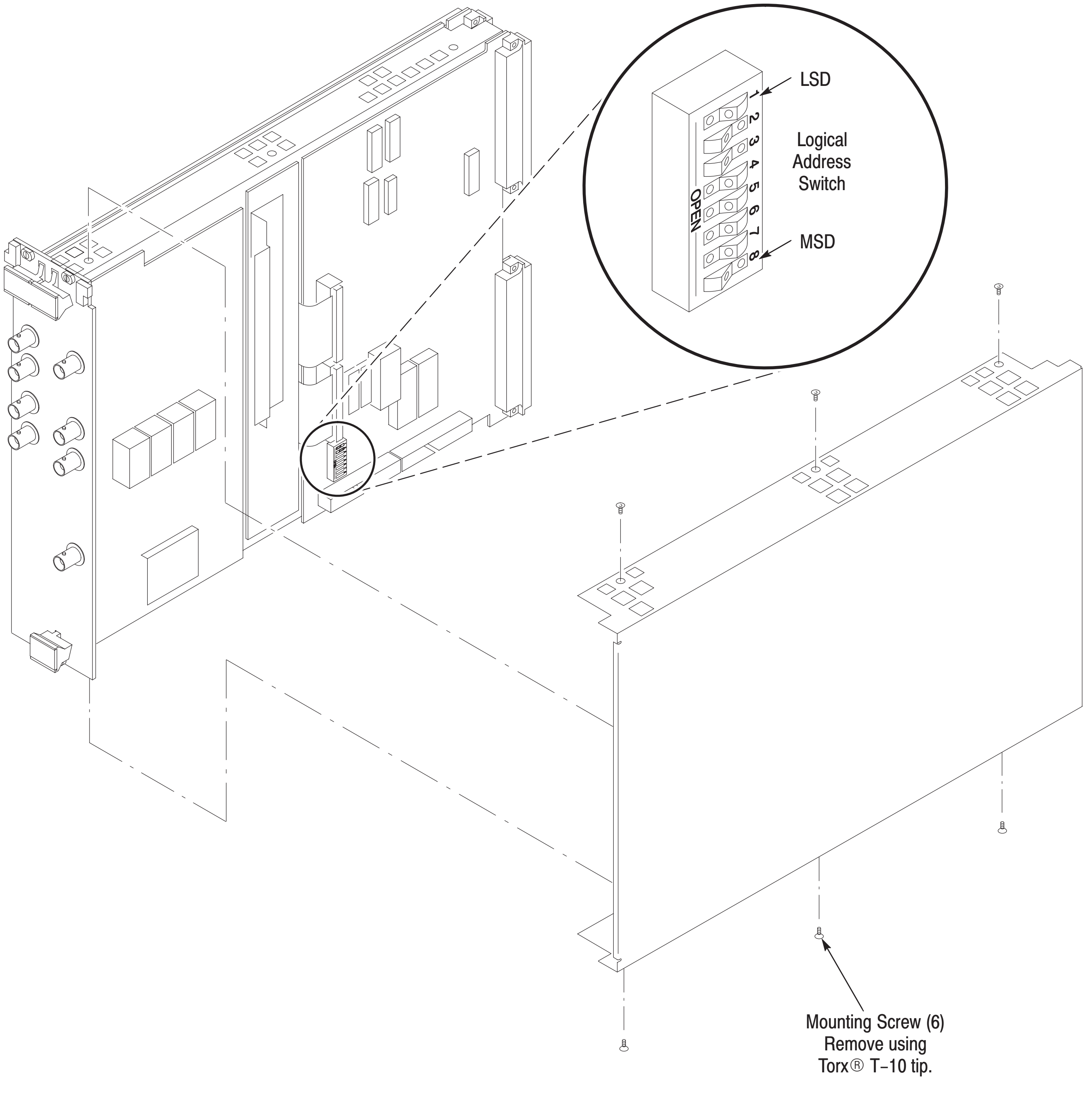


Figure 1-1: Setting the Logical Address

Preparing the VXIbus Mainframe

This section tells you how to install the waveform generator into a Tektronix VXIbus mainframe. If you are installing the VX4792 Arbitrary Waveform Generator into a different mainframe, refer to the instruction manual for that mainframe for any pertinent installation or capacity information.

⚠ ***CAUTION.*** *The waveform generator cooling requirements may exceed the capability of some mainframes. To avoid electrical damage to the waveform generator or mainframe, verify that your mainframe meets the cooling requirements specified in Table E–3 of this manual (see page E–7).*

Voltage, Current, and Cooling Requirements

Table E–1 on page E–2, and Table E–3 on page E–7, list the voltage, current, and cooling requirements for the VX4792 Arbitrary Waveform Generator. (These requirements also appear on the right cover of the module. In these pages of this manual however, additional specification informations are being written.) Be sure your mainframe can supply adequate current and cooling to the VX4792 Arbitrary Waveform Generator and the other modules you plan to install into the same mainframe.

⚠ ***WARNING.*** *Shock hazards exist due to high currents within the mainframe compartment. Do not change configuration of the Bus Grant and Interrupt Acknowledge jumpers unless you are qualified to do so.*

Jumper Settings

VXIbus mainframes contain daisy-chain jumper straps for the Bus Grant (BG0-BG3) and Interrupt Acknowledge (IACK) signals. Tektronix mainframes include all necessary jumpers. If you are using a Tektronix mainframe, the names of the jumper straps (BG0-BG3 and IACK) are often printed on the circuit board facing the front of the mainframe. Access these jumpers from the front of the mainframe.

IACK Jumper. Remove the IACK jumper strap for the right-most slot in which you will install the instrument. Retain the strap for future reconfiguration.

BG0–BG3 Jumpers. Leave the BG0–BG3 jumpers installed in the mainframe.

VX4792 Installation

You may insert the waveform generator into any empty slot in the mainframe except Slot 0. Be sure the logical address is set before installing the waveform generator into the mainframe (see *Setting the Logical Address* on page 1–4).

CAUTION. *If you install the waveform generator into a D-size mainframe, be sure to connect the P1 and P2 connectors of the module to the P1 and P2 connectors on the mainframe. Electrical damage will result when connecting the P1 and P2 connectors on the module to the P2 and P3 connectors on the mainframe.*

To avoid electrical damage, look for bent pins on P1 and P2 before installation.

NOTE. *Each module runs either synchronously (master and slave modules) or asynchronously. For synchronous operation, modules must be adjacent within the mainframe. See page 2–41 for an example of synchronous operation.*

Use the following installation procedure and Figure 1–2 on page 1–8 to install the waveform generator into the mainframe:

1. On the mainframe, ensure that the power ON/STANDBY switch is OFF.
2. Decide into which slots of the mainframe you install the module. The VX4792 is double wide module and requirs two slots in the VXI mainframe.
3. Insert the module into the mainframe top and bottom module guides and push it partially into the mainframe (Figure 1–2). Then slide the module into the mainframe as far as it will go without forcing it.
4. Be sure the front panel is flush with the front of the mainframe chassis. If so, use a flat-bladed screwdriver to install the top and bottom module retainer screws. Alternate between the screws, applying only a few turns at a time to fully seat the module.

After installation is complete, fill out the Installation Record (Table 1–1). Then perform the *Functional Check Procedure* on page 1–11 to verify that the waveform generator is operating properly.

Table 1–1: VX4792 Arbitrary Waveform Generator Installation Record

Item	Write Your Entries Here
VX4792 serial number	
VX4792 firmware version number[1]	
VX4792 logical address switch settings	
Installed in VXIbus mainframe slot numbers	
Installation performed by	
Date of installation	

[1] **Refer to *Functional Check Procedure* (page 1–11) for a method to determine the Firmware Version Number.**

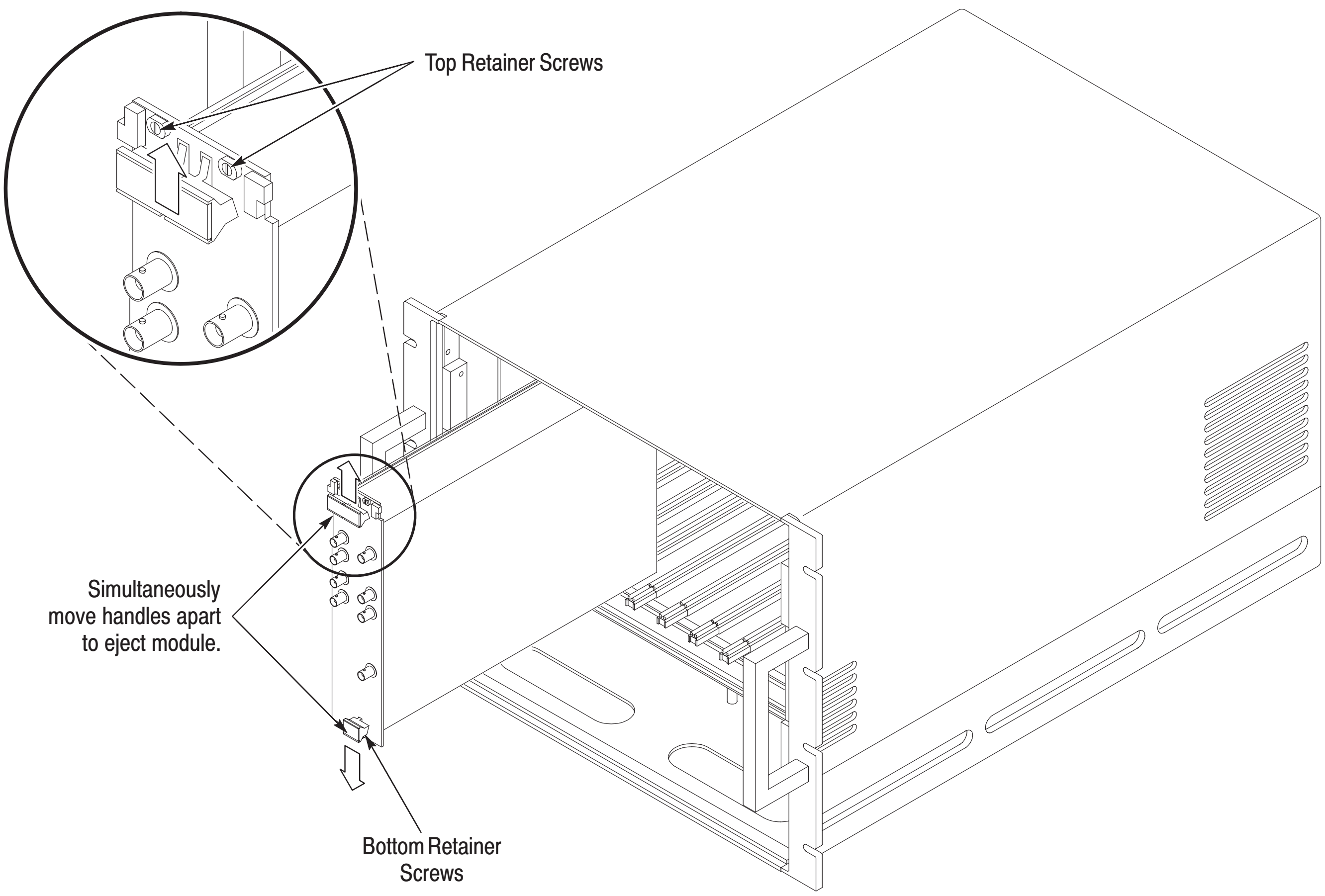


Figure 1–2: Module Retainer Screws and Ejector Mechanism

VX4792 Removal

The following procedure describes module removal from a Tektronix VXIbus mainframe. If you are using a different mainframe, you may need to modify this procedure.

1. On the mainframe, ensure that the power ON/STANDBY switch is set to OFF.
2. Use a flat-bladed screwdriver to loosen the top and bottom retainer screws (Figure 1–2).
3. Grasp both handles of the module. At the same time, move the top handle upward and the bottom handle downward to eject the module.
4. Pull the module out of the mainframe.

Functional Check

The functional check tests the installed module to verify that it is operating properly. Once the VX4792 Arbitrary Waveform Generator is installed into a VXIbus mainframe with an operating computer interface, no additional test equipment is needed to perform the functional check.

This functional check will not verify that the waveform generator is operating within warranted limits. If a more detailed test is required, you may perform the *Performance Verification* procedure found in *Appendix F.*

Initial Setup

Install the waveform generator into a VXIbus mainframe. Refer to *Configuration and Installation* on page 1–3 for instructions. The following *Functional Check Procedure* may be performed as soon as the waveform generator completes the power-on diagnostics.

Functional Check Procedure

NOTE*. This procedure assumes that you are able to communicate with your system using standard talker/listener software for IBM PC- or UNIX-compatible computers with GPIB or other proprietary configurations, or embedded controllers.*

---

1. On the VXIbus mainframe, connect AC power and set the POWER switch to the ON position.

2. Approximately 60 seconds after the power is turned on, verify that only the green POWER indicator is lit.

The waveform generator performs self-calibration and diagnostics at power-on. During this time the POWER indicator blinks on and off.

3. To check the instrument identification with the `*IDN?` query, type the following command:

```
*IDN? <Carriage Return>
```

The waveform generator should respond to the `*IDN?` query as follows:

```
SONY/TEK,VX4792,<0>,<Firmware Level>
```

***NOTE**. If a number other than 0 is returned following steps 4 or 5, the waveform generator must be repaired.*

4. To invoke the internal diagnostics with the *TST? query, type the following command:

```
*TST? <Carriage Return>
```

After approximately 90 seconds, you can read the test result. The waveform generator should return a 0, which means the test terminated without error. You can also query the test result with the DIAG:RESULT? query.

5. To invoke the self-cal with the *CAL? query, type the following command:

```
*CAL? <Carriage Return>
```

After approximately 30 seconds, you can read the calibration result. The waveform generator should return a 0, which means the calibration terminated without error. You can also query the self-cal result with the SELFCAL:RESULT? query.

This completes the functional check. Your waveform generator is ready for use.

Operating Basics

Functional Overview

The *Functional Overview* section provides the information you need to operate the VX4792 Arbitrary Waveform Generator. We suggest that you read this section before attempting to produce waveforms.

Within this section you will find descriptions of the front-panel controls and indicators, the main operating parameters and modes, equations, and internal diagnostics/calibration:

- *Connectors and Indicators* (page 2–2) provides a brief description of each front panel connector and indicator. For detailed electrical specifications, see *Appendix E, Specifications*.
- Beginning with *Generating Waveforms* (page 2–3), you will find descriptions of the main operating parameters and modes. These subsections provide detailed information about waveform files, memory and data length considerations, and specific operating modes.
- *Equations* (page 2–18) provides detailed instructions for producing waveforms using equations. Several sample waveforms and their equations are included.
- *Diagnostics and Calibration* (page 2–29) describes how to invoke the internal diagnostics or calibration routines, and how to read the test results.

Detailed programming information is not usually provided for topics within this section. *Programming Examples* (page 2–39) contains a group of program examples that show how to perform specific tasks. *Syntax and Commands*, beginning on page 3–1, provides a detailed description of each programming command.

Connectors and Indicators

Callouts in Figure 2–1 identify the connectors and indicators on the waveform generator front panel. A description of each item follows.

CLOCK INPUT Connector

The CLOCK INPUT connector allows the use of an external clock signal. This input accepts clock signals up to 250 MHz, with a maximum voltage of ± 2 V. The nominal impedance is 50 Ω.

TRIGGER INPUT Connector

The TRIGGER INPUT connector allows the use of external trigger or gate signals. Input impedance values of 1 MΩ and 50 Ω are available. The maximum input level is 10 V_{p-p} when 1 MΩ is selected and 5 V_{RMS} when 50 Ω is selected.

CLOCK OUTPUT Connector

The CLOCK OUTPUT connector allows access to the internal clock signal. The clock frequency is up to 250 MHz; the output level is approximately 1 V with a 50 Ω termination.

MARKER 1 OUTPUT Connector

The MARKER 1 OUTPUT connector provides a user-specified marker. The output level is approximately 1 V with a 50 Ω termination.

MARKER 2 OUTPUT Connector

The MARKER 2 OUTPUT connector provides a user-specified marker. The output level is approximately 1 V with a 50 Ω termination.

POWER Indicator

The POWER indicator is lit (green) when power to the mainframe is turned on.

FAILED Indicator

The FAILED indicator is lit (red) when an error is detected during either the internal diagnostics or calibration routines.

AM INPUT Connector

The AM INPUT connector enables input of an external AM modulation signal. An input level of 2 V_{p-p} produces 100% modulation.

SYNC OUTPUT Connector

The SYNC OUTPUT connector provides a sync signal. The output level is approximately 1 V with a 50 Ω termination.

WAVEFORM OUTPUT Connector

The WAVEFORM OUTPUT connector provides the output waveform. The maximum output level is 5 V_{p-p} with a 50 Ω termination. A green LED, located next to the connector, is lit when output is enabled.

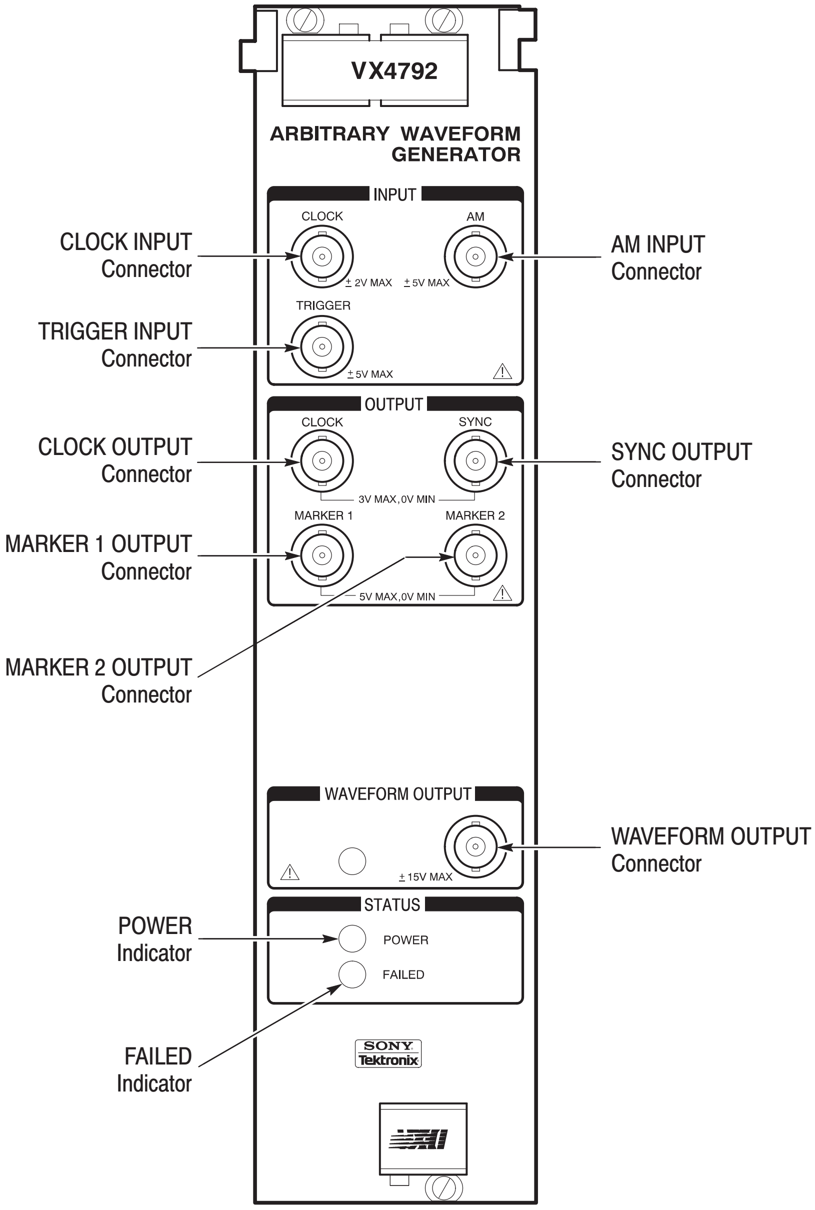


Figure 2–1: Front Panel Connectors and Indicators

Generating Waveforms

Waveform data must be loaded into the internal memory of the VX4792 Arbitrary Waveform Generator before the waveform generator can produce output waveforms. Waveform data is obtained from the four types of waveform files shown in Table 2–1.

Table 2–1: Types of Waveform Files

File Type	Suffix	Description
Waveform data file	Filename.wfm	This data file contains the waveform data. The data can also be transferred from certain Tektronix instruments over a computer interface. You can also create waveform files from equations (see *Equations* on page 2–18).
Waveform equation file	Filename.equ	The waveform equation file contains equations that express waveform characteristics.
Waveform sequence file	Filename.seq	The waveform sequence file specifies a series of waveform data files.
Waveform autostep file	Filename.ast	The waveform autostep file specifies a series of waveform data files and/or waveform sequence files. When the waveform generator executes a waveform autostep file, it generates the waveform for the first file specified. Then it waits for a trigger before generating from the next specified file. Each autostep file may include unique output parameters.

Please remember that the waveform generator does not create waveform files internally. You can load the waveform data in a number of ways, including the following:

- Use a GPIB utility program, such as IBIC, to transfer the data from the standard accessory Sample Waveform Library disk to the VX4792 Arbitrary Waveform Generator.
- Create and compile waveform files using the `EQUAtion:DEFine`, `EQUAtion:COMPile`, and `EQUAtion:WPOints` commands (pages 3–35 through 3–37). See *Equations* on page 2–18 for details.
- Capture waveforms on a digitizer or digitizing oscilloscope and transfer the data to a controller. Then transfer the data to the waveform generator.
- Generate data with C or BASIC programs and transfer the data to the waveform generator.

The waveform data that is available on your controller must be loaded into the internal memory of the waveform generator. Use the `DATA:DESTination` (page 3–29), `WFMPRE` (beginning on page 3–83), and `CURVe` (page 3–27) commands. After waveform data has been loaded into the waveform generator memory, you can select output parameters and other operating modes to produce the desired waveform.

See the *Transferring Waveforms* programming example on page 2–45 for additional information.

Operating Modes

This section describes each of the operating modes used to produce waveforms at the WAVEFORM OUTPUT connector. Use the MODE command (page 3–57) to select between the operating modes.

Continuous Mode

Use continuous mode to continuously output the specified waveform or sequence. In continuous mode, the waveform generator does not require a trigger signal.

Triggered Mode

Use triggered mode to output the specified waveform or sequence once each time the waveform generator receives a trigger signal.

In triggered mode, the waveform generator outputs the specified waveform or sequence once for each trigger it receives. The trigger signal can be an externally applied signal or a START command. If you send another trigger signal during waveform output, the waveform generator ignores the trigger. Figure 2–2 shows the relationship between an example output waveform and an external trigger signal (positive slope) using triggered mode.

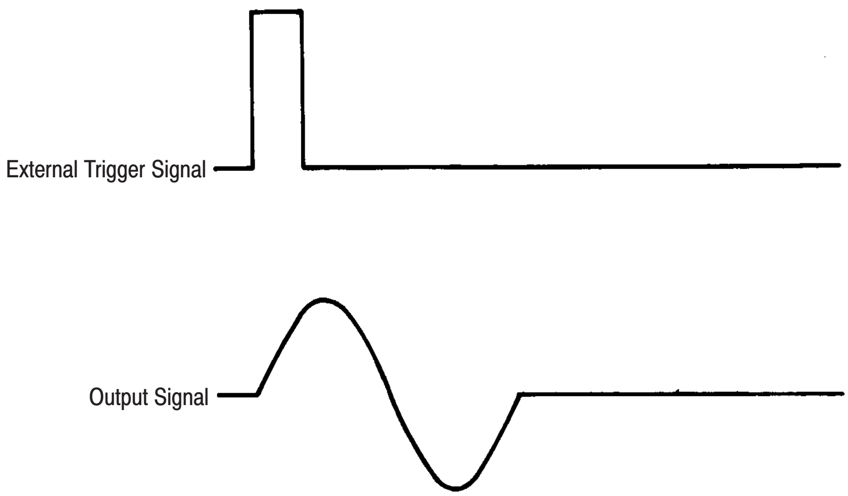


Figure 2–2: External Trigger Signal and Waveform Output in Triggered Mode

Gated Mode

Use gated mode to control waveform or sequence output with a gate signal. You can apply the gate signal to the TRIGGER INPUT connector or simulate a gate signal with START and STOP commands.

When you send a START command, the waveform generator begins to output the waveform. A STOP command stops waveform output. A second START command resumes waveform output from the level where it left off. Figure 2–3 shows the relationship between an example output waveform and an external gate signal (positive polarity) in gated mode.

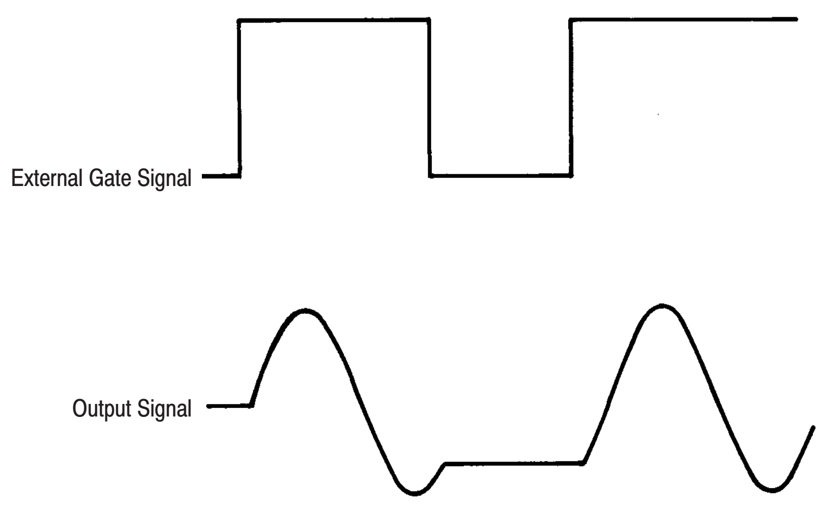


Figure 2–3: External Gate Signal and Waveform Output in Gated Mode

Burst Mode

Use burst mode to produce a specified number of waveform cycles or sequence repetitions. In burst mode, the waveform generator waits for a trigger. On receiving a trigger, the waveform generator generates the specified waveform or sequence until the output signal reaches the burst count (set the count with the `MODE BURST` command, see page 3–57). The trigger can be an external signal or a `START` command. The waveform generator ignores any trigger signals during waveform output.

Figure 2–4 shows the relationship between an example output waveform and trigger signal (positive slope) in burst mode.

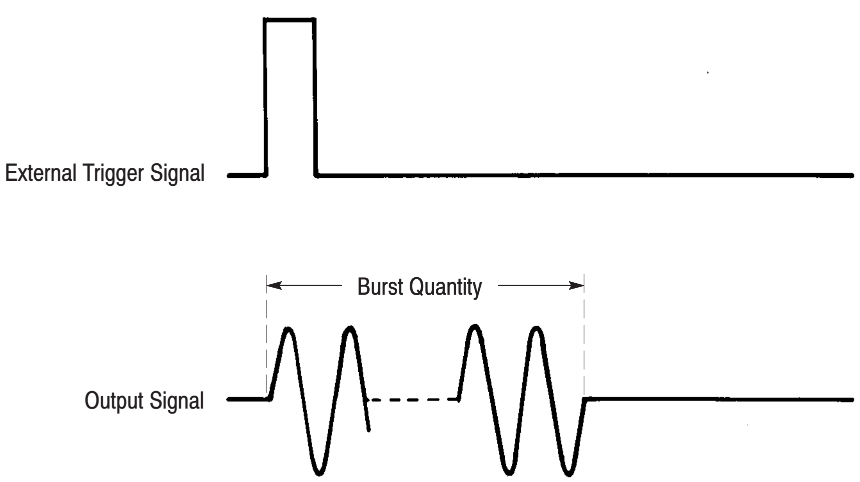


Figure 2–4: Waveform Output in Burst Mode

Waveform Advance Mode

Use waveform advance mode to continuously output a series of specified waveforms in a triggered order. In waveform advance mode, the waveform generator ignores any repetition counts defined in a sequence file. Instead, the waveform generator outputs each waveform within the sequence continuously until it receives the next trigger signal. When the waveform generator reaches the last waveform in the sequence, it returns to the first waveform.

As with all other modes, triggers can be external signals or START commands. Figure 2–5 shows the relationship between trigger (positive slope) and output signals of a two-waveform sequence in waveform advance mode. The waveform file names within the example are Wave-1.WFM and Wave-2.WFM.

Waveform advance mode and continuous mode operate the same way when a single waveform (and not a sequence) is specified.

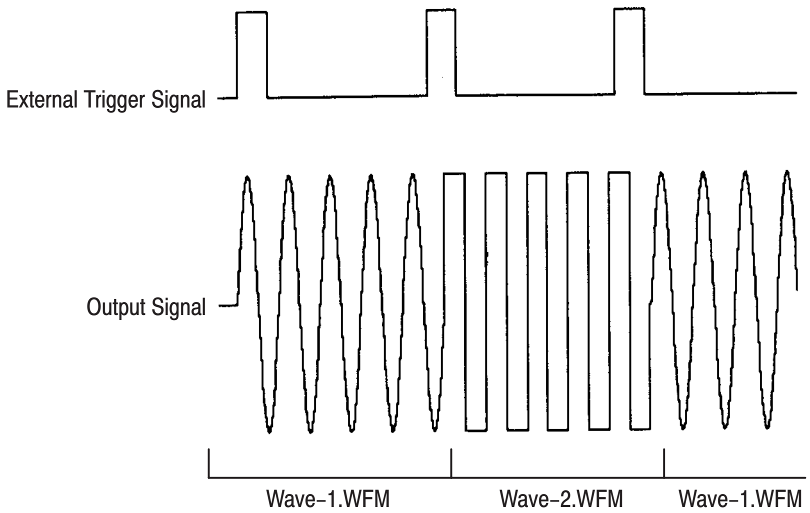


Figure 2–5: Sequence Output in Waveform Mode

Autostep Mode

Use autostep mode to output a specified series of waveforms in a triggered order. Autostep mode operation is similar to waveform advance mode, but autostep mode outputs each waveform one time only. Autostep mode also reads the setup data from the autostep file and then alters the output parameters.

An autostep file stores a program that specifies a waveform or sequence file (including output parameters) for each step. Each time a trigger is received, autostep mode advances to the next waveform or sequence. The waveform for the step is output one time and then stops until the next trigger occurs. Triggers can be external signals or START commands.

Figure 2–6 shows the relationship between trigger (positive slope) and output signals of a two-waveform sequence in autostep mode. The waveform file names within the example are Wave-3.WFM and Wave-4.WFM.

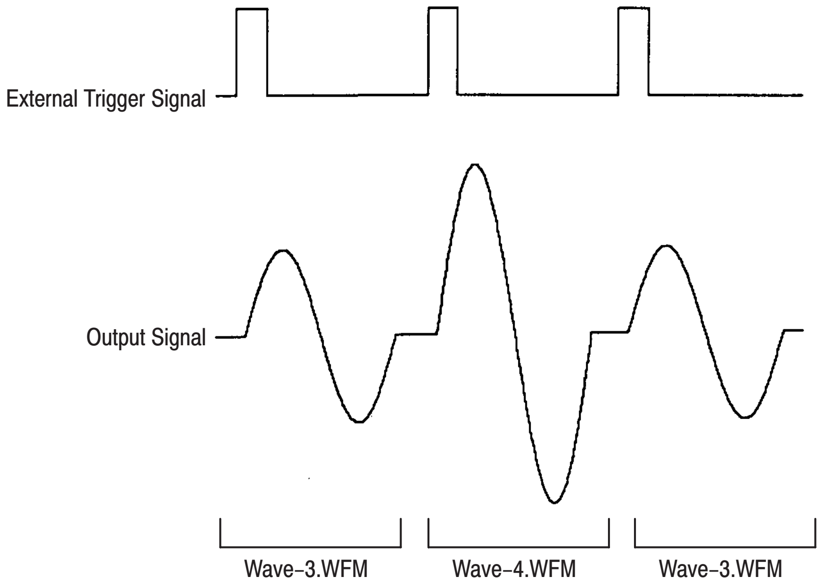


Figure 2–6: Sequence Output in Autostep Mode

Sequence Control

The sequence control system consists of sequence memory that stores the contents of sequence control, and the counters that read out the contents and output the actual waveform memory addresses. Table 2–2 lists an example sequence file.

Table 2–2: An Example Sequence File (XXX.SEQ)

Waveform File Name	Repetition Count
...	...
AAA.WFM	3
BBB.WFM	16
...	...

For the file XXX.SEQ, the VX4792 Arbitrary Waveform Generator will store the following data in memory:

- AAA.WFM actual addresses, AAA.WFM data length, looping counter value: 3
- BBB.WFM actual addresses, BBB.WFM data length, looping counter value: 16

The sequence memory capacity is organized in 8 Kbyte steps for handling complex waveforms. The address and length counters operate with the clock signal from the clock divider frequency divided by eight (because the waveform memory is partitioned into eight banks, this circuit uses a $^1/_8$ clock). Figure 2–7 shows the relationship between sequence memory and waveform memory.

When running the sample sequence file XXX.SEQ (Table 2–2), the sequence control system loads the AAA.WFM addresses into the address counter, the AAA.WFM data length into the length counter, and the looping counter value into the looping counter. When the system reaches the value set in the looping counter (3), it increments the sequence memory address counter and reads the contents of the next step (waveform file BBB.WFM).

When the user sets a burst count, the system places that value in the burst counter and outputs the signal the required number of times.

The waveform generator writes the marker signals into waveform memory in the same manner as waveform data.

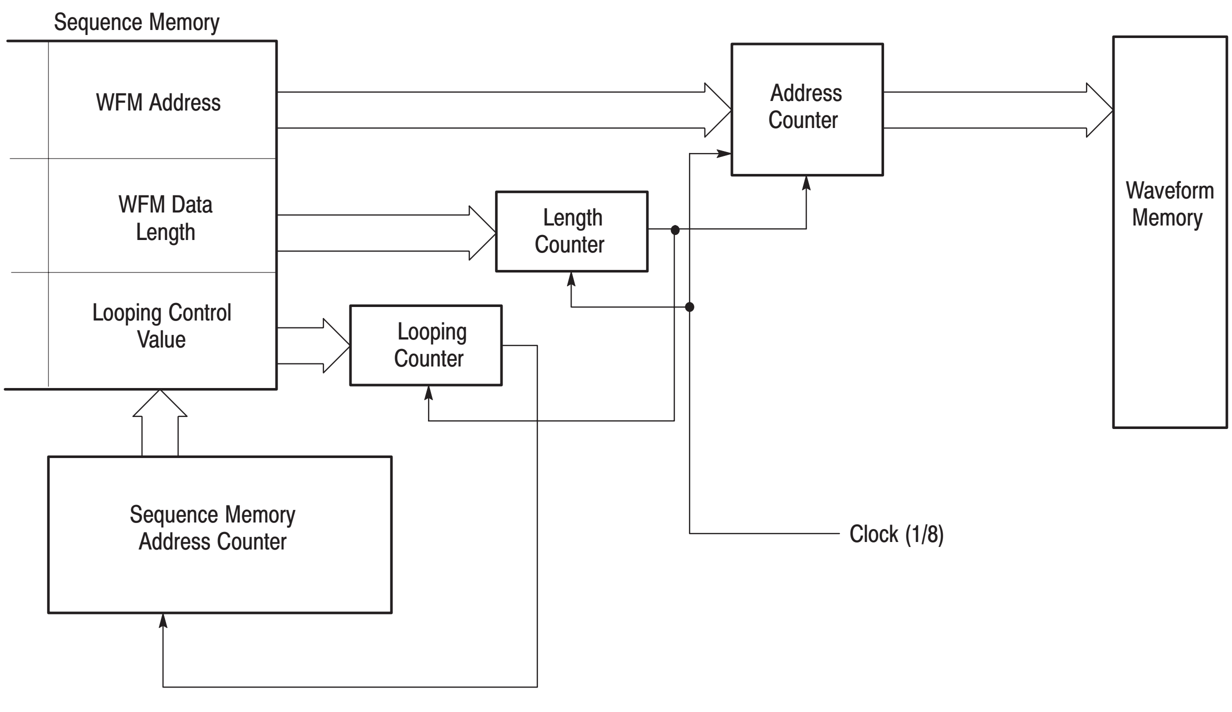


Figure 2–7: Relationship Between Sequence Memory and Waveform Memory

Waveform Memory

The waveform memory consists of 16 32K $\times$ 8 SRAM chips for 256 K words of 16-bit word memory. Of these 16 bits, 12 bits are waveform data, 1 bit is Marker 1, and 1 bit is Marker 2.

Since the system must read waveform memory out at high speed (250 MS/s), the memory is partitioned into eight banks and read out with 8:1 multiplexing (parallel-serial conversion). Therefore, the memory itself operates with a $^1/_8$ clock.

Figure 2–8 shows the waveform memory configuration.

The waveform data length can be any multiple of eight points, from a minimum of 64 points to a maximum of 256 K points. The maximum number of waveform data points in a single sequence file is 256 K.

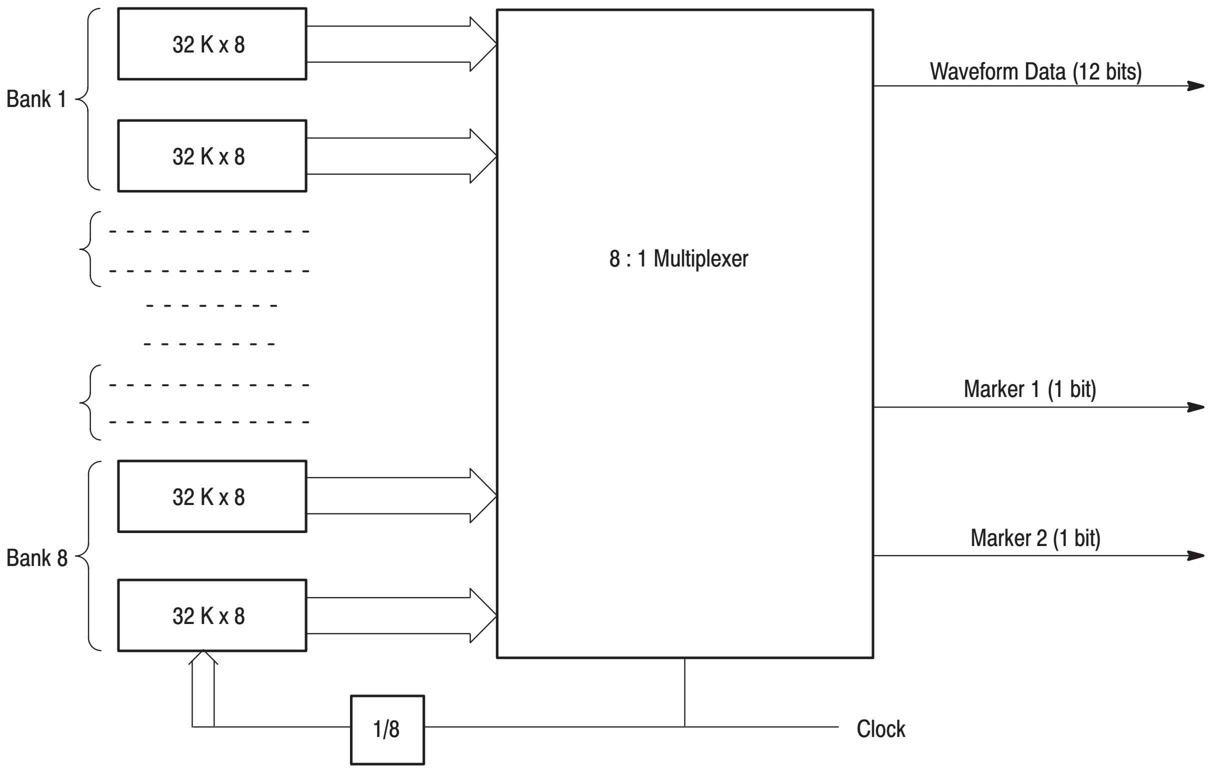


Figure 2–8: Waveform Memory Organization

Data Length

You can specify the number of horizontal data points within a waveform using the `EQUAtion:WPOints` command (see page 3–37) before an equation compiles. Data point values must be multiples of eight, ranging between 64 points and 256 K points.

Generating high-precision (high S/N ratio) waveforms requires an adequate number of data points. For example, triangular waveforms require about 8000 points to minimize the jaggedness of the waveform. The waveform generator uses a Digital-to-Analog Converter with a 12-bit vertical resolution (4096 points) for the vertical amplitude. This resolution allows the instrument to produce complex output waveforms having any waveshape. Figure 2–9 shows the relationship between triangular wave resolution and the number of data points.

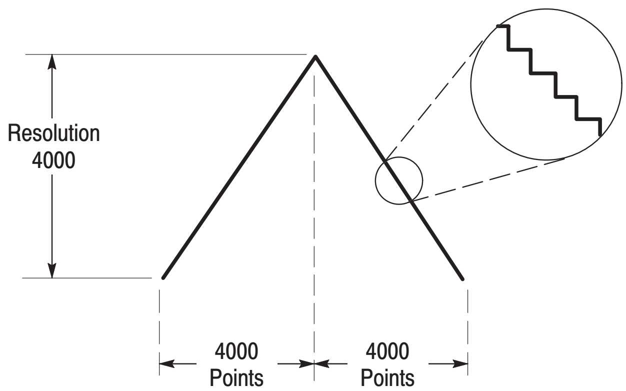


Figure 2–9: Triangular Wave Resolution and Number of Data Points

You will seldom require the level of precision obtained when using maximum resolution. When maximum resolution is not needed, fewer waveform points may be used. In equation and waveform editing, the default value for the number of waveform points is 1000. The size of a fold-back component for a 1000-point sine wave is –60 dB, an acceptable level for most applications.

If you make a waveform with less than 1000 points, be sure to use a filter because you are using the maximum clock frequency. Note the following points:

- According to sampling theory, use of the ideal filter allows you to reproduce the waveform with only two points of data for the highest frequency component in the waveform. The filters in the waveform generator have Bessel functions with relatively gentle shoulder characteristics to eliminate overshoot and ringing. This is why more points are required.
- The amount of data required depends on the waveform shape, the desired S/N ratio, the filter cutoff frequency, and other factors. Check the waveform on an oscilloscope or spectrum analyzer to verify its characteristics.

When Waveform Data Length is Not a Multiple of Eight

Earlier in this section we explained you can only set multiples of eight as the data length, but when data length is small this becomes a problem

If you use triggered mode, you can solve the problem by simply adding data at the end until the total length is a multiple of eight.

In continuous mode, you should multiply the original data length to achieve a multiple of eight. For example, if the original waveform has 60 points, connect two waveforms together to make a waveform with 120 points. Note that when two waveforms are combined, the waveform generator will generate only one sync output for each pair of waveforms (see Figure 2–10).

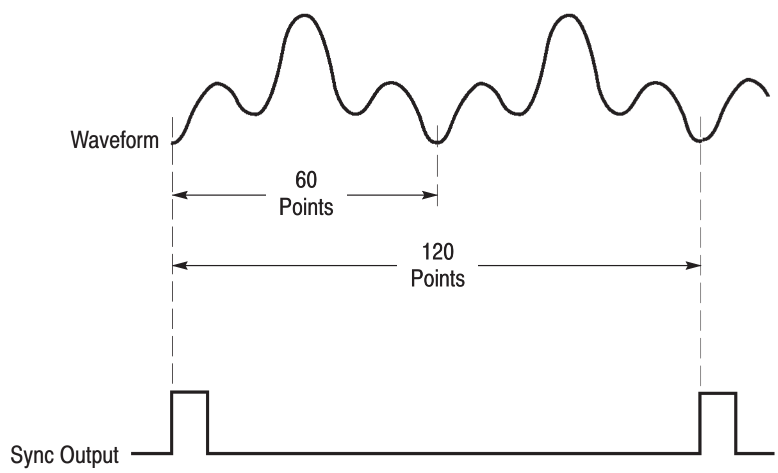


Figure 2–10: Waveform Points and Sync Output in Continuous Mode

Setting Clock Source and Frequency

The clock frequency selected at power-on is 100 MHz using the internal clock source. You can change the clock frequency and source using the `CLOCk:FREQuency` (page 3–25) and `CLOCk:SOURce` (page 3–26) commands, respectively. The internal clock frequency range is 10 Hz to 250 MHz. External clock signals up to 250 MHz may be applied to the CLOCK INPUT connector.

The clock period is the time between the data points for the output waveform. Therefore, the product of that clock period and the number of waveform points is the period for that waveform or sequence waveform. For example, suppose the clock frequency is 1 MHz (for a period of 1 μs). If the waveform has 100 points, the period for the entire waveform is 100 μs (see Figure 2–11).

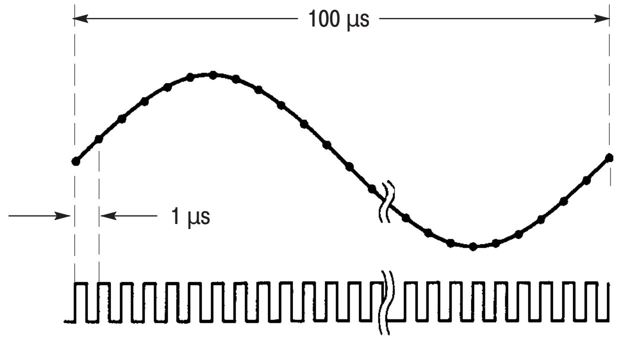


Figure 2–11: Clock and Waveform Points

The output waveform period (frequency) can be changed by selecting a new clock frequency. For example, you may change the clock frequency to 10 MHz (period of 0.1 µs) for the example in Figure 2–11. Now the total period for the 100-point waveform is 10 µs (0.1 µs multiplied by 100).

Setting Amplitude and Offset

The output amplitude and offset settings selected at power-on are 1 V and 0 V, respectively. You can use the `AMPLitude` (page 3–22) and `OFFSet` (page 3–59) commands to select new output amplitude and offset values for the vertical axis (12-bit full scale voltage). The output amplitude can be set in steps of 1 mV in the range from 50 mV_{p-p} to 5.000 V_{p-p}. The output waveform offset can be set in steps of 5 mV within the $\pm$2.5 V range.

Figure 2–12 shows an output waveform when the amplitude is set to 5 V and the offset is set to +1 V.

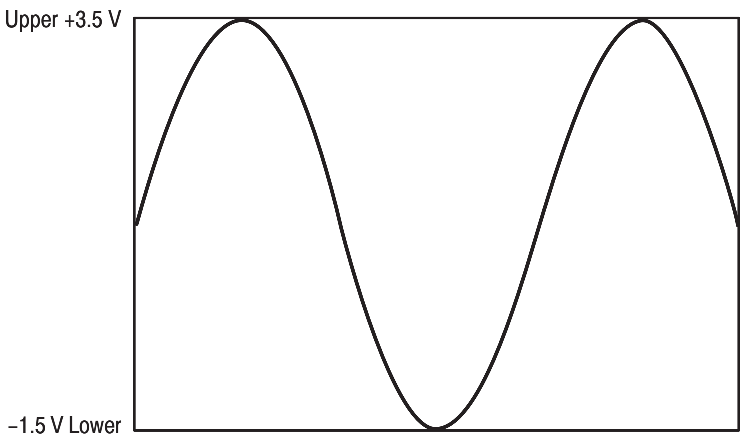


Figure 2–12: Amplitude and Offset Setting

Setting Output Filter

The waveform generator provides four filters for restricting the output frequency bandwidth. The filter selections are 50 MHz, 20 MHz, 5 MHz, 1 MHz, and Through (no filter). You can select a filter using the FILTer command (page 3–41).

Filters within the waveform generator are low pass Bessel-type filters with soft shoulder characteristics to avoid overshooting and ringing. The filter value corresponds to the –3 dB cutoff point. Filter bandwidth is within ± 20% of the value. Figure 2–13 shows a representative curve for the 1 MHz filter.

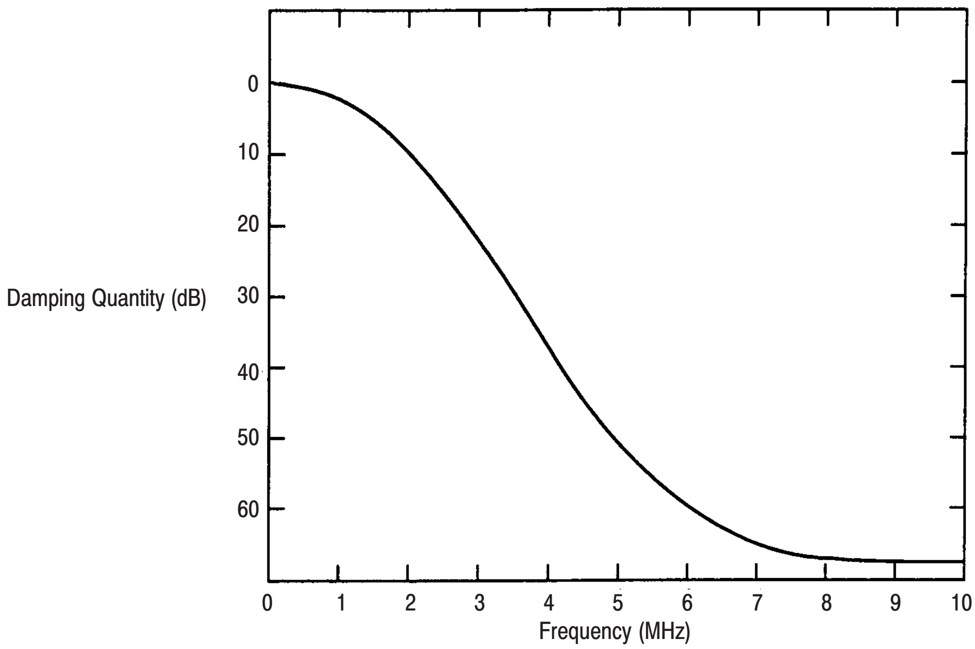


Figure 2–13: Representative Filter Characteristics

The filters can be used to eliminate unwanted noise components of the waveform, and to reduce the fold-back component when there are a small number of waveform points. The filters reduce the jaggedness of the waveform and raise the S/N ratio. The rising and falling time for the waveform is approximately 4 ns when a filter is not used.

Time Delay Due to Filters

Each filter has a unique delay time. This delay affects the timing relationship between the sync signal, marker signals, and waveform output signals. You can reduce the delay by selecting a wider filter value. Table 2–3 shows the delay relative to the sync and marker signals caused by the filters.

Table 2–3: Time Delay Caused By Filters

Filter	Typical Time Delay
1 MHz	390 ns
5 MHz	78 ns
20 MHz	18 ns
50 MHz	11 ns

Setting Trigger Parameters for an External Trigger

The TRIGGER INPUT connector can receive external trigger (gate) signals with a pulse width as narrow as 15 ns. You can use commands to set triggering parameters including slope, polarity, level, and impedance. Each parameter is described below.

Figure 2–14 shows how the slope (polarity) and level controls relate to the external trigger (gate) signal.

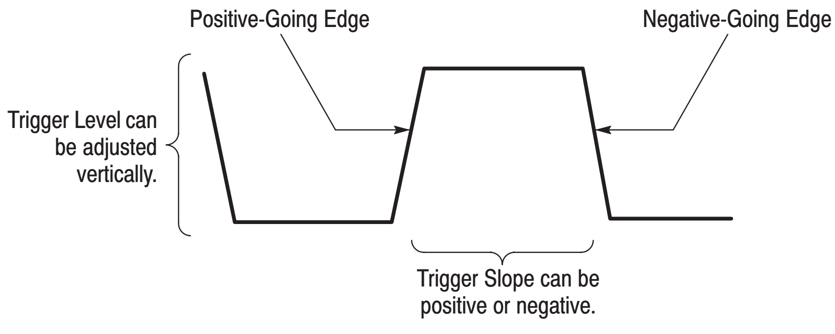


Figure 2–14: Slope and Level Controls

Slope

The `TRIGger:SLOPe` command (page 3–78) sets the slope for the external trigger signal. Slope may be set to either positive or negative. When positive is selected, the output is triggered at the rising edge of the external trigger signal. When negative is selected, the output is triggered at the falling edge of the external trigger signal.

Polarity (Gated Mode)

The `TRIGger:POLarity` command (page 3–77) sets the polarity for the gate that outputs the waveform or sequence with an external gate signal. Polarity may be set to either positive or negative. When positive is selected, the waveform or sequence is output while the level of the gate signal is higher than the gate (trigger) level. When negative is selected, the waveform or sequence is output while the level of the gate signal is lower than the set gate level.

Level

The `TRIGger:LEVel` command (page 3–76) sets the trigger (gate) level for an external trigger (gate) signal. The trigger (gate) level can be set in the range –5.0 V to +5.0 V in steps of 0.1 V.

Impedance

The `TRIGger:IMPedance` command (page 3–74) sets the external trigger (gate) source input impedance to either 50 Ω or 1 MΩ.

Waveform Timing

Figure 2–15 shows the timing relationship between the TRIGGER INPUT signal, the SYNC OUTPUT signal, and the output waveform. Maximum delay times are shown. Typical Sync→Signal Delay is 10 ns; typical Trigger→Signal Delay is 50 ns.

The delay time increases when filters are used. See Table 2–3 on page 2–15 for details.

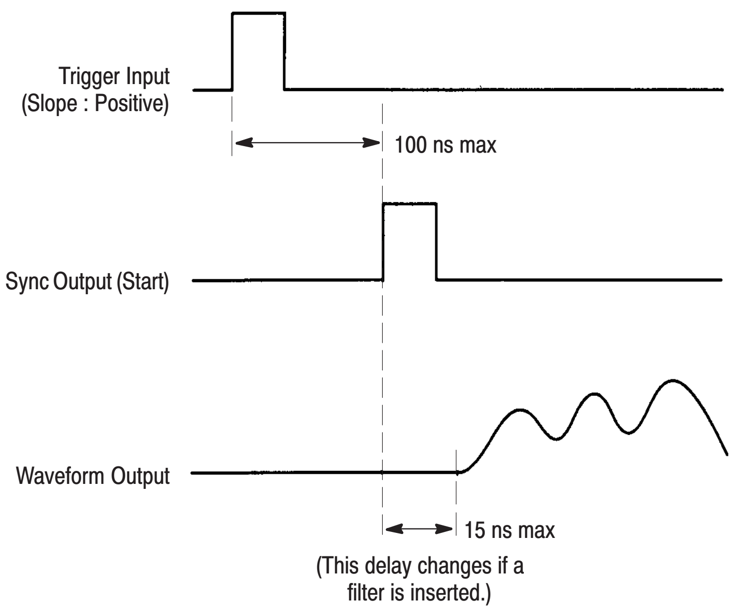


Figure 2–15: Trigger Input, Sync Output, and Waveform Timing

External AM Operation

The EXT AM mode allows you to multiply (modulate) two waveforms and produce the resultant waveform at the WAVEFORM OUTPUT connector. You can use the `OPERation` command (page 3–61) to modulate a waveform in the internal memory with a signal applied to the AM INPUT connector. The modulating signal bandwidth (–3 dB) is DC to 4 MHz.

Figure 2–16 shows the output waveform that would result when multiplying the two waveforms shown using EXT AM mode.

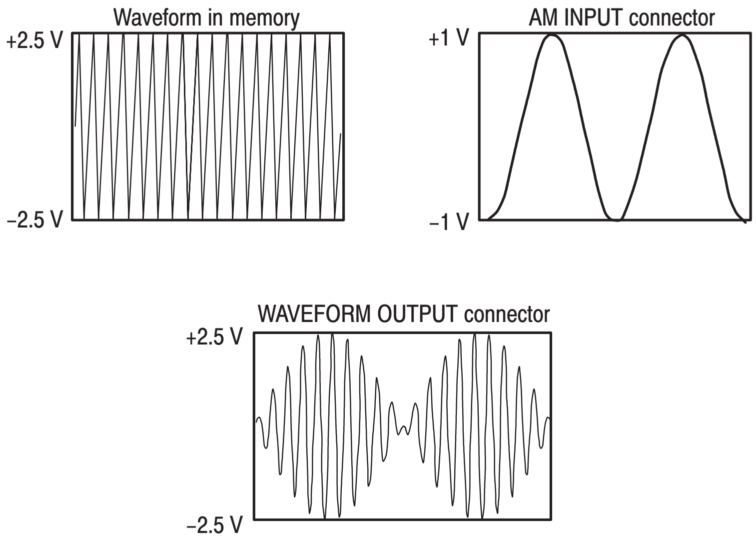


Figure 2–16: External AM Operation (Waveform Multiplication)

Table 2–4 shows the amplitude of the output signal relative to the external modulation signal. 100% amplitude modulation occurs with ± 1 V input. When the external modulation signal is greater than ± 1 V, the output signal can become distorted.

Table 2–4: Output Signal Amplitude Relative to External Modulation Signal

External Modulation Signal	Output Signal Amplitude
0 V	50% of set value
+1 V	100% of set value
–1 V	0% of set value

Setting Sync Signal

The SYNC OUTPUT connector generates a 1 V timing signal with a pulse width of approximately 100 ns when driving a 50 Ω load. The pulse may correspond to either the start or end of the waveform or sequence. You can specify the sync timing using the `OUTPut:SYNC` command (page 3–64).

When START is selected, the sync signal is generated when the trigger or gate signal is generated. However, in continuous mode the sync signal cannot be set to START.

When END is selected, the sync signal is generated at the end of the waveform or sequence waveform for all modes except burst. In burst mode, the sync signal is generated at the end of the burst count.

Equations

You can create waveform files for the waveform generator using the equation feature. Equation file data takes the form of mathematical equations. An equation file (.EQU suffix) is compiled to create a waveform file (.WFM suffix), which can be loaded into memory to generate an output waveform. This section describes the steps needed to create a waveform file, the structure and components used to make equation files, and several example equations.

Creating Equation Files

A programming example located on page 2–43 shows how to generate waveforms using equations. Waveform files are created using a three-step process:

1. Use the `EQUAtion:DEFine` command (page 3–36) to generate the equation (.EQU) file.
2. Use the `EQUAtion:WPOints` commands (page 3–37) to set the number of points within the waveform.
3. Use the `EQUAtion:COMPile` command (page 3–35) to compile the equation file and produce a waveform (.WFM) file.

The compiled waveform file contains the settings for the number of waveform points and the clock frequency (obtained from the total time set with the range statement). Other output parameters are set to the default values.

Each compiled waveform has a vertical axis on which –1.0 is data value 0 and +1.0 is data value 4094. These values have no relationship with the actual output parameters, which are set with the `AMPLitude` (page 3–22) and `OFFSet` (page 3–59) commands.

Equation Structure

When assembling an equation, you first specify the time domain, and then assemble the equation. Figure 2–17 shows an example of equation file data and the waveform obtained by compiling the associated equation file. The equation for Figure 2–17 can be separated into the following statements:

- The range statement “range(0,5 ms)” specifies the time domain
- The statement “sin(4*pi*x)” mathematically defines the waveform, a sine wave with four peaks
- A Line Feed <LF> separates statements within the equation

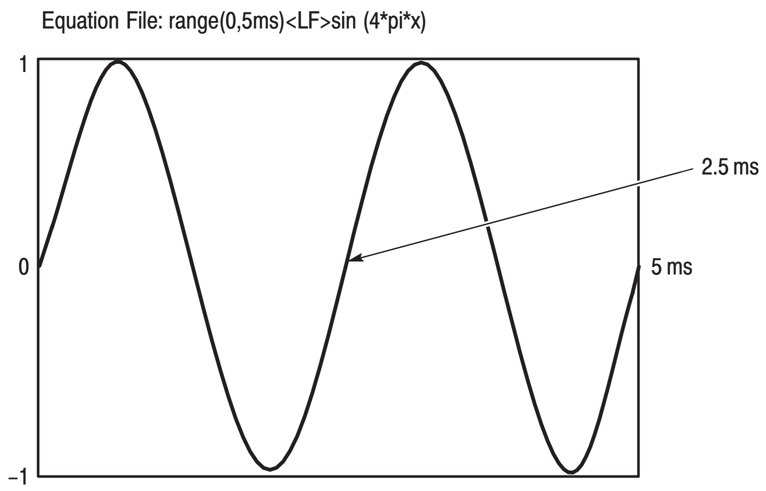


Figure 2–17: Example of Equation File Data and Resulting Waveform

You can use multiple range statements within equations to assemble waveforms of various shapes. Each range statement specifies the time period for the mathematical statements that follow. See Figure 2–28 on page **¿¿¿¿** for an example waveform which requires multiple range statements.

Specifying the Time Domain

Each equation must specify the time domain. An error will result if the time domain is not defined. You specify the time domain with the statement “range()”. Use the following format:

range(Equation starting time,Equation ending time)

When making a new equation file, “range(0,” is the starting point of the waveform. Next you specify the ending time. This setting is valid until another range statement is specified. Following the first range() specification, any number of mathematical statements can be input.

The clock frequency is obtained from the total time (period) set with the range statement and the waveform point count. The resulting clock frequency can be determined as follows:

Clock frequency = Waveform point count ÷ Equation period

32-bit fixed precision is used when precision is not required (minimum unit 15 μs or greater). When precision is required, 32-bit floating point precision (IEEE 754 compatible) is used.

Equation Components

Table 2–5 provides a summary of the syntax items, operators, variables, constants, and functions you may use to assemble equations.

Figure 2–18 through Figure 2–32 provide a variety of example equations and their resulting waveforms. *Appendix D* provides additional examples of equations and their resulting waveforms, for waveform files located on the Sample Waveform Library disk.

Table 2–5: Components for Assembling Equations

Type	Symbol	Description
Syntax items	()	Use parentheses "(" and ")" for specifying the order of operations. Each opening (left) parenthesis must be paired with a closing (right) parenthesis. When there are two arguments — for example, range, max, min — the arguments are separated with a ","(comma).
Operators[1]	*	Multiply
	/	Divide
	+	Add
	–	Subtract
	^	Expresses exponents. ^ has the same priority as * and /. Therefore, parentheses are required to give priority to multiplication. Example: pi * (2^3) * x where 2^3 = two raised to the third power.
Variables	t	Time from the head of that range() statement.
	x	Variable taking on a value from 0.0 to 1.0 within that range(0 statement. X has a value of 0.0 at the beginning of the range() and 1.0 at the end of the range().
	v	Variable showing the current value of the waveform data at that position.
Constants	pi	The circumferential ratio.
	e	Exponent (for an implied 10) (Examples: 1e6=1 10^6=1,000,000, 1e–3=0.001) The range for numbers expressed in this scientific notation is from |5.9e–39| to |3.4e38|.
	k	The k0–k9 can be specified; these are constants that may be used in equations. Specifying a new value for the same k# replaces the old value with the new one.
	=	Equals sign. = is used with k constants. Example: k0=2*pi

1 **The operators * and / have priority over + and –.**

Table 2–5: Components for Assembling Equations (Cont.)

Type	Symbol	Description
Functions	sin(, cos(	The arguments for these trigonometric functions are in radians. See Figure 2–18 and Figure 2–19 for example equations.
	exp(, log(, ln(	Exponential function, common log function, natural log function. See Figure 2–20, Figure 2–21, and Figure 2–22 for example equations.
	sqrt(	The square root; the argument must be a positive value. See Figure 2–23 for example equation.
Functions	abs(	The absolute value. See Figure 2–24 for example equation.
	int(	Truncates the fraction to obtain the integer. See Figure 2–25 for example equation.
	round(	Rounds off the fraction to obtain the integer. See Figure 2–26 for example equation.
	norm(	Normalizes the range specified with range() and scales the amplitude values so that the maximum absolute value is 1.0 (+1.0 or –1.0). The norm() statement comprises an entire line. See Figure 2–27 for example equation.
	max(, min(	Takes the larger or smaller of two values. See Figure 2–28 for example equation.
	rnd	Integer from 1 to 16777215 — When an argument is specified, generates a random number sequence using that argument as the initial value. If the argument is omitted, 1 is used. See Figure 2–29 for example equation. For further information about this function, refer to *Random (rnd) Function*, on page G–4.
	diff(	Differentiates the function over the range specified with range(). Specified with diff(). The diff() comprises an entire line. Differentiating the waveform in Figure 2–30 gives the waveform shown in Figure 2–31. For further information about this function, refer to *Differentiation*, on page 0–1.
	integ(	Integrates the function over the range specified with range(). Specified with integ(). The integ() comprises an entire line. After integ(), specify normalization (norm()) as necessary. Integrating the waveform in Figure 2–30 gives the waveform in Figure 2–32. For further information about this function, refer to *Integration*, on page G–3.
	mark(	Marker 1 or 2; sets the marker for the range set with range(). The mark() statement comprises an entire statement. For example, when mark(1) is input, <LF> must come before and after the statement.

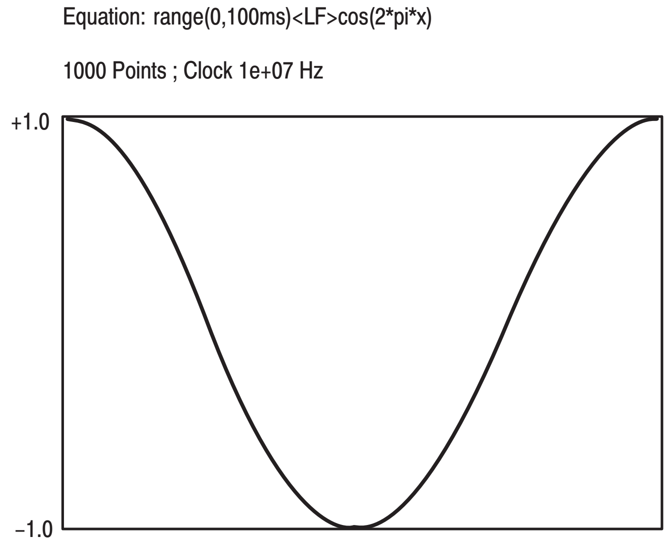


Figure 2–18: Trigonometric Function Waveform Expressed with Variable x

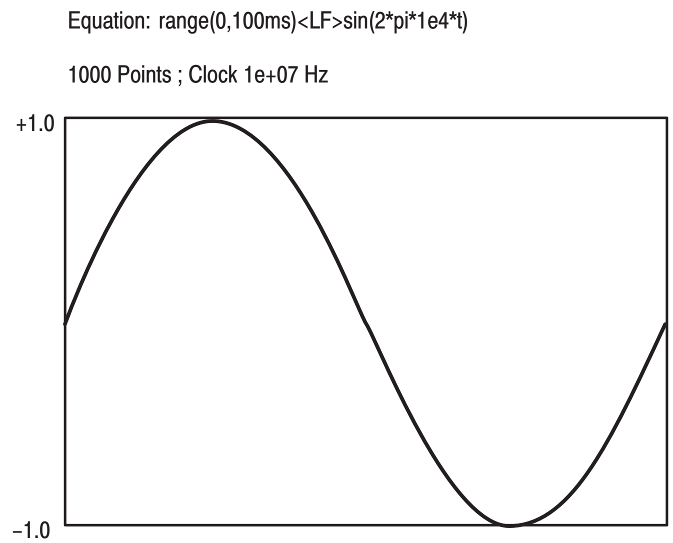


Figure 2–19: Trigonometric Function Waveform Expressed with Variable t

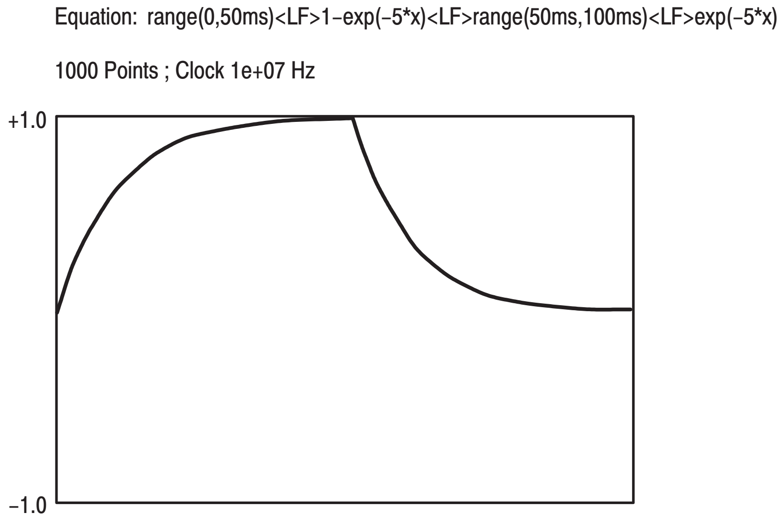


Figure 2–20: Equation Using Exp

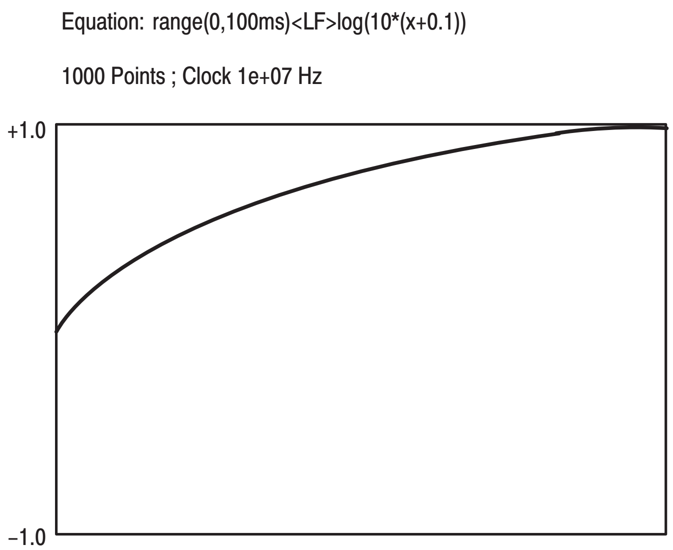


Figure 2–21: Equation Using Log

Equation: range(0,100ms)<LF>ln(2*(x+0.2))

1000 Points ; Clock 1e+07 Hz

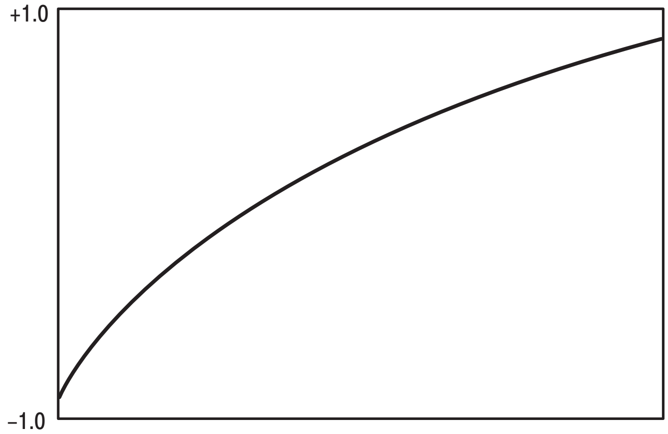


Figure 2–22: Equation Using Ln

Equation: range(0,100ms)<LF>sqrt(sin(pi*x))

1000 Points ; Clock 1e+07 Hz

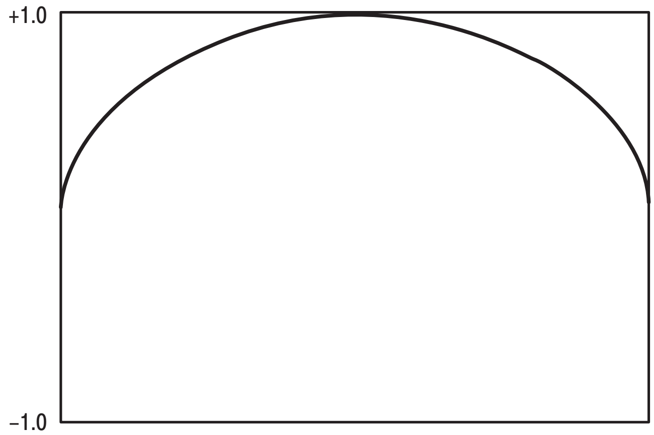


Figure 2–23: Equation Using Sqrt

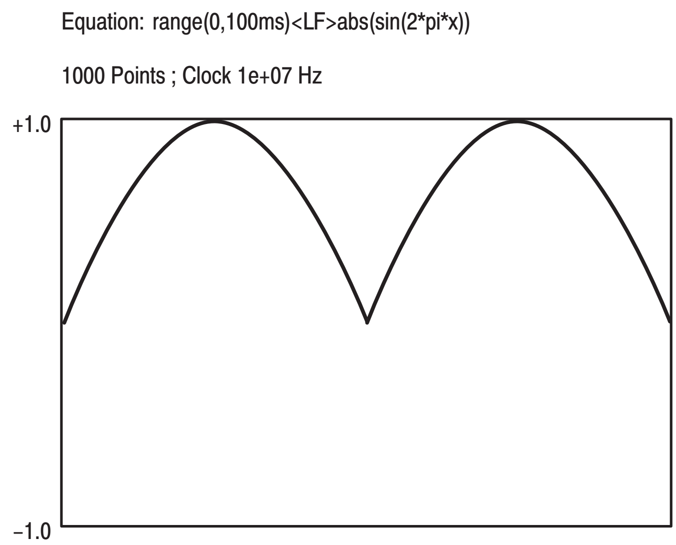


Figure 2–24: Equation Using Abs

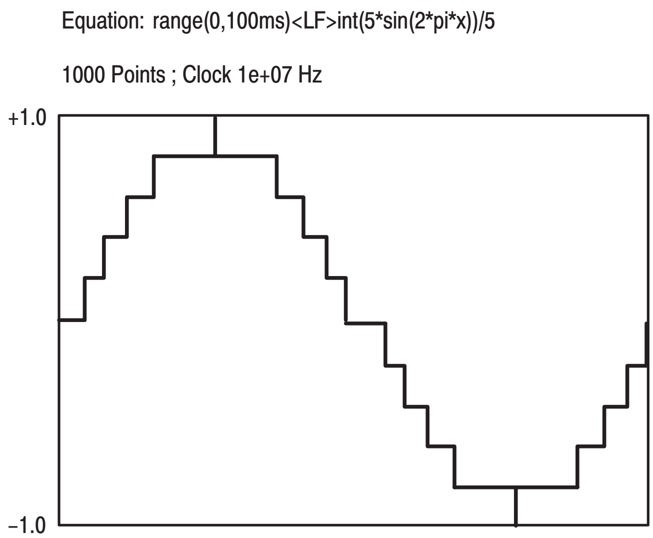


Figure 2–25: Equation Using Int

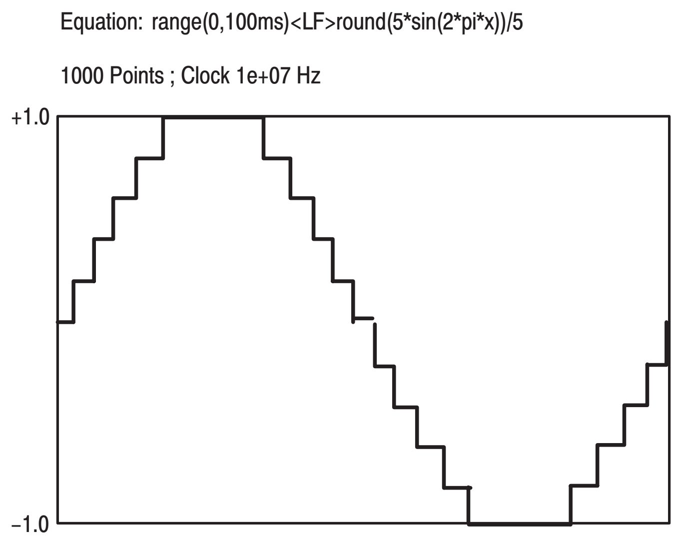


Figure 2–26: Equation Using Round

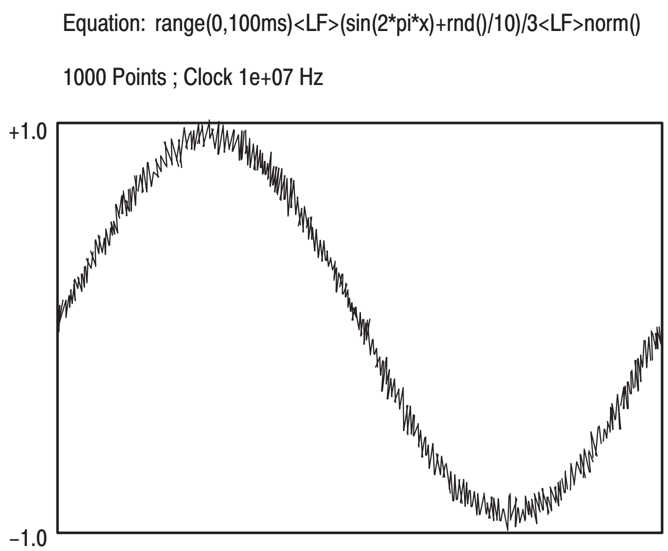


Figure 2–27: Equation Using Norm

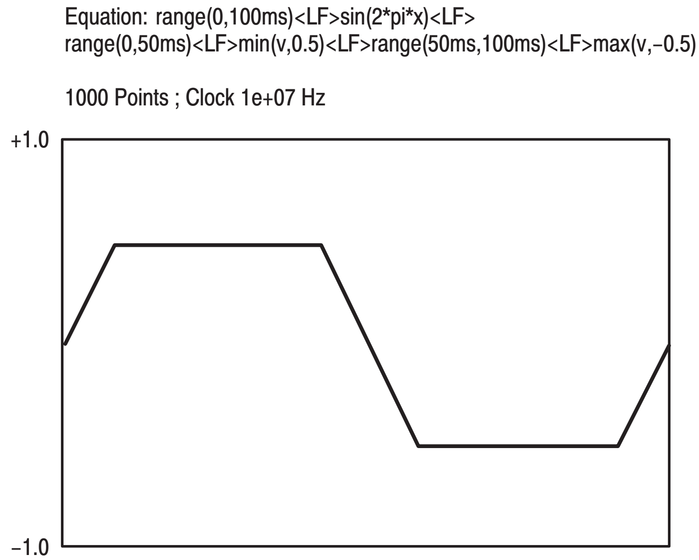


Figure 2–28: Equation Using Min and Max

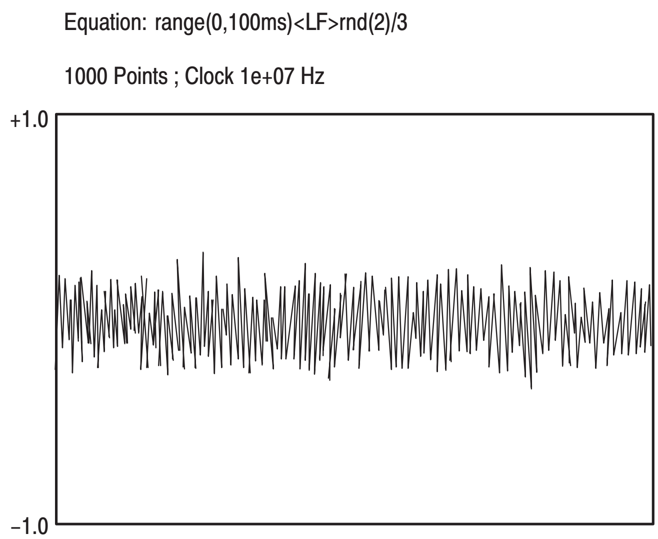


Figure 2–29: Equation Using Rnd

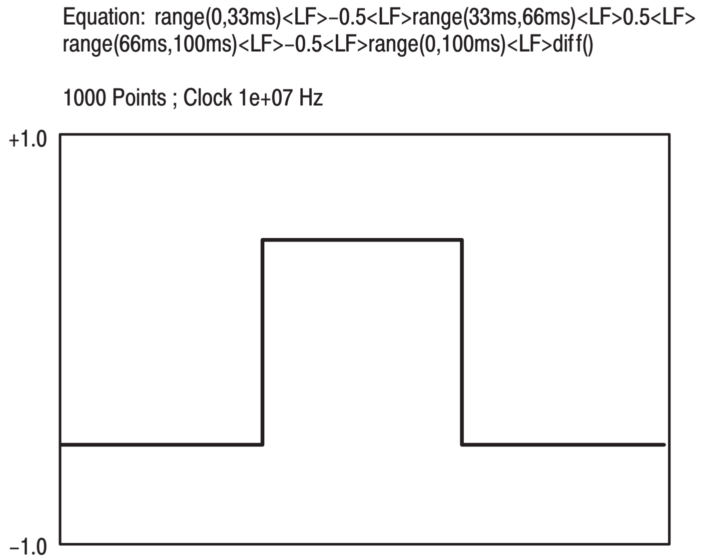


Figure 2–30: Waveform Before Calculating Differentiation

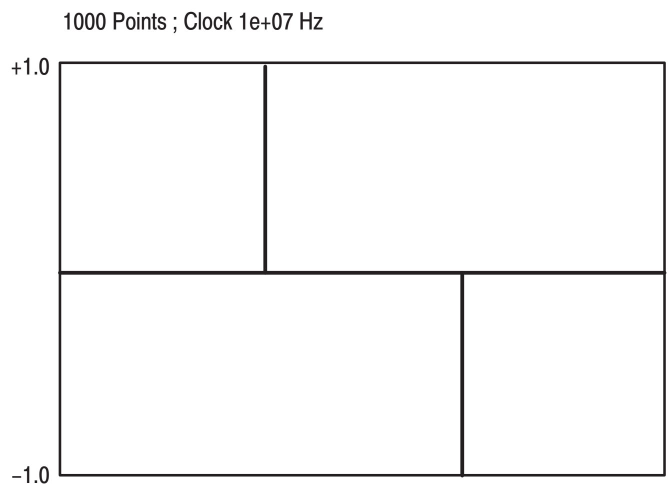


Figure 2–31: Waveform After Differentiation Using Diff

Equation: range(0,33ms)<LF>–0.5<LF>range(33ms,66ms)<LF>0.5<LF>range(66ms,100ms)<LF>
–0.5<LF>range(0,100ms)<LF>integ()<LF>norm()

1000 Points ; Clock 1e+07 Hz

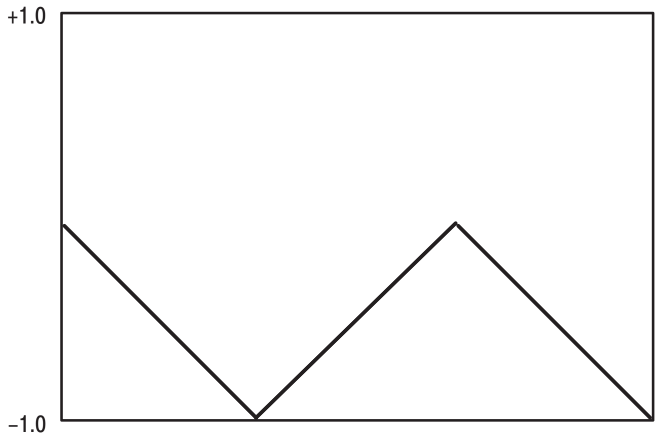


Figure 2–32: Waveform After Integration Using Integ

Diagnostics and Calibration

The diagnostic routines provide information regarding VX4792 Arbitrary Waveform Generator functionality. In the event of failure, these routines provide information you can use to identify the faulty circuits. However, these routines isolate to the functional block level only; they do not provide direct information about failed components.

The calibration routine verifies and adjusts the basic analog calibration of the waveform generator.

Background Information

First-line service strategy for VXIbus products calls for replacement of failed instrument modules. If a VXIbus subsystem fails, the diagnostic and calibration routines identify the faulty module, which should be replaced. The functions described in this section support such failure detection and correction.

SYSFAIL~ Signal

For purposes of VXIbus products, the SYSFAIL~ signal asserted by an instrument module indicates to the Resource Manager that the instrument kernel has not passed a self-test.

A lighted front-panel FAILED indicator signifies to the user that the SYSFAIL~ signal is asserted.

Front-Panel Indicators

In many instances, the front-panel indicators provide sufficient information to identify an instrument module failure. A red-lighted indicator signifies that the CPU is inoperative.

Diagnostic Routines

Diagnostic routines for the waveform generator consist of the CPU test, setup test, and waveform test. These routines run automatically at power-on. You can invoke these diagnostics at any time with the *TST? command.

The CPU test routine verifies that the processor and ancillary circuits are capable of supporting the operating system.

The setup test and waveform test routines verify that the waveform generator hardware is capable of proper operation.

***NOTE**. The instrument immediately reports module failures by lighting the front-panel FAILED indicator and by asserting the VXIbus SYSFAIL~ signal.*

Self-Calibration

The self-calibration routines run automatically at power-on. You can obtain maximum accuracy from the waveform generator by performing a self-calibration (using the *CAL? query) just before use.

Diagnostic and Calibration Commands

The commands and queries listed in Table 2–6 are used for diagnostics and calibration. Refer to *Syntax and Commands* for detailed descriptions of each command.

Table 2–6: Diagnostic and Calibration Commands

Header	Description
*CAL?	Perform calibration
DIAG?	Query all current settings related to self-test
DIAG:RESUlt?	Query self-test result
DIAG:SELect(?)	Select self-test routine
DIAG:STATe	Perform self-test
SELFcal?	Query all current settings related to calibration
SELFcal:RESUlt?	Query calibration result
SELFcal:SELect(?)	Select calibration routine
SELFcal:STATe	Perform calibration
*TST?	Perform self-test

Operational Considerations

If the waveform generator hardware fails during normal operation, the diagnostics (invoked by the *TST? command) may detect the failure. The module controller can then read the error messages generated by the waveform generator. You can also verify or optimize calibration during normal operation by invoking the *CAL? command with appropriate arguments.

The waveform generator will light its FAILED indicator if the diagnostics or calibration routines detect an error. This indicates that the module requires servicing.

Instrument I/O

The VX4792 Arbitrary Waveform Generator rear-panel interface consists of VXIbus connectors P1 and P2. The pins on P1 and the inner row (Row B) on P2 are configured as defined in the *VMEbus Specification*, *IEEE Standard 1014*. The pins on the outer rows of P2 (Rows A and C) are configured as defined in the *VXIbus System Specification*, *Revision 1.3, dated July 14, 1988*.

Tables 2–7 through 2–10 list the pin assignments for each connector. Connectors are identified (left or right) when viewing the waveform generator from the front panel. Refer to Figure 2–33 for connector locations. For detailed information regarding the signals on the pins, refer to the VMEbus and VXIbus standards referenced above.

NOTE. *In the following tables, "NC" means "No Connection."*

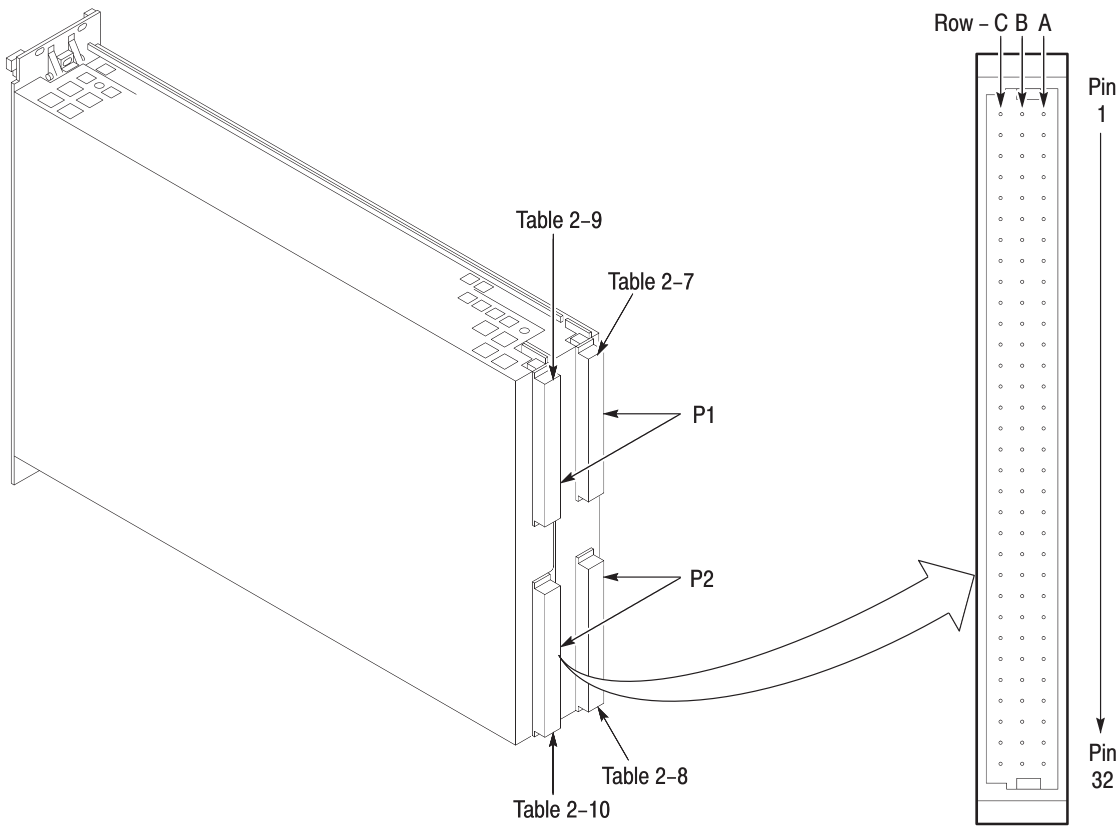


Figure 2–33: VXIbus Connectors P1 and P2

Table 2–7: Left Slot P1 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	NC	NC	NC
2	NC	NC	NC
3	NC	NC	NC
4	NC	BG0IN˜	NC
5	NC	BG0OUT˜	NC
6	NC	BG1IN˜	NC
7	NC	BG1OUT˜	NC
8	NC	BG2IN˜	NC
9	GND	BG2OUT˜	GND
10	NC	BG3IN˜	NC
11	GND	BG3OUT˜	NC
12	NC	NC	NC
13	NC	NC	NC
14	NC	NC	NC
15	GND	NC	NC
16	NC	NC	NC
17	GND	NC	NC
18	NC	NC	NC
19	GND	NC	NC
20	NC	GND	NC
21	IACKIN˜	NC	NC
22	IACKOUT˜	NC	NC
23	NC	GND	NC
24	NC	NC	NC
25	NC	NC	NC
26	NC	NC	NC
27	NC	NC	NC
28	NC	NC	NC
29	NC	NC	NC
30	NC	NC	NC
31	–12 V	NC	+12 V
32	+5 V	+5 V	+5 V

Table 2–8: Left Slot P2 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	ECLTRG0	+5 V	NC
2	–2 V	GND	NC
3	ECLTRG1	NC	GND
4	GND	NC	–5.2 V
5	LBUSA00	NC	LBUSC00
6	LBUSA01	NC	LBUSC01
7	–5.2 V	NC	GND
8	LBUSA02	NC	LBUSC02
9	LBUSA03	NC	LBUSC03
10	GND	NC	GND
11	LBUSA04	NC	LBUSC04
12	LBUSA05	GND	LBUSC05
13	–5.2 V	+5 V	–2 V
14	LBUSA06	NC	LBUSC06
15	LBUSA07	NC	LBUSC07
16	GND	NC	GND
17	LBUSA08	NC	LBUSC08
18	LBUSA09	NC	LBUSC09
19	–5.2 V	NC	–5.2 V
20	LBUSA10	NC	LBUSC10
21	LBUSA11	NC	LBUSC11
22	GND	GND	GND
23	NC	NC	NC
24	NC	NC	NC
25	+5 V	NC	GND
26	NC	NC	NC
27	NC	NC	NC
28	GND	NC	GND
29	NC	NC	NC
30	NC	NC	GND
31	GND	GND	+24 V
32	NC	+5 V	–24 V

Table 2-9: Right Slot P1 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	D00	BBSY	D08
2	D01	BCLR	D09
3	D02	ACFAIL	D10
4	D03	BG0IN	D11
5	D04	BG0OUT	D12
6	D05	BG1IN	D13
7	D06	BG1OUT	D14
8	D07	BG2IN	D15
9	GND	BG2OUT	GND
10	SYSCLK	BG3IN	SYSFAIL˜
11	GND	BG3OUT	BERR˜
12	DS1	BR0	SYSRESET˜
13	DS0	BR1	NC
14	WRITE	BR2	AM5
15	GND	BR3	A23
16	DTACK˜	AM0	A22
17	GND	AM1	A21
18	AS	AM2	A20
19	GND	AM3	A19
20	IACK	GND	A18
21	IACKIN	SERCLK(1)	A17
22	IACKOUT˜	SERDAT˜(1)	A16
23	AM4	GND	A15
24	A07	IRQ7	A14
25	A06	IRQ6	A13
26	A05	IRQ5˜	A12
27	A04	IRQ4˜	A11
28	A03	IRQ3˜	A10
29	A02	IRQ2˜	A09
30	A01	IRQ1˜	A08
31	–12 V	+5 VSTDBY	+12 V
32	+5 V	+5 V	+5 V

Table 2–10: Right Slot P2 Pinout

Pin Number	Row A Signal Mnemonic	Row B Signal Mnemonic	Row C Signal Mnemonic
1	NC	+5 V	CLK10+
2	–2 V	GND	CLK10–
3	NC	RSV1	GND
4	GND	A24	–5.2 V
5	LBUSA00	A25	LBUSC00
6	LBUSA01	A26	LBUSC01
7	–5.2 V	A27	GND
8	LBUSA02	A28	LBUSC02
9	LBUSA03	A29	LBUSC03
10	GND	A30	GND
11	LBUSA04	LBUSC04	A31
12	LBUSA05	GND	LBUSC05
13	–5.2 V	+5 V	–2 V
14	LBUSA06	D16	LBUSC06
15	LBUSA07	D17	LBUSC07
16	GND	D18	GND
17	LBUSA08	D19	LBUSC08
18	LBUSA09	D20	LBUSC09
19	–5.2 V	D21	–5.2 V
20	LBUSA10	D22	LBUSC10
21	LBUSA11	D23	LBUSC11
22	GND	GND	GND
23	TTLTRG0˜	D24	TTLTRG1˜
24	TTLTRG2˜	D25	TTLTRG3˜
25	+5 V	D26	GND
26	TTLTRG4˜	D27	TTLTRG5˜
27	TTLTRG6˜	D28	TTLTRG7˜
28	GND	D29	GND
29	RSV2	D30	RSV3
30	MODID	D31	GND
31	BND	GND	+24 V
32	SUMBUS	+5 V	–24 V

Programming Examples

The *Programming Examples* section provides programming examples that describe the following instrument control functions:

- Instrument control and basic waveform generation (page 2–40)
- Synchronous arbitrary waveform generation (page 2–41)
- Define equations (page 2–43)
- Compile equations (page 2–44)
- Transfer waveforms (page 2–45)

Before using the example programs, be sure to read the *Preliminary Information* that follows.

Preliminary Information

The VX4792 Arbitrary Waveform Generator requires some setup before you can generate waveforms. The module must be configured and installed into a VXIbus mainframe (see page 1–3 for instructions). A controller and software that can communicate manually with the waveform generator are also required. A possible configuration may include, but is not limited to, the following components:

- Tektronix VX1410 VXIbus Mainframe
- Tektronix VX4521 Advanced Slot 0 Resource Manager
- IBM-PC compatible computer with National Instruments GPIB-PC2A card
- IBIC software

Refer to the applicable user manuals of your system components for additional information.

To generate arbitrary waveforms for use within some of the examples, you must first generate waveform data externally and load it into the waveform generator over the VXIbus. Refer to *Generating Waveforms* on page 2–3 for additional details.

Some commands and instrument responses in the following examples may be too long to fit on one printed line. When this happens, the line will break at some point and continue on the line immediately following. The line will never break where a space character should be entered. Be careful to type the entire command as one program message; do not break it up (as it appears in the manual).

The examples in this section use the complete command forms, but the abbreviated forms are also valid.

Instrument Control and Basic Waveform Generation

This procedure describes the general approach for loading a waveform file, changing the waveform parameters, and enabling the output. When performing this example, you can view the output waveform on an oscilloscope to see how each new command changes the output parameters. The procedure assumes that a waveform file already exists in the computer memory.

NOTE. *If no waveform data is located in the internal memory of the VX4792 Arbitrary Waveform Generator, the WAVEFORM OUTPUT connector cannot be activated.*

1. Load the waveform data into waveform memory with the following command:

```
WAVEFORM "sample.wfm"
```

where `"sample.wfm"` is a waveform file that is in the waveform generator memory.

2. Change the output parameters as desired. You can alter clock source, clock frequency, maximum full-scale voltage, offset voltage, low pass filters, and many other parameters. For example, the commands

```
CLOCK:FREQUENCY 250MHZ
AMPLITUDE 5
OFFSET 1
FILTER M1
```

will set the source clock frequency to 250 MHz, the maximum full-scale voltage to 5 V, the offset voltage to +1 V, and activate the low pass filter with a cut-off frequency of 1 MHz. You can verify your settings with the `CLOCK:FREQUENCY?` and `CH1?` queries.

3. Activate the output at the WAVEFORM OUTPUT connector with the following command:

```
OUTPUT:STATE ON
```

The LED beside the WAVEFORM OUTPUT connector turns green to show the output is active.

Synchronous Operation

Generating synchronous output from multiple instrument modules requires careful preparation. During synchronous operation of two or more modules, the left-most module within the mainframe is the master. Slave modules, which are controlled by the master module, are located to the right of the master module.

The operating mode determines whether a trigger is needed to generate synchronous outputs. Table 2–11 summarizes the trigger requirements for the various output modes. When operating synchronously, the waveform advance mode should be used instead of continuous mode when generating one waveform.

Table 2–11: Trigger Requirements for Synchronous Output

Output Mode	Trigger Required
Triggered Mode	Yes
Burst Mode	Yes
Waveform Advance Mode	Yes
Waveform Advance Mode (one waveform)	No
Gated Mode	No

The following procedure illustrates a setup with a single, line-triggered master module and a single slave module.

1. *Setup the master module.*

a. Set the master module to operate in waveform advance mode with the following command:

```
MODE WADVANCE
```

In waveform advance mode, the waveform generator will wait for a `START` command or an external trigger before generating the waveform.

b. Load a waveform file into the waveform memory using the following command:

```
WAVEFORM "sample.wfm"
```

The file `"sample.wfm"` is any waveform file that is in the waveform generator memory. The waveform generator settings will match the output parameters in the waveform file.

c. Select a trigger line using the following commands:

```
TRIGGER:INPUT ECLTRIGO
TRIGGER:OUTPUT ECLTRIGO
```

d. Set the output parameters as desired (see step 2 of the procedure on page 2–40).

2. *Setup the slave module.*

***NOTE**. Always select the operating mode and clock source before loading a waveform.*

---

a. Set the slave module to operate in waveform advance mode (same as the master module) with the following command:

```
MODE WADVANCE
```

b. Set the clock source to the local VXIbus with the following command:

```
CLOCK:SOURCE LBUS
```

c. Load a waveform file into the waveform memory with the following command:

```
WAVEFORM "sample.wfm"
```

The waveform generator settings will match the output parameters in the waveform file.

d. Set the slave module to trigger from the same line as the master module with the following command:

```
TRIGGER:INPUT ECLTRIGO
TRIGGER:OUTPUT OFF
```

e. Set the output parameters as desired (see step 2 of the procedure on page 2–40).

3. Activate the output at the WAVEFORM OUTPUT connector with the following command:

```
OUTPUT:STATE ON
```

4. Now the master module is waiting for a trigger signal. Apply a trigger signal to the TRIGGER connector or enter the following command:

```
START
```

Both modules will output waveforms that are synchronized to the same trigger.

Defining Equations

The following procedure shows how to define, compile, and output a waveform using an equation to generate a waveform file. You might want to view the output waveform on an oscilloscope to verify its shape (sine wave) and period (1 μs).

***NOTE**. Do not use CR (0x0D) as a terminator in your equation file. Use only LF (0x0A).*

---

1. Define the waveform file. Use the following command

   ```
   EQUATION:DEFINE "new.equ",#224range(0,1µs)<LF>sin(2*pi*x)<LF>
   ```

   to define an equation file named "new.equ" that produces a one-cycle sine wave when compiled.

2. Set the number of waveform points. Use the following command

   ```
   EQUATION:WPOINTS "new.equ",128
   ```

   to set the number of waveform points to 128 for the `"new.equ"` equation file.

3. Compile the `"new.equ"` equation file to generate a waveform file named `"new.wfm."` Use the following command

   ```
   EQUATION:COMPILE EXECUTION,"new.equ"
   ```

   to compile the equation file. A waveform file called `"new.wfm"` will be loaded into memory.

4. Select the `"new.wfm"` file for output. Use the following command

   ```
   WAVEFORM "new.wfm"
   ```

   to select the `"new.wfm"` file.

5. Turn on the WAVEFORM OUTPUT connector. Use the following command

   ```
   OUTPUT:STATE ON
   ```

   to turn on the output. The LED beside the WAVEFORM OUTPUT connector turns green to show the output is active.

Compiling Equations

Compiling functions only work on equations that are already in main memory. One way to load an equation is to define it with the `EQUATION:DEFINE` command and then compile it. Once you have compiled the equation file, you can load the resulting waveform file into the VX4792 Arbitrary Waveform Generator with the IBIC command `IBWRTF` (for GPIB systems), or the `WAVeform` command.

***NOTE**. Do not use CR (0x0D) as a terminator in your equation file. Use only LF (0x0A).*

The following procedure illustrates how to set the number of equation points, and how to compile an equation.

1. Before compiling an equation, set the number of equation points with the `EQUATION:WPOINTS` command. For example,

   ```
   EQUATION:WPOINTS "sample.equ",10000
   ```

   sets the number of equation points for the "`sample.equ`" equation to 10,000 points.

2. To begin compilation of the "`sample.equ`" equation, use the `EQUATION:COMPILE EXECUTION` command as shown below:

   ```
   EQUATION:COMPILE EXECUTION,"sample.equ"
   ```

Checking Status

At any time during compilation, you can check status with the `EQUATION:COMPILE:STATE?` query. If the compiler is still working, it will return a 1. A return of 0 indicates that the compiler is finished.

Aborting Compilation

You can stop a compilation in process with the `EQUATION:COMPILE ABORT` command.

When you stop waveform compilation, a warning message is stored in the output queue. To see this message, use the following commands:

```
*ESR?
EVMSG?
```

The response will have the following form:

```
16
:EVMSG 501,"Equation compile has aborted"
```

Transferring Waveforms

This procedure describes the general approach for transferring waveform data from a controller to the waveform generator. The procedure assumes that a waveform file already exists in the computer memory.

1. Specify the file name where the waveform data will be stored within the waveform generator memory with the following command:

```
DATA:DESTination "sample.wfm"
```

where "sample.wfm" is the filename where the data will be stored. The filename must have the appropriate extension (.wfm).

2. Specify output parameters such as the number of points, output voltage, and clock frequency, using the WFMPre command. Select the parameters that apply to your application.

The following example is the complete form of the WFMPre command. Subcommands that have no argument have been omitted.

```
WFMPRE:ENCDG BIN;BN_FMT RP;BYT_OR MSB;BYT_NR 2;BIT_NR 12;
PT_FMT Y;PT_OFF 0;XZERO 0.000;XUNIT "S";YUNIT "V";YOFF 2047;
NR_PT 512;YMULT 0.00122;YZERO 2;XINCR 1E-6;
```

This example selects the following output parameters:

- Number of points: 512 (see NR_PT)
- Amplitude: 5 volts (see YMULT 0.00122)
- Offset: 2 volts (see YZERO 2)
- Clock frequency: 1 MHz (see XINCR 1E-6)

3. Transfer the waveform data with the CURVe command, followed by the waveform block data, as shown in the following example:

```
CURVE #41024.....
```

The block data requires a header string that defines the data length. The data length is a calculated value based on the number of data bytes. Each waveform point consists of 12 bits (two bytes). For the 512-point waveform used in this example, the CURVe command consists of the following:

- The left-most “4” represents the number of digits in the byte count field
- The “1024” represents the number of bytes; this is the byte count field
- The following information (“.....”) is an array of binary data

See page 3–27 for additional details about the CURVe command syntax.

Syntax and Commands

Command Syntax

A large set of commands can be used to control the operations and functions of the VX4792 Arbitrary Waveform Generator from an external controller. The *Command Syntax* section describes the syntax and communication rules for using these commands to operate the waveform generator.

Be sure to read this section before attempting to send commands to your waveform generator.

Command Notation

The command syntax is in extended BNF (Backus-Naur Form) notation. The extended BNF symbols used in the command set are shown in the Table 3–1.

Table 3–1: BNF Symbols and Meanings

Symbol	Meaning
< >	Indicates a defined element
|	Delimits Exclusive OR elements
{ }	Delimits a group of elements one of which the programmer must select
[ ]	Delimits an optional element that the programmer may omit
[ ]...	Delimits an optional element that the programmer may omit or may repeat one or more times
::=	Indicates that the left member is defined as shown by the the right member

Program and Response Messages

Programs created or placed in an external controller are transferred to the VX4792 Arbitrary Waveform Generator as a program message. A program message is a sequence of zero or more program message units delimited by the program message unit delimiter, the semicolon (;).

A program message unit is a set command or query command. The waveform generator performs a function or changes a setting or mode when it receives a set command; when the waveform generator receives a query command, it returns measurement data, settings, status codes and/or status messages. The waveform generator transfers these response messages to the external controller.

Command and Query Structure

Commands are either set commands or query commands (usually just called commands and queries in this manual). Most commands have both a set form and query form. The query form of a command is the same as the set form, except that the query form ends with a question mark.

Figure 3–1 shows a flowchart of the structure of the commands and queries. The structure of the header is described in detail in *Header* on page 3–6.

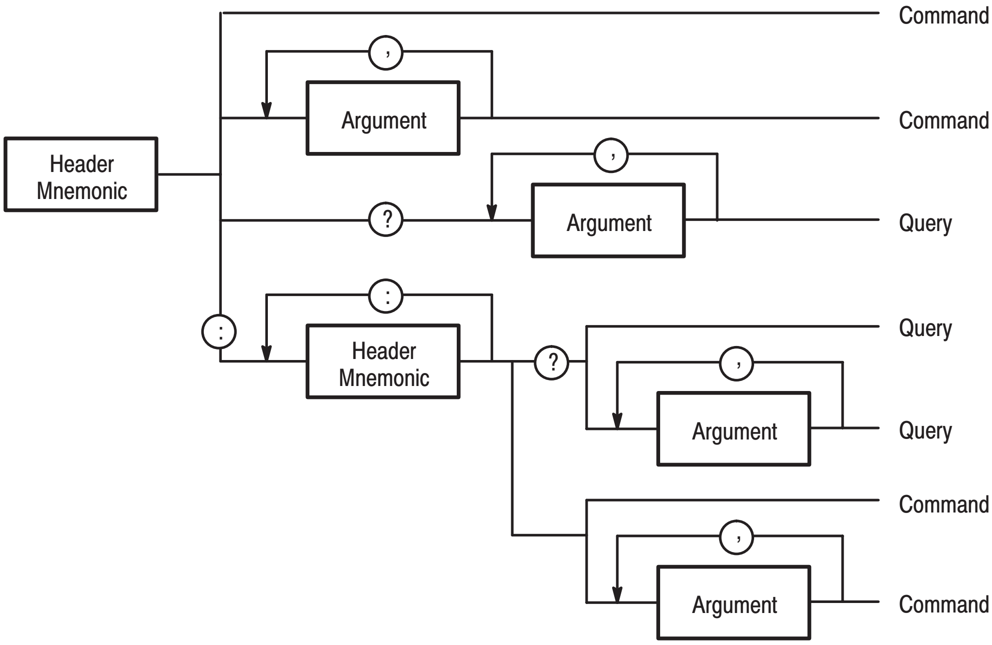


Figure 3–1: Command and Query Structure Flowchart

Character Encoding

The program can be described using the American Standard Code for Information Interchange (ASCII) character encoding.

This seven-bit ASCII code is used for the majority of syntactic elements and semantic definitions. In special cases, an eight-bit ASCII Code is allowed in the arbitrary block arguments described on page 3–6. The ASCII code character set table is found in *Appendix A*.

Syntactic Delimiters

Syntactic elements in a program message unit are delimited (differentiated) with colons, white space, commas, or semicolons.

- **Colon (:).** Typically delimits the compound command header.

  ```
  EQUATION:COMPILE:ABORT
  ```

- **White Space.** Typically delimits command/query headers from the argument.

  ```
  DIAG:SELECT ALL
  MODE BURST,4000
  ```

  `DIAG:SELECT` and `MODE` are the command headers, and `ALL` and `BURST,4000` are the arguments.

- **Comma (,).** Typically delimits between multiple arguments. In the above example, a comma delimits the multiple arguments `BURST` and `4000`.

- **Semicolon (;).** Typically delimits between multiple commands (or multiple program message units). For more information about using the semicolon, refer to *Concatenating Commands* on page 3–8.

White Space

White space, which is used to delimit certain syntactic elements in a command, is defined in the VX4792 Arbitrary Waveform Generator as a single ASCII-encoded byte in the range ASCII 0-32 (decimal). This range consists of the standard ASCII characters exclusively except for ASCII 10, which is the Line Feed (LF) or New Line (NL) character.

Special Characters

The Line Feed (LF) character or the New Line (NL) character (ASCII 10) and all characters in the range of ASCII 127-255 are defined as special characters. These characters are used in arbitrary block arguments only; using these characters in other parts of any command yields unpredictable results.

Arguments

In a command or query, one or more arguments follow the command header. The argument, sometimes called program data, is a quantity, quality, restriction, or limit associated with the command or query header. Depending on the command or query header given, the argument is one of the following types:

- Decimal Numeric
- String
- Arbitrary Block

Decimal Numeric

The VX4792 Arbitrary Waveform Generator defines a decimal numeric argument as one expressed in one of three numeric representations: NR1, NR2, or NR3. This definition complies with that found in ANSI/IEEE Std 488.2-1987. Any commands that use arguments in any of the first three notations can use a fourth notation NRf (for Numerical Representation flexible). The four formats are shown in Table 3–2.

Table 3–2: Decimal Numeric Notation

Type	Format	Examples
NR1	Implicit-point (integer)	1, +3, –2, +10, –20
NR2	Explicit-point unscaled (fixed point)	1, 2, +23.5, –0.15
NR3	Explicit-point scaled (floating point)	1E+2, +3.36E–2, –1.02E+3
NRf	Numeric representation-flexible; any of NR1, NR2, and NR3 may be used	1, +23.5, –1.02E+3

As just implied, you can use NRf notation for arguments in your programs for any commands that this manual lists as using any of NR1, NR2, or NR3 notation in its arguments. Be aware, however, that query response will still be in the format specified in the command. For example, if the command description is `:DESE <NR1>`, you can substitute NR2 or NR3 when using the command in a program. However, if you use the query `:DESE?`, the waveform generator will respond in the format <NR1> to match the command description in this manual.

Unit and SI Prefix

If the decimal numeric argument refers to a voltage, frequency, or percentage, you can express it using SI units instead of the scaled explicit point input value format <NR3>. (SI units are units that conform to the Systeme International d'Unites standard.) For example, you can use the input format 200 mV or 1.0 MHz instead of 200.0E-3 or 1.0E+6, respectively, to specify voltage or frequency.

You can omit the unit, but you must include the SI unit prefix. You can use either uppercase or lowercase units.

V or v for voltage

Hz, HZ, or hz for frequency

PCT, PCt, PcT, Pct, pct, pCT, or pcT, for % (percentage)

The SI prefixes, which must be included, are shown below. Note that either lowercase or uppercase prefixes can be used.

***NOTE**. Note that the prefix m/M indicates 10^{-3} when the decimal numeric argument denotes voltage, but 10^{6} when it denotes frequency.*

SI Prefix	m/M	k/K	m/M
Corresponding Power	10^{-3}	10^{3}	10^{6}

String

String, sometimes referred to as a string literal, a literal, or just a string, is defined as a series of characters enclosed by double quotation marks (”) as in:

“This is a string constant”

or:

“0 .. 127”

To include a double quoted character in the string, insert an additional double quote character ahead of the double quote character in the string. For example, the string:

serial number “B010000”

would be defined as:

“serial number ““B010000”””

Single quotation marks (’) can also be used instead of double quotation marks as in this example:

‘serial number ‘‘B010000’’’

String constants may be of any length up to the memory limits of the waveform generator in which the message is parsed.

Arbitrary Block

An arbitrary block argument is defined in one of these ways:

#<byte count digit><byte count>[<contiguous eight-bit data byte>]...

or:

#<contiguous 8-bit data byte]... <terminator>

where:

`<byte count digit>`::= a non-zero digit in the range ASCII 1-9 that defines the number of digits (bytes) in the `<byte count>` field.

`<byte count>::=` any number of digits in the range ASCII 0-9 that define how many bytes are in the `<contiguous 8-bit data byte>` field.

`<contiguous 8-bit data byte>`::= a `<byte count>` number of 8-bit bytes in the range ASCII 0-255 that define the message. Each byte defines one character.

`<terminator>`::= a software LF followed by a hardware EOI. For example,

```
#16AB4ZLT<LF><&EOI>
#0EHTGNILEDOM<LF><&EOI>
```

Header

Header Mnemonic

The header mnemonic represents a header node or a header subfunction. The command or query header comprises one or more header mnemonics that are delimited with the colon (:).

Marker Representation

In a command or query, a marker can be specified with the header mnemonics MARKER<x>. CH<x> can be CH1 for channel 1. Similarly, MARKER<x> can be either MARKER1 for marker 1 and MARKER2 for marker 2.

Header Structure

Commands and queries can be structured into six basic forms.

- Simple command header
- Simple query header
- Compound command header
- Compound query header
- Common command header
- Common query header

Figure 3–1 on page 3–2 shows the syntax for all possible structures, and an explanation of each of the six basic forms follows.

Simple Command Header. A command that contains only one header mnemonic or only one header mnemonic, plus one or more arguments. Its message format is:

```
[:]<Header Mnemonic> [<Argument>[,<Argument>]...]
```

such as:

`START` or `STOP`

Simple Query Header. A command that contains only one header mnemonic followed by a question mark (?). Its message format is:

```
[:]<Header Mnemonic>? [<Argument>[,<Argument>]...]
```

such as:

`MEMORY?` or `TRIGGER?`

Compound Command Header. A command that contains multiple header mnemonics plus argument(s). Its message format is:

```
[:]<Header Mnemonic>[:<Header Mnemonic>]...
[<Argument>[,<Argument>]...]
```

such as:

`OUTPUT:STATE ON` or `CLOCK:FREQUENCY 250MHZ`

Compound Query Header. A command that contains multiple header mnemonics followed by a question mark (?). Its message format is:

```
[:]<Header Mnemonic>[:<Header Mnemonic>]...?
[<Argument>[,<Argument>]...]
```

such as:

`TRIGGER:INPUT?` or `MEMORY:CATALOG:ALL?`

Common Command Header. A command that precedes its header mnemonic with an asterisk (*). Its message format is:

```
<Header Mnemonic> [<Argument>[,<Argument>]...]
```

such as:

`*RST`

The common commands are defined by IEEE Std 488.2 and are common to all devices which support IEEE Std 488.2 on the GPIB bus.

Common Query Header. A command that precedes its header mnemonic with an asterisk (*) and follows it with a question mark (?). Its message format is:

```
<Header Mnemonic>? [<Argument>[,<Argument>]...]
```

such as:

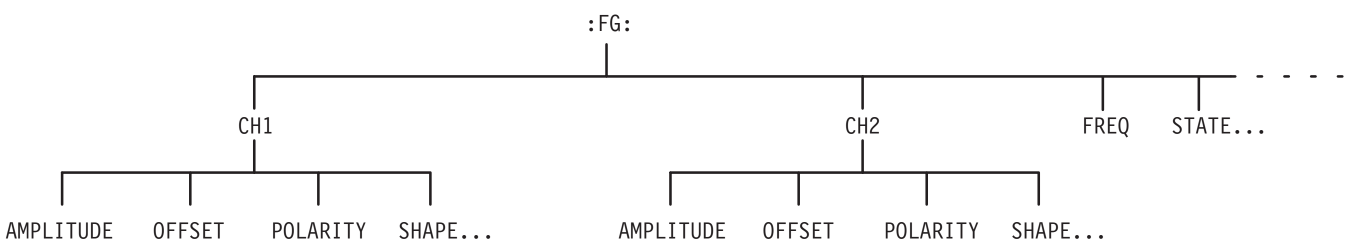
```
*IDN?
```

The common commands are defined by IEEE Std 488.2 and are common to all devices which support the IEEE Std 488.2 on the GPIB bus.

Concatenating Commands

Most of the compound command headers are in a tree structure. The tree structure of an example command is diagrammed below. Note that the top of the structure always begins with a colon (:).

***NOTE**. The following command is NOT in the VX4792 Arbitrary Waveform Generator command set. It is included only as an example of command concatenation.*



The following example of a compound command combines four headers delimited by semicolons:

```
:FG:CH1:AMPLITUDE 3.5; :FG:CH1:OFFSET 1.5;
:FG:CH1:POLARITY INVERTED; :FG:CH1:SHAPE SQUARE
```

You must include the complete path in each header when there is no common complete path to the start of the tree structure (the :). However, note that part of each header in the above example has a common path `:FG:CH1`. You may shorten compound command structures with such headers. For example, the command above may be rewritten as follows.

```
:FG:CH1:AMPLITUDE 3.5; OFFSET 1.5; POLARITY INVERT; SHAPE
SQUARE
```

Note that the mnemonics :FG and :CH1 are assumed from the first header by the headers that follow. The following command descriptions are valid examples of commands shortened using the principle just described. (Note that the insertion of common commands, such as *SRE, between headers does not prevent the headers that follow from assuming the earlier header mnemonics.)

```
:FG:CH1:AMPLITUDE 3.5; OFFSET 1.5; :FG:CH2:AMPLITUDE 3.5;
OFFSET 1.5
```

```
:FG:STATE ON; CH1:SHAPE SQUARE; POLARITY INVERTED
```

```
:FG:CH1:AMPLITUDE 3.5; *SRE; OFFSET 1.5;
POLARITY INVERTED; SHAPE SQUARE
```

The following examples have been shortened incorrectly and cause errors.

```
:FG:CH1:AMPLITUDE 5.0; FG:CH2:AMPLITUDE 5.0
```

```
:FG:CH1:SHAPE SQU; CH2:SHAPE SQUARE
```

```
:FG:CH1:AMPLITUDE 5.0; STATE ON
```

Query Responses

The query causes the VX4792 Arbitrary Waveform Generator to return information about its status or settings. A few queries also initiate an operation action before returning information; for instance, the *CAL? query runs a calibration.

If the programmer has enabled headers to be returned with query responses, the waveform generator formats a query response like the equivalent set-command header followed by its argument(s). When headers are turned off for query responses, only the values are returned. Table 3–3 shows the difference in query responses.

Table 3–3: Header in Query Responses

Query	Header On	Header Off
AMPLITUDE?	AMPLITUDE 5.000 V	5.000 V
DIAG:SELECT?	:DIAG:SELECT WMEMORY	WMEMORY

Use the command HEADER ON (see page 3–42) when you want the header returned along with the information. You can save such a response and send it back as a set-command later. Use HEADER OFF when you want only the information back.

Other General Command Conventions

Upper and Lower Case

The waveform generator accepts upper-, lower-, or mixed-case alphabetic messages. The following three commands are recognized as identical:

`HEADER ON` or `header on` or `Header On`

Abbreviation

Any header, argument, or reserved word that is sent to the waveform generator can be abbreviated. The minimum required spelling is shown in uppercase throughout the subsection *Command Groups* beginning on page 3–13. For example, the command `CLOCk:SOURce INTernal` can be rewritten in either of the following forms:

`CLOCK:SOURCE INTERNAL` or `CLOC:SOUR INT`

Syntax Diagrams

The syntax of each command and query is explained by syntax diagrams as well as the BNF notation. Figure 3–2 shows some typical syntax diagram structures.

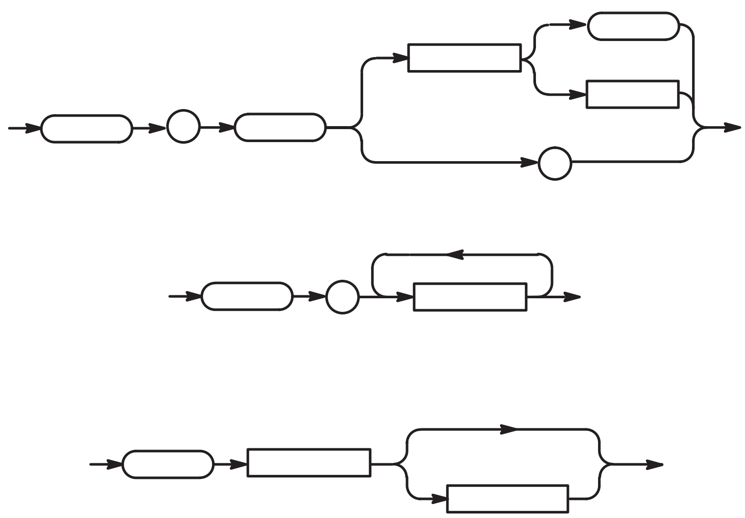


Figure 3–2: Typical Syntax Diagrams

The syntax diagrams are described by the following symbols and notation:

- Oval symbols contain literal elements such as a command or query header and a non-quoted string argument; command name, query name, and nonquoted string argument are abbreviated
- Circle symbols contain separators or special symbols such as (:), (,), and (?)
- Box symbols contain the defined element
- Arrow symbols connect elements to show the paths that can be taken through the diagram and, thereby, the order in which the elements can be sent in a command structure
- Parallel paths show that one and only one of the paths must be taken in the command (see the top diagram of Figure 3–2)
- A loop around an element(s) shows the element can be repeated (see the middle diagram)
- A path around a group of elements shows that those elements are optional (see the bottom diagram.)

***NOTE**. The unit and SI prefix that can be added to decimal numeric arguments are not described in the syntax diagram. See* Units and SI Prefix *on page 3–4.*

Command Groups

The *Command Groups* section divides the organization of the waveform generator command set into functional groups. (See *Command Descriptions* beginning on page 3–21 for a complete description of each command in alphabetical order.)

A command quick reference (page 3–14) that summarizes the commands available for the VX4792 Arbitrary Waveform Generator has been provided for your convenience. We encourage you to duplicate the command quick reference for easy access to the complete list of commands.

Throughout this section, the parenthesized question symbol (?) follows the command header to indicate that both a command and query form are included for the command.

Commands Grouped by Function

Table 3–4 lists the nine functional groups into which the commands are classified.

Table 3–4: Function Groups in the Command Set

Group	Functions Controlled
Calibration and Diagnostic	Control calibration and self-test diagnostics according to selected routines
Memory	Control internal memory operations
Mode	Control operating mode and set trigger parameters
Output	Turn output waveform on and off and select the sync signal position
Setup	Select clock source and its parameters
Status and Event	Set and query the registers and queues of the reporting system
Synchronization	Control operation complete and pending command execution
System	Control miscellaneous instrument functions such as data and time, local lockout, query response forms, and instrument ID
Waveform	Control transfer of waveforms

Command Quick Reference

This page lists all the commands in each functional group and can be copied for use as a quick reference. The minimum accepted character string for each command is in uppercase. The symbol (?) follows the command header of those commands that can be used as either a command or a query; the symbol ? follows those commands that can only be a query. If neither symbol follows the command, it can only be used as a command.

Calibration and Diagnostic Commands

- *CAL?
- DIAG?
- DIAG:RESUlt?
- DIAG:SELect(?)
- DIAG:STATe
- SELFcal?
- SELFcal:RESUlt?
- SELFcal:SELect(?)
- SELFcal:STATe
- *TST?

Memory Commands

- MEMory?
- MEMory:CATalog?
- MEMory:CATalog:ALL?
- MEMory:CATalog:AST?
- MEMory:CATalog:EQU?
- MEMory:CATalog:SEQ?
- MEMory:CATalog:WFM?
- MEMory:COMMent(?)
- MEMory:COPY
- MEMory:DELete
- MEMory:FREE?
- MEMory:FREE:ALL?
- MEMory:LOCk(?)
- MEMory:REName

Mode Commands

- MODE(?)
- RUNNing?
- STARt
- STOP
- *TRG
- TRIGger?
- TRIGger:IMPedance(?)
- TRIGger:INPut(?)
- TRIGger:LEVel(?)
- TRIGger:OUTPut(?)
- TRIGger:POLarity(?)
- TRIGger:SLOPe(?)

Output Commands

- OUTPut?
- OUTPut:STATe(?)
- OUTPut:SYNC(?)

Setup Commands

- AMPLitude(?)
- CH1?
- CLOCk?
- CLOCk:FREQuency(?)
- CLOCk:SOURce(?)
- FILTer(?)
- OFFSet(?)
- OPERation(?)
- WAVeform(?)

Status and Event Commands

- ALLEv?
- *CLS
- DESE(?)
- *ESE(?)
- *ESR?
- EVENT?
- EVMsg?
- EVQty?
- *SRE(?)
- *STB?

Synchronization Commands

- *OPC(?)
- *WAI

System Commands

- HEADer(?)
- ID?
- *IDN?
- *LRN?
- *OPT?
- *RST
- VERBose(?)

Waveform Commands

- AUTOStep:DEFine(?)
- CURVe(?)
- DATA?
- DATA:DESTination(?)
- DATA:ENCDG(?)
- DATA:SOURce(?)
- EQUAtion:COMPile(?)
- EQUAtion:DEFine(?)
- EQUAtion:WPOints(?)
- MARKER<x>:AOFF
- MARKER<x>:POINt(?)
- MARKER:DATA(?)
- SEQUence:DEFine(?)
- SEQUence:EXPand
- WAVFrm?
- WFMPre?
- WFMPre:BIT_NR(?)
- WFMPre:BN_FMT(?)
- WFMPre:BYT_NR(?)
- WFMPre:BYT_OR(?)
- WFMPre:CRVCHK(?)
- WFMPre:ENCDG(?)
- WFMPre:NR_PT(?)
- WFMPre:PT_FMT(?)
- WFMPre:PT_OFF(?)
- WFMPre:WFID(?)
- WFMPre:XINCR(?)
- WFMPre:XUNIT(?)
- WFMPre:XZERO(?)
- WFMPre:YMULT(?)
- WFMPre:YOFF(?)
- WFMPre:YUNIT(?)
- WFMPre:YZERO(?)

Command Summaries

Tables 3–5 through 3–13 describe each command in each of the nine functional groups.

Calibration and Diagnostic Commands

The Calibration and Diagnostic commands perform calibration and self-test diagnostic routines.

Table 3–5: Calibration and Diagnostic Commands

Header	Description
`*CAL?`	Perform calibration
`DIAG?`	Query all current settings related to self-test
`DIAG:RESUlt?`	Query self-test result
`DIAG:SELect(?)`	Select self-test routine
`DIAG:STATe`	Perform self-test
`SELFcal?`	Query all current settings related to calibration
`SELFcal:RESUlt?`	Query calibration result
`SELFcal:SELect(?)`	Select calibration routine
`SELFcal:STATe`	Perform calibration
`*TST?`	Perform self-test

Memory Commands

The Memory commands perform operations on the internal memory, such as renaming a file or returning information about a file.

Table 3–6: Memory Commands

Header	Description
`MEMory?`	Query information on all files and the size of the used and unused memory
`MEMory:CATalog?`	Query information on all files
`MEMory:CATalog:ALL?`	Query information on all files
`MEMory:CATalog:AST?`	Query information on all auto step files
`MEMory:CATalog:EQU?`	Query information on all equation files
`MEMory:CATalog:SEQ?`	Query information on all sequence files
`MEMory:CATalog:WFM?`	Query information on all waveform files
`MEMory:COMMent(?)`	Write a comment into a file in internal memory
`MEMory:COPY`	Copy a file in internal memory

Table 3-6: Memory Commands (Cont.)

Header	Description
MEMory:DELete	Delete a file
MEMory:FREE?	Query the size of the used and unused memory
MEMory:FREE:ALL?	Query the size of the used and unused memory
MEMory:LOCk(?)	Lock a file
MEMory:REName	Change the name of a file in the internal memory

Mode Commands

The Mode commands select the manner in which waveforms are output, such as continuously or in bursts of a certain number of waveform cycles. These commands also generate triggering events for waveforms and set trigger parameters, such as impedance, level, polarity, and slope.

Table 3-7: Mode Commands

Header	Description
MODE(?)	Select waveform output mode
RUNNing?	Query whether a waveform is currently being generated
STARt	Start the waveform output by generating a triggering event
STOP	Stop waveform from being output and initialize for output of another waveform
*TRG	Generate the triggering event (equivalent to STARt)
TRIGger?	Query all current trigger-related settings
TRIGger:IMPedance(?)	Select the impedance presented to the the external trigger signal
TRIGger:INPut(?)	Select the trigger source
TRIGger:LEVel(?)	Set the level on the external trigger signal that generates the triggering event
TRIGger:OUTPut(?)	Select the trigger signal to output to the VXIbus
TRIGger:POLarity(?)	Set the polarity of external signal that generates a triggering event
TRIGger:SLOPe(?)	Select the slope of external signal that generates a triggering event

Output Commands

The Output commands turn the output waveform on or off and select the position on the waveform at which an external sync signal is generated.

Table 3–8: Output Commands

Header	Description
`OUTPut?`	Query all the current settings related to output
`OUTPut:STATe(?)`	Turn the output on or off
`OUTPut:SYNC(?)`	Select position where the sync signal is generated

Setup Commands

The Setup commands are used to set parameters for the clock, such as clock source and frequency, and for the waveform output channel, such as the waveform amplitude or cutoff frequency.

Table 3–9: Setup Commands

Header	Description
`AMPLitude(?)`	Set full scale voltage for the output waveform
`CH1?`	Query all current settings for the output waveform
`CLOCk?`	Query all current settings related to clock
`CLOCk:FREQuency(?)`	Set source clock frequency
`CLOCk:SOURce(?)`	Select clock source
`FILTer(?)`	Select frequency cut-off filter for the output waveform
`OFFSet(?)`	Set offset voltage for the output waveform
`OPERation(?)`	Set the mathematical operation with the external AM signal
`WAVeform(?)`	Specify the output waveform or sequence

Status and Event Commands

The Status and Event commands are used by the external controller to set and query the registers and queues of the VX4792 Arbitrary Waveform Generator event and status reporting system. These commands let the external controller coordinate operation between the waveform generator and other devices on the bus. For the registers and queues described in Table 3–10, refer to the status and event reporting system described in *Status and Events*.

Table 3–10: Status and Event Commands

Header	Description
ALLEv?	Dequeue all events from Event Queue
*CLS	Clear SESR, SBR and Event Queue
DESE(?)	Set and query DESER
*ESE(?)	Set and query ESER
*ESR?	Query SESR
EVENT?	Dequeue event from Event Queue
EVMsg?	Dequeue event from Event Queue
EVQty?	Query number of event on Event Queue
*SRE(?)	Set and query SRER
*STB?	Query SBR

Synchronization Commands

The Synchronization commands are used by the external controller to prevent communication to the VX4792 Arbitrary Waveform Generator from interfering with commands or other operations that the waveform generator is currently executing.

Table 3–11: Synchronization Commands

Header	Description
*OPC(?)	Generate or return the operation complete message
*WAI	Hold off all commands until all pending operations complete

System Commands

The System commands reset the system and return system-related information.

Table 3–12: Synchronization Commands

Header	Description
HEADer(?)	Allow or suppress the return of the control header in response messages
ID?	Query ID information about the waveform generator
*IDN?	Query ID information about the waveform generator
*LRN?	Query all settings of the waveform generator
*OPT?	Query which options are implemented for the waveform generator
*RST	Reset the waveform generator
VERBose(?)	Select short or long response headers

Waveform Commands

The Waveform commands control the transfer of, and parameters related to the transfer of, waveform-related information between the VX4792 Arbitrary Waveform Generator and external controller. This information includes unscaled waveform data, the waveform preamble that specifies how to reconstruct the waveform data, equations defining waveforms, and formats for transferring waveforms. Consider the following points when using waveform commands:

- Waveform data transferred includes only raw, binary-formatted data; the preamble contains the data-encoding format, waveform scale, and related data that allow a scaled waveform to be obtained
- The `CURVe` command or query transfers the unscaled waveform, marker, and sequence data
- The `WAVFrm` command or query transfers both the waveform and the preamble
- The `WFMPre` commands and queries set up the waveform preamble
- The `DATA` commands and queries specify the format and location of the waveform and marker data
- The `EQUAtion` commands define, compile, and otherwise control the conversion of an equation expression into a waveform

Table 3–13: Waveform Commands

Header	Description
AUTOStep:DEFine(?)	Send the auto step data associated with the specified channel to a file in the waveform generator
CURVe(?)	Transmit a waveform between the external controller and the waveform generator
DATA?	Query all current settings related to the waveform or marker data to be transferred
DATA:DESTination(?)	Define the destination to which the waveform is to be transferred
DATA:ENCDG(?)	Select the waveform data transfer format
DATA:SOURce(?)	Designate the source from which waveform is transferred
EQUAtion:COMPile	Compile the equation expression
EQUAtion:DEFine(?)	Write the equation expression into a file
EQUAtion:WPOints(?)	Write a specified number of waveform points
MARKer:DATA(?)	Transmit marker data between the external controller and the waveform generator
MARKER<x>:AOFF	Set all markers to off
MARKER<x>:POINt(?)	Set the marker to the specified point
SEQUence:DEFine(?)	Write a sequence to a file
SEQUence:EXPand	Expand a sequence file into a single waveform file
WAVFrm?	Transmit the waveform preamble and waveform from the waveform generator to the external controller
WFMPre?	Query all the current preamble settings
WFMPre:BIT_NR(?)	Specify the bits of precision per byte
WFMPre:BN_FMT(?)	Specify the binary data format
WFMPre:BYT_NR(?)	Specify the data field width for each binary data point
WFMPre:BYT_OR(?)	Specify the byte order
WFMPre:CRVCHK(?)	Specify the error check method
WFMPre:ENCDG(?)	Set waveform data encoding
WFMPre:NR_PT(?)	Set the number of waveform data points
WFMPre:PT_FMT(?)	Define format of data
WFMPre:PT_OFF(?)	Define the X-axis point offset value
WFMPre:WFID(?)	Set comment and additional information
WFMPre:XINCR(?)	Define the X-axis increment value
WFMPre:XUNIT(?)	Define the X-axis data unit type
WFMPre:XZERO(?)	Define the X-axis origin offset value
WFMPre:YMULT(?)	Define the Y-axis data multiplier value
WFMPre:YOFF(?)	Define the Y-axis offset value
WFMPre:YUNIT(?)	Define the Y-axis data unit type
WFMPre:YZERO(?)	Define the Y-axis origin offset value

Command Descriptions

The *Command Descriptions* section lists each command and query in the waveform generator command set alphabetically. Each command entry includes the command description and command group, related commands (if any), syntax, and arguments. Each entry also includes one or more usage examples.

This section provides the long form of out headers, mnemonics, and arguments with the minimal spelling shown in upper case. For example, to use the abbreviated version of the AUTOStep:DEFine command, just type AUTOS:DEF.

The symbol (?) follows the command header of those commands that can be used as either a command or a query; the symbol ? follows those commands that can only be a query. If neither symbol follows the command, it can only be used as a command.

***NOTE**. For compatibility purposes, the mnemonic "CH1" may be included with the following commands:* AMPlitude, FILTer, AUTOStep:DEFine, OFFSet, OPERation, OUTput:STATe, *and* WAVeform. *The mnemonic "STATE" may be included with the* EQUAtion:COMPile *command.*

ALLEv?

The ALLEv? query dequeues all event codes and their corresponding event messages. Use the *ESR? query to make events available for dequeuing using the ALLEv? query.

Group

STATUS and EVENT

Related Commands

*CLS, DESE, *ESE, *ESR?, EVENT?, EVMsg?, EVQty?, *SRE, *STB?

Syntax

ALLEv?

Arguments

None

Examples

ALLEV?
might return the string: ALLEV 113,"Undefined header; unrecognized command - AMP"; 420, "Query UNTERMINATED".

AMPLitude (?)

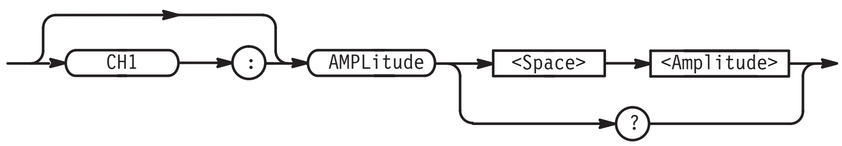
The AMPLitude command sets maximum full scale voltage for the waveform output at the specified channel. The AMPLitude? query returns the maximum voltage currently set.

Group SETUP

Related Commands FILTer, OFFSet, OPERation, WAVeform

Syntax [CH1:]AMPLitude <Amplitude>

[CH1:]AMPLitude?



Arguments <Amplitude>::=<NR2>[<unit>]
where <NR2> has a range of 0.050 V to 5.000 V in steps of 0.001 V and <unit>::={ V | mV }, for volts or millivolts.

Examples AMPLITUDE 230.0mV
sets the amplitude of the waveform to 230 mV.

AUTOStep:DEFine (?)

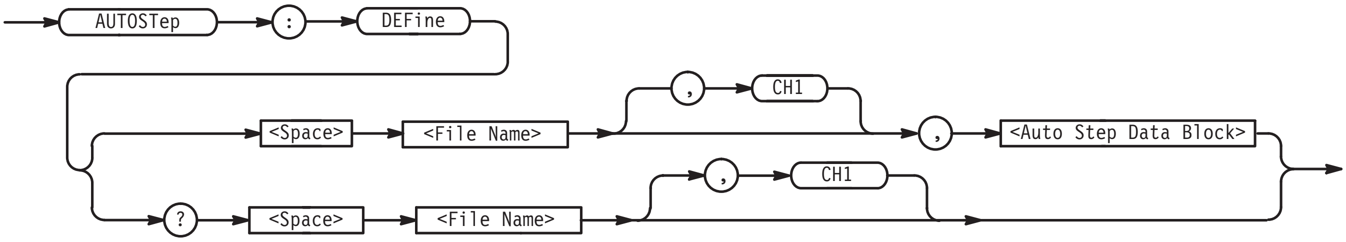
The AUTOStep:DEFine command sends auto step data for the specified channel to a specified file internal to the waveform generator. The AUTOStep:DEFine? query returns the auto step data for the specified channel from the specified file internal to the waveform generator.

Group WAVEFORM

Related Commands None

Syntax AUTOStep:DEFine <File Name>[,CH1],<Auto Step Data Block>

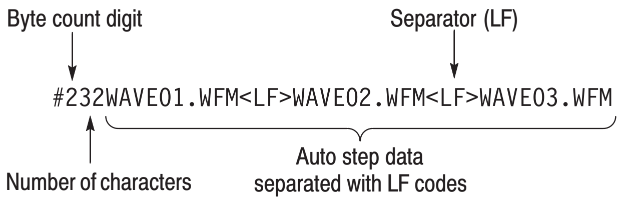
AUTOStep:DEFine? <File Name>[,CH1]



Arguments `<File Name>::=<string>`
which is the name of the file to which the auto step data is transmitted.

`CH1`
This argument is included for compatibility purposes only.

`<Auto Step Data Block>::=<Arbitrary Data Block>`
where `<Arbitrary Data Block>` consists of auto step data written in ASCII code, with the data of each step separated with a Line Feed (LF) code. The following diagram and examples reflect these requirements.



Examples `AUTOSTEP:DEFINE "AUTOS01.AST",`
`#232WAVE01.WFM<LF>WAVE02.WFM<LF>WAVE03.WFM`
sends the auto step data to the file `AUTOS01.AST`.

*CAL?

The `*CAL?` common query performs an internal calibration and returns status that indicates whether the waveform generator completes the self-calibration without error. If an error is detected during calibration, execution immediately stops and an error code is returned.

***NOTE**. The waveform generator does not respond to any commands or queries issued during the self-cal. The self-cal takes up to 30 seconds to complete.*

Group CALIBRATION and DIAGNOSTIC

Related Commands SELFcal:RESUlt, SELFcal:SELect, SELFcal:STATe

Syntax *CAL?

Arguments None

Returns <Result>
where <Result>::=<NR1>, which is one of following decimal integers:

0	Terminated without error
200	Detected errors in the clock unit
600	Detected errors in the setup-related unit

Examples *CAL?
performs an internal calibration and returns the results (for example, it might return 0, which indicates the calibration terminated without any detected errors).

CH1?

The CH1? query returns all current waveform output settings.

Group SETUP

Related Commands AMPLitude, FILTer, OFFSet, OPERation, WAVeform

Syntax CH1?

Arguments None

Returns Returns the settings as a sequence of commands, suitable for sending as set commands later to restore a setup. See *Examples* below.

Examples :CH1?
might return :CH1:AMPLITUDE 1.000;FILTER THRU;OFFSET 0.000;OPERATION NORMAL; WAVEFORM "WAV2400.WFM".

CLOCk?

The CLOCk? query returns all clock settings.

Group SETUP

Related Commands CLOCk:FREQuency, CLOCk:SOURce

Syntax CLOCk?

Arguments None

Examples :CLOCK?
might return CLOCK:FREQUENCY 1.000E+08; SOURCE INTERNAL

CLOCk:FREQuency (?)

The CLOCk:FREQuency command sets source clock frequency. The CLOCk:FREQuency? query returns the frequency currently set.

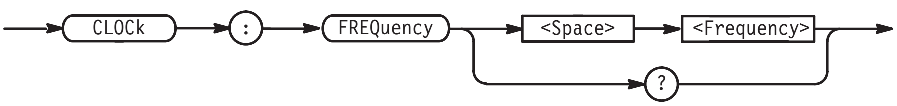
This command is effective only when the internal clock source is selected.

Group SETUP

Related Commands CLOCk:SOURce

Syntax CLOCk:FREQuency <Frequency>

CLOCk:FREQuency?



Arguments

`<Frequency>::=<NR3>[<unit>]`
where `<NR3>` is a decimal number that combines with `[<unit>]` to have a range of 10.0 Hz to 250.0E+6 Hz, and `[<unit>]::={ HZ | KHZ | MHZ }`, for hertz, kilohertz, or megahertz.

Examples

`:CLOCK:SOURCE INTERNAL;FREQUENCY 245.0KHZ`
selects internal clock as a clock source and sets the frequency to 245 kHz.

CLOCk:SOURce (?)

The `CLOCk:SOURce` command selects clock source. The `CLOCk:SOURce?` query returns the currently selected clock source.

Group

SETUP

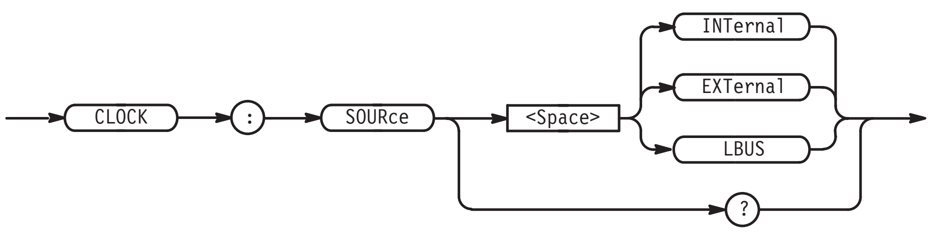
Related Commands

`CLOCk?`, `CLOCk:FREQuency`

Syntax

`CLOCk:SOURce { INTernal | EXTernal | LBUS }`

`CLOCk:SOURce?`



Arguments

`INTernal` — use the internal clock source

`EXTernal` — use the external clock source supplied through the external connector

`LBUS` — use the local bus clock source supplied by another module

Examples

`:CLOCK:SOURCE EXTERNAL`
selects the external clock source.

*CLS

The *CLS common command clears the SESR (Standard Event Status Register), the SBR (Status Byte Register), and the Event Queue, which are used in the waveform generator status and event reporting system. For more details, refer to *Status and Events*.

Group STATUS and EVENT

Related Commands DESE, *ESE, *ESR?, *EVENT?, EVMsg?, EVQty?, *SRE, *STB?

Syntax *CLS

Examples *CLS
clears the SESR, the SBR, and the Event Queue.

CURVe (?)

The CURVe command transmits unscaled, binary-formatted waveform data from an external controller to the location inside the waveform generator specified with the DATA:DESTination command.

The CURVe? query transmits unscaled data for a waveform in binary format to the external controller from the source located inside the waveform generator specified with the DATA:SOURce command.

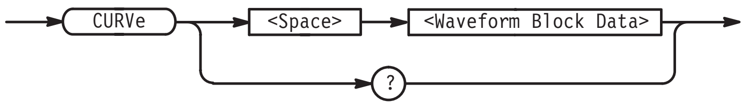
Each waveform transmitted should have an associated waveform preamble that contains information such as data format, scale, and other information that the waveform generator uses to construct a scaled waveform from the waveform data.

Group WAVEFORM

Related Commands WAVFrm?, WFMPre?, DATA:SOURce, DATA:DESTination, DATA:ENCDG

Syntax

CURVe <Waveform Block Data>
CURVe?



Arguments

<Block Data>::=<Arbitrary Data>
where <arbitrary data> is the unscaled waveform data in binary format. The data consists of the lower 12 bits of each two bytes. The value is expressed as a signed binary with offset $0x0800_{16}$. So $0x0800_{16}$ equals zero level, $0x0FFF_{16}$ equals positive maximum level, and $0x0000_{16}$ equals negative maximum level.

Examples

:CURVE #3256...
transmits an unscaled waveform to the waveform generator. The block data element #3256 indicates that 256 bytes of binary data are to be transmitted.

DATA?

The DATA? query returns all settings related to the data command currently in effect for waveform or marker transfer.

Group

WAVEFORM

Related Commands

DATA:DESTination, DATA:ENCDG, DATA:SOURce

Syntax

DATA?

Arguments

None

Returns

The response is returned in a continuous format (see *Examples* below)

Examples

DATA?
might return :DATA:DESTINATION "WAVESP.WFM";ENCDG RPBINARY;SOURCE "CH1".

DATA:DESTination (?)

The DATA:DESTination command specifies the destination within the waveform generator to which the waveform or the marker data is transmitted and stored using CURVe or MARKer:DATA commands.

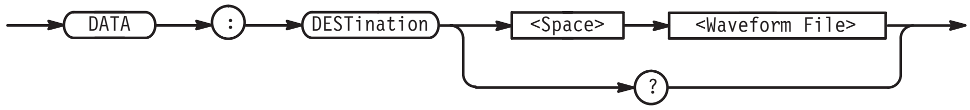
The DATA:DESTination? query returns the destination currently specified.

Group WAVEFORM

Related Commands CURVe, MARKER<x>:AOFF, MARKER<x>:POINt, MARKer:DATA

Syntax DATA:DESTination <Waveform File>

DATA:DESTination?



Arguments <Waveform File>::=<string>
where <string> must be the name of a waveform file to be transferred into the internal memory of the waveform generator. If the waveform file name specified already exists in internal memory, the file is overwritten. Also, if the overwritten file contains a waveform currently loaded and output on a channel, transmitting the new waveform replaces the current waveform at the channel output as well as in the file.

Examples :DATA:DESTINATION "WAVE_EXT.WFM"
specifies the waveform file: WAVE_EXT.WFM as a destination.

DATA:ENCDG (?)

The DATA:ENCDG command sets the encoding format for the waveform transferred using the CURVe command or WAVFrm command.

The DATA:ENCDG? query returns the waveform encoding format currently set.

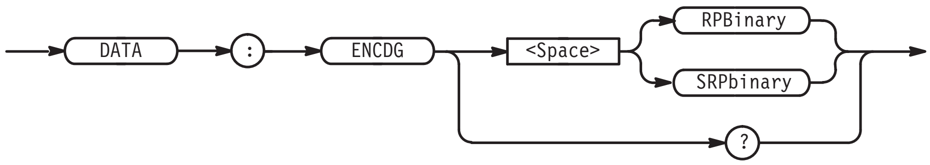
Group WAVEFORM

Related Commands CURVe, WAVFrm?, WFMPre:ENCDG, WFMPre:BYT_OR

Syntax

DATA:ENCDG { RPBinary | SRPbinary }

DATA:ENCDG?



Arguments

RPBinary — specifies positive integer data point representation with the most significant byte transferred first

SRPbinary — specifies positive integer data point representation with the least significant byte transferred first

Examples

:DATA:ENCDG RPBINARY
specifies the format RPBinary, which is described under *Arguments* above.

DATA:SOURce (?)

The DATA:SOURce command specifies the waveform generator source (channel or waveform file) from which the waveform is transmitted to an external controller using the CURVe? query.

The DATA:SOURce? query returns the source that is currently specified.

Group

WAVEFORM

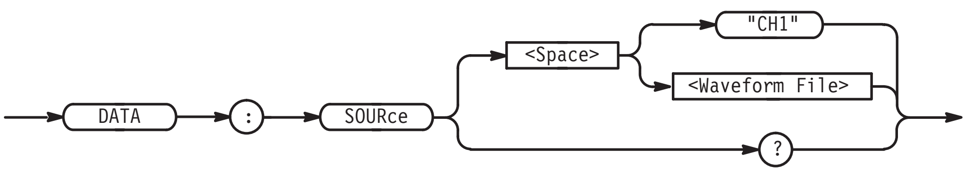
Related Commands

CURVe?

Syntax

DATA:SOURce { "CH1" | <Waveform File> }

DATA:SOURce?



Arguments

<Waveform File>::=<string>
where the string is "CH1" or the name of a waveform file located in internal memory. No other source strings are allowed.

Examples

:DATA:SOURCE "CH1"
specifies the current waveform as the source.

DESE (?)

The DESE command sets the bits of the DESER (Device Event Status Enable Register) used in the status and event reporting system of the waveform generator. The DESE? query returns the contents of the DESER. Refer to *Status and Events* for more information about DESE.

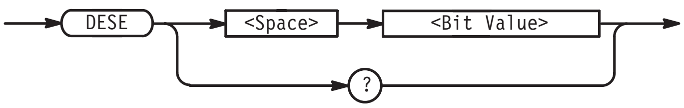
The power-on default for the DESER sets all bits to 1 if the power-on status flag is TRUE. If this flag is set to FALSE, the DESER maintains the current value through a power cycle.

Group STATUS and EVENT

Related Commands *CLS, *ESE, *ESR?, EVENT?, EVMsg?, EVQty?, *SRE, *STB?

Syntax DESE <Bit Value>

DESE?



Arguments <Bit Value>::=<NR1>
where <NR1> is a decimal integer within the range from 0 to 255, that sets the DESER bits to the binary equivalent.

Examples :DESE 177
sets the DESER to 177 (binary 10110001), which sets the PON, CME, EXE and OPC bits.

:DESE?
might return :DESE 176, which indicates that the DESER contains the binary number 10110000.

DIAG?

The `DIAG?` query returns the selected self-test routine(s), runs the routine, and returns the results.

Group CALIBRATION and DIAGNOSTIC

Related Commands `DIAG:SELect`, `DIAG:STATe`, `DIAG:RESUlt?`

Syntax `DIAG?`

Arguments None

Responses `:DIAG:SELECT <Self-test Routine>; [RESULT],<Result>[,<Result>]...`

`<Self-test Routine>::= <label>`
where `<label>` is one of following routines:

`ALL`	all routines
`CPU`	CPU unit check routine
`SETUp`	setup related unit check routine
`WMEMory`	waveform memory check routine

and where `<Result>::=<NR1>` is one of following responses:

`0`	terminated without error
`100`	detected an error in the CPU unit
`600`	detected an error in the setup-related unit
`700`	detected an error in the waveform memory unit

***NOTE**. The waveform generator does not respond to any commands or queries issued during the self-test. The self-test takes up to 90 seconds to complete.*

Examples `DIAG?`
might return `:DIAG:SELECT ALL;RESULT 0`.

DIAG:RESUlt?

The `DIAG:RESUlt?` query returns results of self-test execution.

Group CALIBRATION and DIAGNOSTIC

Related Commands `DIAG:SELect`, `DIAG:STATe`

Syntax `DIAG:RESUlt?`

Arguments None

Returns `:DIAG:RESULT<Result>[,<Result>]...<Result>::=<NR1>`
where `<NR1>` is one of following values:

0	terminated without error
100	detected an error in the CPU unit
600	detected an error in the setup-related unit
700	detected an error in the waveform memory unit

Examples `:DIAG:RESULT?`
might return `:DIAG:RESULT 100`

DIAG:SELect (?)

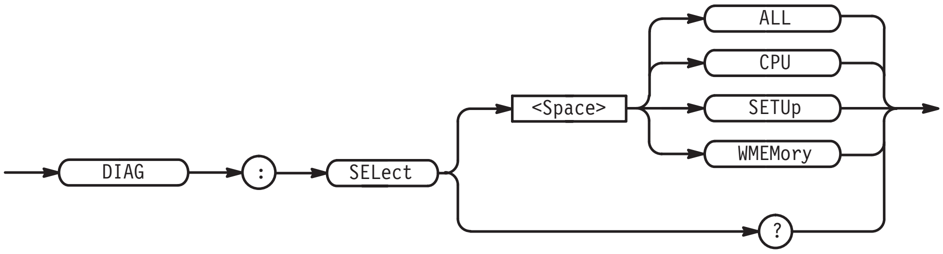
The `DIAG:SELect` command selects the self-test routine. The `DIAG:SELect?` query returns currently selected routine. The `DIAG:STATe` command executes the routine.

Group CALIBRATION and DIAGNOSTIC

Related Commands `DIAG:STATe`, `DIAG:RESULt?`

Syntax `DIAG:SELect { ALL | CPU | SETUp | WMEMory }`

`DIAG:SELect?`



Arguments

ALL	checks all routines that follow
CPU	checks the CPU unit
SETUp	checks the unit for setup
WMEMory	checks the waveform memory

Examples `:DIAG:SELECT CPU;STATE EXECUTE`
executes the CPU self-test routine.

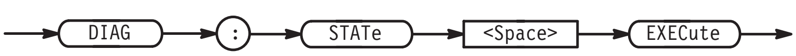
DIAG:STATe

The `DIAG:STATe` command executes the self-test routine(s) selected with the `DIAG:SELect` command. If an error is detected during execution, the routine that detected the error terminates. If all of the self-test routines are selected using the `DIAG:SELect` command, self-testing continues with execution of the next self-test routine.

Group CALIBRATION and DIAGNOSTIC

Related Commands `DIAG:SELect`, `DIAG:RESUlt?`

Syntax `DIAG:STATe EXECute`



Arguments `EXECute` — performs the self-test using the selected routine

Examples `:DIAG:SELECT ALL;STATE EXECUTE;RESULT?`
executes all of the self-tests routines. After all self-test routines finish, the results of the self-tests are returned.

EQUAtion:COMPile (?)

The `EQUAtion:COMPile` command compiles the specified equation expression into a waveform and stores the resulting waveform in a waveform file, or stops compilation of the equation file.

The query form returns the status of the compilation.

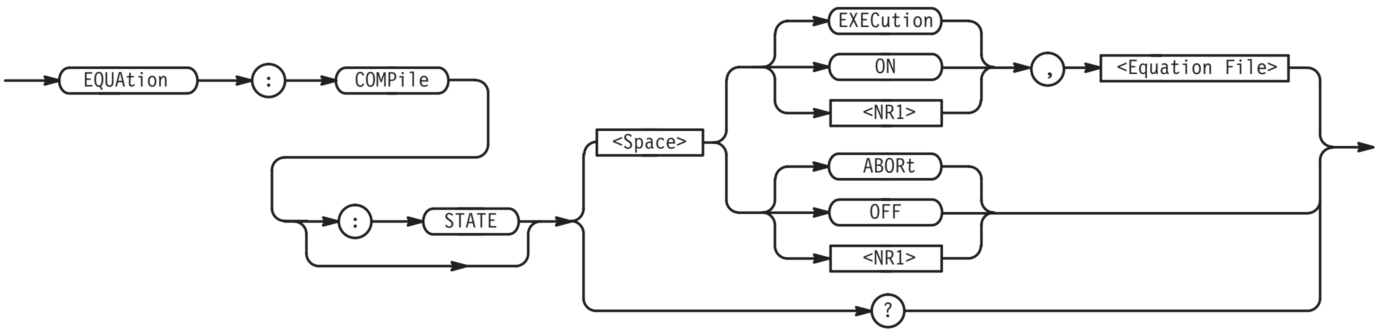
Group WAVEFORM

Related Commands `EQUAtion:DEFine`, `EQUAtion:WPOints`

Syntax `EQUAtion:COMPile[:STATE] { EXECution | ON | <NR1> },<Equation File>`

`EQUAtion:COMPile[:STATE] { ABORt | OFF | <NR1> }`

`EQUAtion:COMPile[:STATE]?`



Arguments `<Equation File>::=<string>`
where `<string>` must be the name of an equation file in the internal memory of the waveform generator. The waveform file that results is given the base name of the equation file.

`EXECution`, `ON`, or `<NR1> (any number except 0)` — compile the specified expression

`ABORt`, `OFF`, or `0` — stop compiling

Examples `EQUATION:COMPILE EXECUTE, "EXP_SAMP.EQU"`
compiles the equation expression in the equation file `EXP_SAMP.EQU` and stores the results in file `EXP_SAMP.WFM`.

`EQUATION:COMPILE ABORT`
aborts compilation. No waveform file is generated.

EQUATION:COMPILE?
might return :EQUATION:COMPILE 0, "sample.equ" if the compilation has already been done, or :EQUATION:COMPILE 1, "sample.equ" if the compilation is still working.

EQUAtion:DEFine (?)

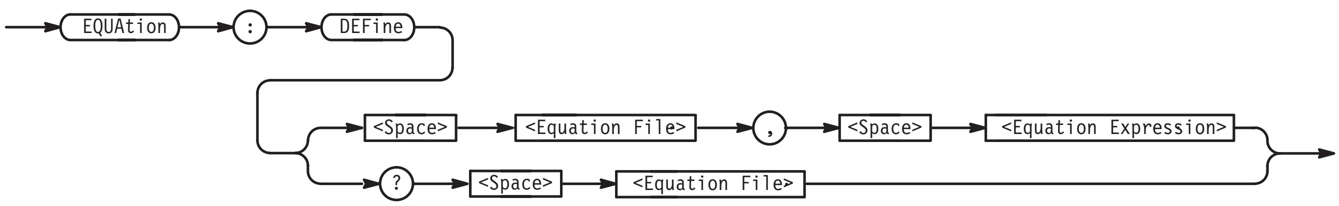
The EQUAtion:DEFine command writes an equation expression into the specified equation file. The EQUAtion:DEFine? query returns the equation expression that is stored in the specified equation file.

Group WAVEFORM

Related Commands EQUAtion:COMPile, EQUAtion:WPOints

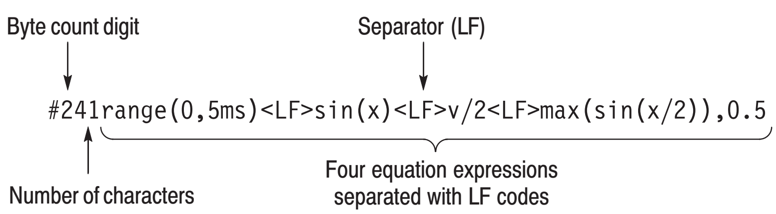
Syntax EQUAtion:DEFine <Equation File>, <Equation Expression>

EQUAtion:DEFine? <Equation File>



Arguments <Equation File>::=<string>
where <string> must be the name of an equation file to be stored in internal memory.

<Equation Expression>::=<Arbitrary Data>
where the <Arbitrary Data> for the equation expression must be written in ASCII code with each expression separated by a Line Feed (LF) code as follows:



Examples

:EQUATION:DEFINE "EXP_SAMP.EQU", #241range(0.5ms)
<LF>sin(x)<LF>v/2<LF>max(sin ...
writes an equation expression into the equation file EXP_SAMP.EQU.

EQUAtion:WPOints (?)

The EQUAtion:WPOints command specifies the number of waveform points, from the equation file, to be written to the waveform file when an equation file is compiled. The EQUAtion:WPOints? query returns the number of waveform points set to be written to the equation file.

Group

WAVEFORM

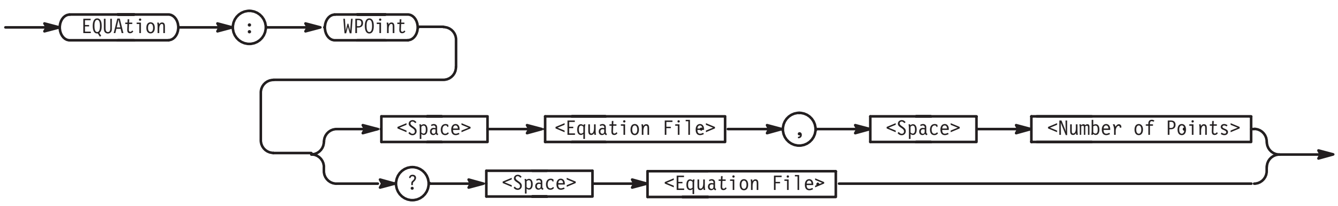
Related Commands

EQUAtion:COMPile, EQUAtion:DEFine

Syntax

EQUAtion:WPOint <Equation File>, <Number of Points>

EQUAtion:WPOints? <Equation File>



Arguments

<Equation File>::=<string>
where <string> must be the name of an equation file in internal memory.

<Number of Points>::=<NR1>
where <NR1> must be in the range of 1 to 32768 (32 K).

Examples

:EQUATION:WPOINTS "EXP_SAMP.EQU", 1000
specifies 1000 as a number of waveform points to be written to the file EXP_SAMP.EQU.

*ESE (?)

The *ESE common command sets the bits of the ESER (Event Status Enable Register) used in the status and events reporting system of the waveform generator. The *ESE? query returns the contents of the ESER. Refer to *Status and Events* for more information about the ESER.

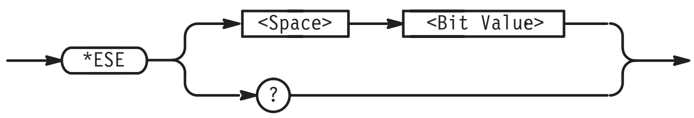
If the power-on status flag is TRUE, the power-on default for the ESER will reset all bits to zero. If this flag is set to FALSE, the ESER bits do not change value during the power-on cycle.

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESR?, EVENT?, EVMsg?, EVQty?, *SRE, *STB?

Syntax *ESE <Bit Value>

*ESE?



Arguments <Bit Value>::=<NR1>
where <NR1> is a decimal integer within the range from 0 to 255. The ESER bits will be set to the binary equivalent of the decimal integer sent.

Examples *ESE 177
sets the ESER to 177 (binary 10110001), which sets the PON, CME, EXE, and OPC bits.

*ESE?
might return 176,which indicates that the ESER contains the binary number 11010000.

*ESR?

The *ESR? common query returns the contents of the SESR (Standard Event Status Register) used in the status and events reporting system. Refer to *Status and Events* for more information about *ESR? or SESR.

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESE?, EVENT?, EVMsg?, EVQty?, *SRE, *STB?

Syntax *ESR?

Arguments None

Examples *ESR?
might return 181, which indicates that the SESR contains the binary number 10110101.

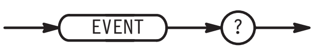
EVENT?

The EVENT? query dequeues the event code of the event that has been in the Event Queue for the longest time period. Use the *ESR? query to make the events available for dequeuing using EVENT?. Refer to *Status and Events* for more information about *EVENT? or ESR?.

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESE, *ESR?, EVMsg?, EVQty?, *SRE, *STB?

Syntax EVENT?



Arguments None

Examples EVENT?
might return :EVENT 113

EVMsg?

The EVMsg? query dequeues the event code and event message of the event that has been in the Event Queue for the longest time period. Use the *ESR? query to make the events available for dequeuing using EVMsg? For more details, refer to *Status and Events.*

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESE, *ESR?, EVENT?, EVQty?, *SRE, *STB?

Syntax EVMsg?

Arguments None

Examples :EVMSG?
might return :EVMSG 420,"Query UNTERMINATED".

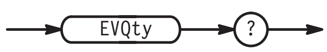
EVQty?

The EVQty? query returns the number of events currently stacked in the Event Queue. If no event is being queued, 0 is returned.

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESE, *ESR, EVMsg?, EVENT?, *SRE, *STB?

Syntax EVQty?



Arguments None

Examples :EVQty?
might return :EVQTY 5.

FILTer (?)

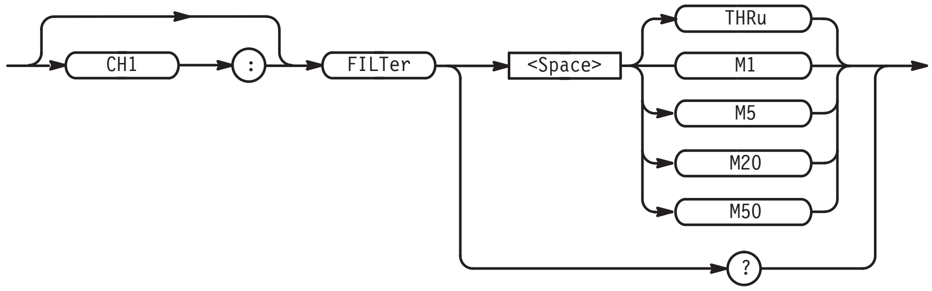
The `FILTer` command selects one of four low pass filters, or no filter. The `FILTer?` query returns the name of the currently selected filter.

Group SETUP

Related Commands `AMPLitude`, `OFFSet`, `OPERation`, `WAVeform`

Syntax `[CH1:]FILTer { THRu | M1 | M5 | M20 | M50 }`

`[CH1:]FILTer?`



Arguments

THRu	OFF (no filter is used)
M1	1 MHz
M5	5 MHz
M20	20 MHz
M50	50 MHz

Examples `FILTER M20`
selects a low-pass filter that rolls off frequencies above a 20 MHz cut off frequency.

HEADer (?)

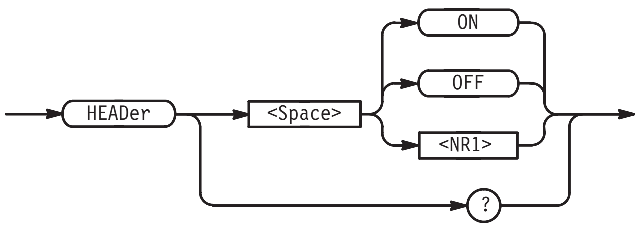
The `HEADer` command enables or disables the command header responses to all queries except IEEE Std 488.2 common commands. The `HEADer?` query returns the status indicating whether the command header responses are enabled or not.

Group SYSTEM

Related Commands `VERBose`

Syntax `HEADer { ON | OFF | <NR1> }`

`HEADer?`



Arguments `ON` or `non-zero value` — enables the command header responses
`OFF` or `zero value` — disables the command header responses

Returns

1	command header responses are currently enabled
0	command header responses are currently disabled

Examples `:HEADER OFF`
disables the command header responses.

`:HEADER?`
might return 1 which indicates command headers are currently enabled for return in query responses.

ID?

The ID? query returns the ID information of the waveform generator.

Group SYSTEM

Related Commands `*IDN?`

Syntax `ID?`

Arguments None

Returns `ID <Manufacturer>/<Model>`, `<Firmware Level>`
where
`<Manufacturer>::=SONY_TEK`,
`<Model>::=VX4792`,
`<Firmware Level>::=CF:<Code and Format Version>`, and
`FV:<Firmware Version>`.

Examples `:ID?`
returns `ID SONY_TEK/VX4792,CF:91.1CT,FV:1.0r1.0`

*IDN?

The `*IDN?` common query returns the ID information of the waveform generator.

Group SYSTEM

Related Commands `ID?`

Syntax `*IDN?`

Arguments None

Returns

`<Manufacturer>`, `<Model>`, `<Serial Number>`, `<Firmware Level>`
where
`<Manufacturer>::=SONY/TEK`,
`<Model>::=VX4792`,
`<Serial Number>::=0`,
`<Firmware Level>::=CF:<Code and Format Version>`,
`<sp>FV:<Firmware Version>`, and
`<sp>::= Space`.

Examples

`*IDN?`
might return `SONY/TEK,VX4792,0,CF:91.1CT FV:1.0r1.0`

*LRN?

The `*LRN?` common query returns all current settings for the waveform generator. The settings returned are in the format of a sequence of commands. If you save this query response, you can send it back later as a command sequence to reestablish the saved settings.

Group

SYSTEM

Related Commands

None

Syntax

`*LRN?`

Arguments

None

Returns

Returns the settings as a sequence of commands, suitable for sending as set commands later to restore a setup (see *Examples* below)

Examples

`*LRN?`
might return the following response:

```
:HEADER 1;:VERBOSE 1;:DIAG:SELECT ALL;:SELFCAL:SELECT ALL;
:MODE CONTINUOUS;:TRIGGER:IMPEDANCE HIGH;LEVEL 1.400;POLARITY
POSITIVE;SLOPE POSITIVE;INPUT INTERNAL;OUTPUT OFF;:WAVEFORM
"";:AMPLITUDE 1.000;:FILTER THRU;:OFFSET 0.000;:OPERATION
NORMAL;:CLOCK:FREQUENCY 1.000E+08;SOURCE INTERNAL;:OUTPUT:STATE
0;SYNC END;:DATA:DESTINATION "GPIB.WFM";ENCDG RPBINARY;SOURCE
"CH1"
```

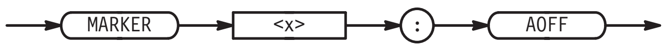
MARKER<x>:AOFF

The MARKER<x>:AOFF command resets all markers (1 or 2) in the file specified by the DATA:DESTination command.

Group WAVEFORM

Related Commands MARKER<x>:POINt, MARKer:DATA, DATA:DESTination

Syntax MARKER<x>:AOFF



Arguments None

Examples :DATA:DESTINATION "WAVE01.WFM";:MARKER1:AOFF
resets all marker 1 values in the file WAVE01.WFM.

MARKER<x>:POINt (?)

The MARKER<x>:POINt command sets or resets the marker (1 or 2) specified, at the data position specified, in the file specified using the DATA:DESTination command.

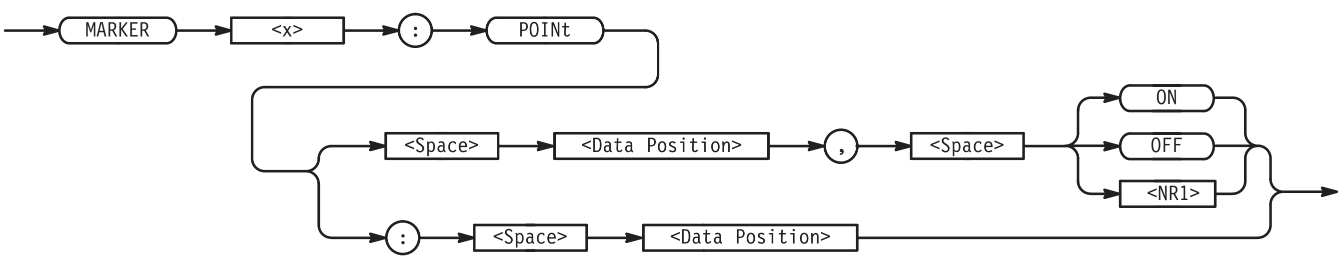
The MARKER<x>:POINt? query returns marker data state at the specified data position specified in the file specified using the DATA:SOURce command.

Group WAVEFORM

Related Commands MARKER<x>:AOFF, MARKer:DATA, DATA:DESTination, DATA:SOURce

Syntax MARKER<x>:POINt <Data Position>, { OFF | ON | <NR1> }

MARKER<x>:POINt? <Data Position>



Arguments

`<Data Position>::=<NR1>`, `ON`, or `OFF`

where <NR1> is a decimal integer,
ON or non-zero sets a marker at <Data Position>, and
OFF or zero value resets a marker at <Data Position>

Examples

`:DATA:DESTINATION "WAVE01.WFM" ;:MARKER1:POINT 2001, ON`
sets marker 1 at 2001st data point in the file WAVE01.WFM.

`:DATA:SOURce "WAVE02.WFM" ;:MARKER1:POINT? 1400`
might return `:MARKER1:POINT 1400, 1`

MARKer:DATA (?)

The MARKer:DATA command writes marker data to the file specified with the DATA:DESTination command.

The MARKer:DATA? query returns marker data written in the file specified with the DATA:SOURce command.

Group

WAVEFORM

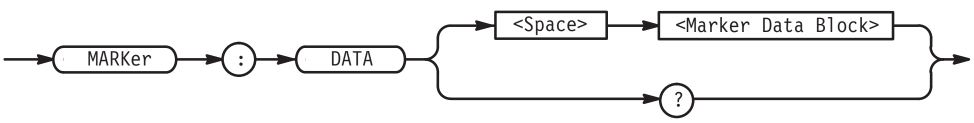
Related Commands

MARKER<x>:AOFF, MARKER<x>:POINt, DATA:DESTination, DATA:SOURce

Syntax

`MARKer:DATA <Marker Data Block>`

`MARKer:DATA?`



Arguments

`<Marker Data Block>::=<Arbitrary Block>`

Binary Data	Description
0x00	Turn off marker 1 and marker 2
0x01	Turn on marker 2 and turn off marker 1
0x02	Turn on marker 1 and turn off marker 2
0x03	Turn on marker 1 and marker 2

Examples

`:MARKER:DATA #41000`
sets marker data.

MEMory?

The MEMory? query returns file-specific information on all files in the internal memory, and used size and unused size of the internal memory. This query is equivalent to sending the MEMory:CATalog:ALL? query followed by the MEMory:FREE:ALL? query.

Group MEMORY

Related Commands MEMory:CATalog:ALL?, MEMory:FREE:ALL?

Syntax MEMory?

Arguments None

Returns :MEMORY:CATALOG:ALL<File Entry>[,<File Entry>]...;
:MEMORY:FREE:ALL<Unused Size>, <Used Size>
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>
<File Name>::=<string>,
<File Size>::=<NR1>,
<Time Stamp>::=<string>,
<Unused Size>::=<NR1>, and
<Used Size>::=<NR1>.

Examples :MEMORY?
might return the following response:
:MEMORY:CATALOG:ALL "AUTOSTEP.AST",142,"92-04-23 16:49","EQUATION.EQU",296,"92-04-23 16:54","SEQUENCE.SEQ",960,"92-04-23 16:48","WAVE2.WFM", 2948,"92-04-23 16:47","WAVEFORM.WFM", 2948,"92-04-23 16:47";:MEMORY:FREE:ALL 1696220,28500

MEMory:CATalog?

The `MEMory:CATalog?` query returns file-related information about all files in internal memory. This query is equivalent to the `MEMory:CATalog:ALL?` query.

Group

MEMORY

Related Commands

`MEMory:CATalog:ALL?`, `MEMory?`

Syntax

`MEMory:CATalog?`

Arguments

None

Returns

:Memory:Catalog:All<File Entry>[,<File Entry>]...
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>,
<File Name>::=<string>,
<File Size>::=<NR1>, and
<Time Stamp>::=<string>

Examples

`:MEMORY:CATALOG?`
might return the following response:
`:MEMORY:CATALOG:ALL "AUTOSTEP.AST",142,"92-04-23 16:49","EQUATION.EQU",296,"92-04-23 16:54","SEQUENCE.SEQ",960,"92-04-23 16:48","WAVE2.WFM",2948, "92-04-23 16:47","WAVEFORM.WFM",2948,"92-04-23 16:47"`

MEMory:CATalog:ALL?

The `MEMory:CATalog:ALL?` query returns file-related information about all files in the internal memory.

Group MEMORY

Related Commands `MEMory:CATalog?`, `MEMory:CATalog:AST?`, `MEMory:CATalog:EQU?`, `MEMory:CATalog:SEQ?`, `MEMory:CATalog:WFM?`, `MEMory?`

Syntax `MEMory:CATalog:ALL?`

Arguments None

Returns `[:MEMORY:CATALOG:ALL]<File Entry>[,<File Entry>]...`
where
`<File Entry>::=<File Name>`, `<File Size>`, `<Time Stamp>`,
`<File Name>::=<string>`,
`<File Size>::=<NR1>`, and
`<Time Stamp>::=<string>`

Examples `:MEMORY:CATALOG:ALL?`
might return the following response:

```
:CATALOG:ALL "AUTOSTEP.AST",142,
"92-04-23 16:49","EQUATION.EQU",296,"92-04-23 16:54",
"SEQUENCE.SEQ",960,"92-04-23 16:48","WAVE2.WFM",2948,
"92-04-23 16:47","WAVEFORM.WFM",2948,"92-04-23 16:47"
```

MEMory:CATalog:AST?

The `MEMory:CATalog:AST?` query returns file-related information about all autostep files in the internal memory of the waveform generator.

Group MEMORY

Related Commands `MEMory:CATalog:ALL?`, `MEMory?`

Syntax `MEMory:CATalog:AST?`

Arguments None

Responses :MEMORY:CATALOG:AST<File Entry>[,<File Entry>]...
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>,
<File Name>::=<string>,
<File Size>::=<NR1>, and
<Time Stamp>::=<string>

Examples :MEMORY:CATALOG:AST?
might return :MEMORY:CATALOG:AST "AUTOSTEP.AST",142,"92-04-23 16:49"

MEMory:CATalog:EQU?

The MEMory:CATalog:EQU? query returns file-related information about all equation files in the internal memory of the waveform generator.

Group MEMORY

Related Commands MEMory:CATalog:ALL?, MEMory?

Syntax `MEMory:CATalog:EQU?`

Arguments None

Returns :MEMORY:CATALOG:EQU<File Entry>[,<File Entry>]...
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>,
<File Name>::=<string>,
<File Size>::=<NR1>, and
<Time Stamp>::=<string>.

Examples :MEMORY:CATALOG:EQU?
might return :MEMORY:CATALOG:EQU "EQUATION.EQU",296,"92-04-23 16:54"

MEMory:CATalog:SEQ?

The MEMory:CATalog:SEQ? query returns file information on all sequence files in the internal memory of the waveform generator.

Group MEMORY

Related Commands MEMory:CATalog:ALL?, MEMory?

Syntax MEMory:CATalog:SEQ?

Arguments None

Returns :MEMORY:CATALOG:SEQ<File Entry>[,<File Entry>]...
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>,
<File Name>::=<string>,
<File Size>::=<NR1>, and
<Time Stamp>::=<string>

Examples :MEMORY:CATALOG:SEQ?
might return :MEMORY:CATALOG:SEQ "SEQUENCE.SEQ",960,"92-04-23 16:48"

MEMory:CATalog:WFM?

The MEMory:CATalog:WFM? query returns file-specific information about all waveform files in the internal memory of the waveform generator.

Group MEMORY

Related Commands MEMory:CATalog:ALL?, MEMory?

Syntax MEMory:CATalog:WFM?

Arguments None

Returns :MEMORY:CATALOG:WFM<File Entry>[,<File Entry>]...
where
<File Entry>::=<File Name>, <File Size>, <Time Stamp>,
<File Name>::=<string>,
<File Size>::=<NR1>, and
<Time Stamp>::=<string>

Examples :MEMORY:CATALOG:WFM?
might return the following response:
:MEMORY:CATALOG:WFM "WAVE2.WFM",2948,"92-04-23 16:47","WAVE-FORM.WFM", 2948,"92-04-23 16:47"

MEMory:COMMent (?)

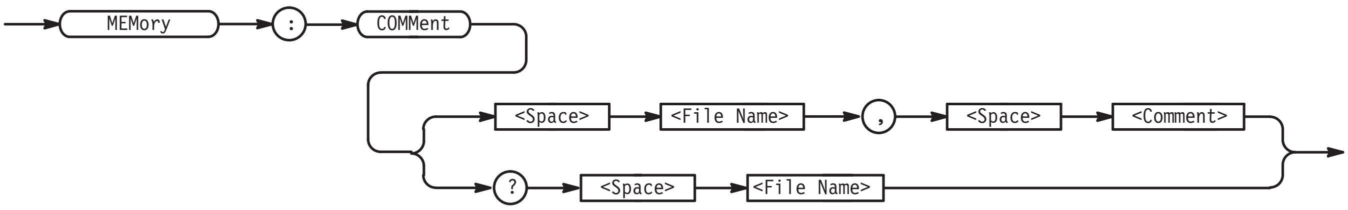
The MEMory:COMMent command writes a comment into the comment column of the specified file in the internal memory of the waveform generator. The MEMory:COMMent? query returns comments in the comment column of the specified file. A comment cannot be written to a file that is locked using the MEMory:LOCk command.

Group MEMORY

Related Commands MEMory:COPY, MEMory:DELete, MEMory:REName, MEMory:LOCk

Syntax MEMory:COMMent <File Name>, <Comment>

MEMory:COMMent? <File Name>



Arguments <File Name>::=<string>
where <string> is the name of the file to which to write the comment.

<Comment>::=<string>
where <string> is a comment of up to 24 characters.

Examples :MEMORY:COMMENT "TDS_REF.WFM", "COPIED FROM TDS REF."
writes the comment into the file TDS_REF.WFM.

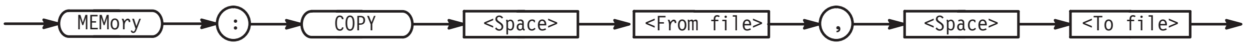
MEMory:COPY

The `MEMory:COPY` command copies a file in internal memory. If the destination file <To file> does not exist, a file will be created. If the destination file already exists, the file will be overwritten. (Files locked using the MEMory:LOCk command cannot be overwritten by `MEMory:COPY`.)

Group MEMORY

Related Commands `MEMory:DELete`, `MEMory:REName`, `MEMory:COMMent`

Syntax `MEMory:COPY <From file>, <To file>`



Arguments `<From-file>::=<string>`
where `<string>` is the source file name.

`<To-file>::=<string>`
where `<string>` is the destination file name.

Examples `:MEMORY:COPY "TDS_REF.WFM", "VXCH1.WFM"`
copies the file TDS_REF.WFM to the file VXCH1.WFM.

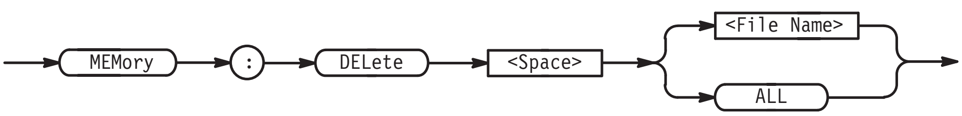
MEMory:DELete

The `MEMory:DELete` command deletes a file in the internal memory. A file locked with the `MEMory:LOCk` command cannot be deleted.

Group MEMORY

Related Commands `MEMory:COPY`, `MEMory:REName`, `MEMory:COMMent`

Syntax `MEMory:DELete { <File Name> | All }`



Arguments

<File Name>::=<string>
where <string> is either the name of the file to be deleted or ALL when every file in internal memory is to be deleted.

Examples

:MEMORY:DELETE "TEST1.WFM"
deletes the file TEST1.WFM from internal memory.

MEMory:FREE?

The MEMory:FREE? query returns used size and unused size of the internal memory. This query is equivalent to the MEMory:FREE:ALL? query.

Group

MEMORY

Related Commands

MEMory:FREE:ALL?, MEMory?

Syntax

MEMory:FREE?

Arguments

None

Returns

:MEMORY:FREE:ALL<Unused Size>, <Used Size>
where
<Unused Size>::=<NR1> and
<Used Size>::=<NR1>

Examples

:MEMORY:FREE?
might return :MMEMORY:FREE:ALL 1696220,28500

MEMory:FREE:ALL?

The MEMory:FREE:ALL? query returns used size and unused size of the internal memory. This query is equivalent to the MEMory:FREE? query.

Group MEMORY

Related Commands MEMory:FREE?, MEMory?

Syntax MEMory:FREE:ALL?

Arguments None

Returns :MEMORY:FREE:ALL<Unused Size>, <Used Size>
where
<Unused Size>::=<NR1> and
<Used Size>::=<NR1>

Examples :MEMORY:FREE:ALL?
might return :MEMORY:FREE:ALL 1696220,28500.

MEMory:LOCk (?)

The MEMory:LOCk command locks or unlocks a file in the internal memory. The MEMory:LOCk? query returns status indicating whether a file is locked or not locked. The following operations cannot be performed on a locked file:

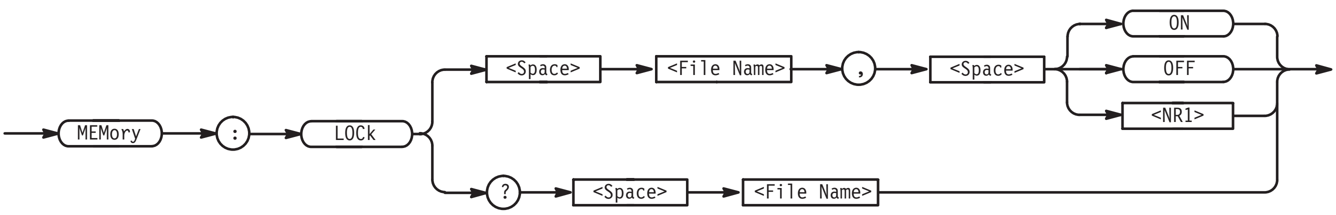
- File deletion using MEMory:DELete
- File overwriting using MEMory:COPY or load operations
- Commenting of files using MEMory:COMMent
- File renaming using MEMory:REName

Group MEMORY

Related Commands MEMory:DELete, MEMory:COPY, MEMory:REName, MEMory:COMMent

Syntax MEMory:LOCk <File Name>, { ON | OFF | <NR1> }

MEMory:LOCk? <File Name>



Arguments <File Name>::=<string>
where
<string> is the name of the file to be locked or unlocked,
ON or a non-zero value (locks the file), and
OFF or zero value (unlocks the file)

Examples :MEMORY:LOCK "RAMP_W1.WFM", 1
locks the file RAMP_W1.WFM.

MEMory:REName

The MEMory:REName command changes the name of a file in the internal memory. A file that is locked using the MEMory:LOCk command cannot be renamed.

Group MEMORY

Related Commands MEMory:DELete, MEMory:LOCk

Syntax MEMory:REName <From filename>, <To filename>

Arguments <From-filename>::=<string>
where <string> is the name of the file *to be* changed.
<To-filename>::=<string>
where <string> is the name of the file *after it is* changed. The file extensions of both files must be same. Specifying a different file extension for the files causes an error.

Examples :MMEMORY:RENAME "TDS_REF.WFM", "VXCH1.WFM"
renames the file TDS_REF.WMF to VXCH1.WFM.

MODE (?)

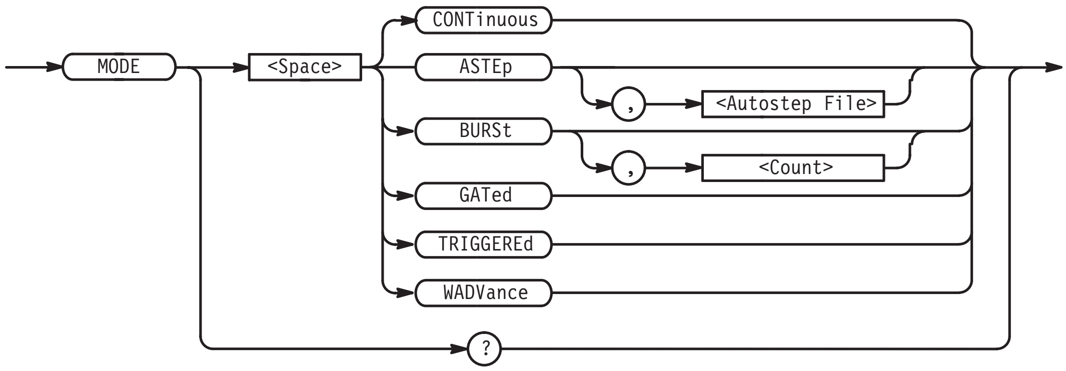
The MODE command selects the mode used to output a waveform or sequence. The MODE? query returns the current selected mode.

Group MODE

Related Commands None

Syntax

```
MODE { CONTinuous | ASTEp [,<Autostep File>]| BURSt[,<Count>] |
GATed | TRIGGEREd | WADVance }

MODE?
```



Arguments

Argument	Description
CONTinuous	Selects the continuous mode which continuously outputs waveform or sequence.
ASTEp	Selects the auto step mode which outputs one cycle of a waveform or step of a sequence per trigger. For example, this mode advances one step per trigger of a sequence stored in an autostep file. `<Autostep File>::=<string>`
BURSt	Selects the burst mode which outputs <Count> waveform cycles or sequence steps for each trigger. `<Count>::=<NR1> burst count (range: 1 to 65535)`
GATed	Selects the gated mode which continuously outputs waveforms or sequences as long as the trigger remains enabled. The trigger remains effective as long as any of the following events occur: ■ A valid external gate signal remains input ■ A STARt/*TRG command has been executed but a STOP command has not yet been issued
TRIGGEREd	Selects the triggered mode, which outputs one waveform cycle or sequence step for each trigger.
WADVance	Selects the waveform advance mode which continuously outputs one step of a sequence, as when advancing one step for each trigger. The repetition criteria defined in the sequence editor is neglected in this case. Output continues until next trigger event is generated.

Examples

```
:MODE BURST, 200
```

sets output for burst mode with 200 waveform cycles.

OFFSet (?)

The OFFSet command sets the offset voltage of waveforms output from the specified channel. The OFFSet? query returns the offset voltage currently set.

Group

SETUP

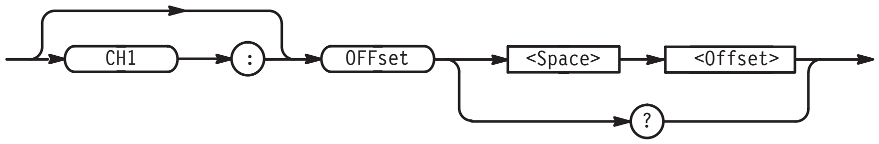
Related Commands

AMPLitude, FILTer, OPERation, WAVeform

Syntax

```
[CH1:]OFFSet <Offset>
```

```
[CH1:]OFFSet?
```



Arguments

<Offset>::=<NR2>[<unit>]
where <NR2> has a range of –2.500 V to 2.500 V in steps of 0.005 V and <unit>::={ V | mV }.

Examples

OFFSET 50.0mV
sets the offset voltage of channel 1 to 50 mV.

*OPC (?)

The *OPC common command generates the operation complete message by setting bit 0 in the SESR (Standard Event Status Register), when all pending operations are finished.

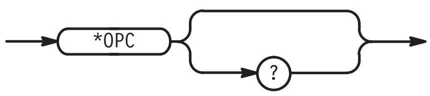
The *OPC? query returns a "1" ASCII character when all pending operations are finished.

Group SYNCHRONIZATION

Related Commands *WAI

Syntax *OPC

*OPC?



Arguments None

Examples :DESE 1

*ESE 1

*SRE 0 (serial poll used)	or	SRE 32 (service request used)
*OPC	or	*OPC? (start operations)

The above sequence sets up the system to wait until all pending operations are finished, which occurs when the ASCII character 1 is returned to the controller. Note that different *SRE commands are used depending on whether serial poll or service request handler is used for synchronization.

OPERation (?)

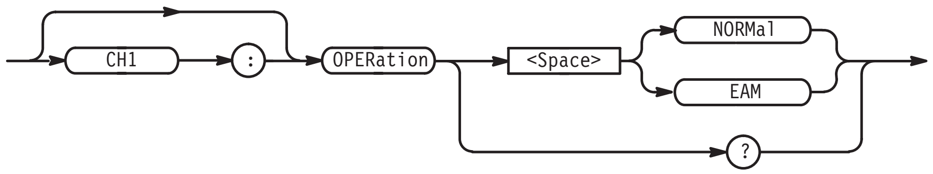
The OPERation command selects an operator that mathematically modifies the waveform. The OPERation? query returns the currently selected operation.

Group SETUP

Related Commands AMPLitude, FILTer, OFFSet, WAVeform

Syntax [CH1:]OPERation { NORMal | EAM }

[CH1:]OPERation?



Arguments The choices are described below.

Argument	Description
NORMal	Applies no operation to the waveform
EAM	Multiplies the waveform by the external signal applied through the AM INPUT BNC connector

Examples OPERATION EAM
selects External AM mode.

*OPT?

The *OPT? common query returns the implemented options of the waveform generator.

Group SYSTEM

Related Commands None

Syntax *OPT?

Arguments None

Returns <Option>[,<Option>]...
where
0 indicates no option

Examples *OPT?
will only return 0 because no installed options are available for the instrument.

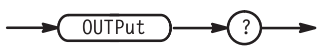
OUTPut?

The OUTPut? query returns all settings which can be set with the OUTPUT commands.

Group OUTPUT

Related Commands All output commands

Syntax OUTPut?



Arguments None

Returns Returns the settings as a sequence of commands, suitable for sending as set commands later to restore a setup (see *Examples* below)

Examples :OUTPUT?
might return :OUTPUT:STATE 0:OUTPUT:SYNC END

OUTPut:STATe (?)

The `OUTPut:STATe` command turns waveform output on or off for the specified channel. The `OUTPut:STATe?` query returns status indicating whether the output is turned on or not.

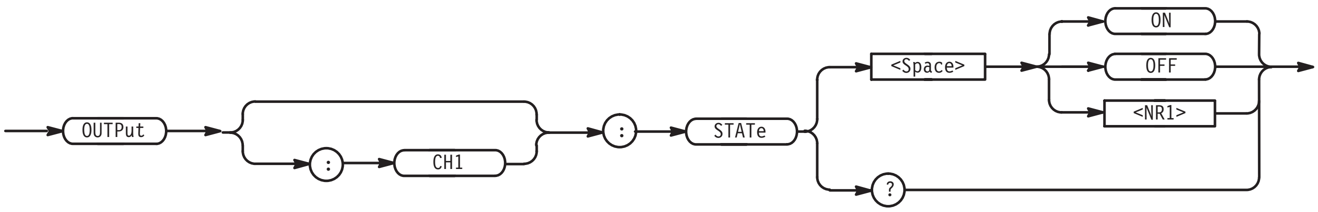
Group OUTPUT

Related Commands None

Syntax

```
OUTPut:[CH1:]STATe { ON | OFF | <NR1> }
```

```
OUTPut:[CH1:]STATe?
```



Arguments

`ON` or any non-zero value for <NR1> turns the output on
`OFF` or any zero value turns output off

Returns

1	the output is currently turned on
0	the output is currently turned off

Examples

`:OUTPUT:STATE 1`
turns on the output.

OUTPut:SYNC (?)

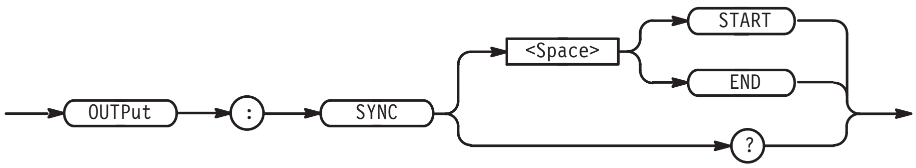
The `OUTPut:SYNC` command selects a point on the waveform at which the sync signal is generated and output at the SYNC connector on the front panel. The `OUTPut:SYNC?` query returns the currently selected position.

Group OUTPUT

Related Commands None

Syntax `OUTPut:SYNC { START | END }`

`OUTPut:SYNC?`



Arguments `START` — generates a sync signal when a waveform is triggered

`END` — generates a sync signal at the end of a waveform

Examples `:OUTPUT:SYNC END`
sets the sync signal to output at the end of a waveform.

*RST

The `*RST` common command resets the waveform generator to the default state (default values are listed in *Appendix C*).

Group SYSTEM

Related Commands None

Syntax `*RST`

Arguments None

Examples `*RST`
resets the instrument.

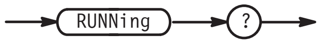
RUNNing (?)

The `RUNNing?` query returns status that indicates whether a waveform is being output or not.

Group MODE

Related Commands `STARt`, `STOP`

Syntax `RUNNing?`



Arguments None

Returns

`1` a waveform or a sequence is being output
`0` nothing is being output

Examples

`:RUNNING?`
might return `:RUNNING 1`.

SELFcal?

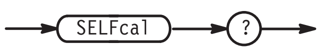
The `SELFcal?` query runs the selected calibration routine(s) and returns the results of the routine's execution.

***NOTE**. The waveform generator does not respond to any commands or queries issued during the self-cal. The self-cal takes up to 30 seconds to complete.*

Group CALIBRATION and DIAGNOSTIC

Related Commands `SELFcal:SELect`, `SELFcal:STATe`, `SELFcal:RESUlt?`

Syntax `SELFcal?`



Arguments None

Returns :SELFCAL:SELECT<Calibration Routine>;RESULT <Result>[,<Result>]...
where <Calibration Routine>::= one of following arguments:

ALL	is all routines below
CLOCk	is the clock unit calibration routine
SETUp	is the setup-related unit calibration routine

and where <Result>::=<NR1> is one of following responses:

0	terminated without error
200	detected errors in the clock unit
600	detected errors in the setup-related unit

Examples :SELFCAL?
might return :SELFCAL:SELECT ALL;RESULT 0

SELFcal:RESULt?

The SELFcal:RESUlt? query returns results of calibration execution.

Group CALIBRATION and DIAGNOSTIC

Related Commands SELFcal:SELect, SELFcal:STATe

Syntax SELFcal:RESUlt?

Arguments None

Returns :SELFCAL:RESULT<Result>[,<Result>]...
where <Result>::=<NR1> is one of following values:

0	terminated without error
200	detected errors in the clock unit
600	detected errors in the setup-related unit

Examples :SELFCAL:RESULT?
queries the result of executing a calibration.

SELFcal:SELect (?)

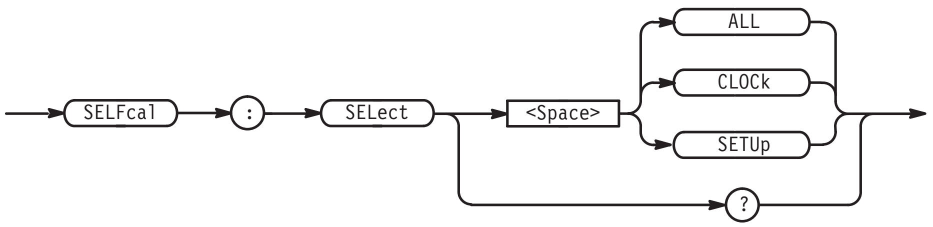
The `SELFcal:SELect` command selects the calibration routine(s). The `SELFcal:SELect?` query returns the currently selected routine.

Group CALIBRATION and DIAGNOSTIC

Related Commands `SELFcal:STATe`, `SELFcal:RESUlt?`

Syntax `SELFcal:SELect { ALL | CLOCk | SETUp }`

`SELFcal:SELect?`



Arguments

`ALL` calibrates all (both units listed below)
`CLOCk` calibrates the clock unit
`SETUp` calibrates the unit related to instrument setup

Examples

`:SELFCAL:SELECT CLOCK;STATE EXECUTE`
selects the clock for calibration and then calibrates the clock.

SELFcal:STATe

The `SELFcal:STATe` command executes the calibration routine(s) selected with the `SELFcal:SELect` command. If an error is detected during execution, the routine that detected the error stops immediately. If ALL (for all routines) is selected with the `SELFcal:SELect` command, self-calibration continues at the next routine.

Group CALIBRATION and DIAGNOSTIC

Related Commands SELFcal:SELect, SELFcal:RESult?

Syntax `SELFcal:STATe EXECute`

Arguments `EXECute` — performs calibration on selected routine

Examples `:SELFCAL:SELECT ALL;STATE EXECUTE; RESULT?`
executes all calibration routines. After calibration is finished, the results are returned.

SEQUence:DEFine (?)

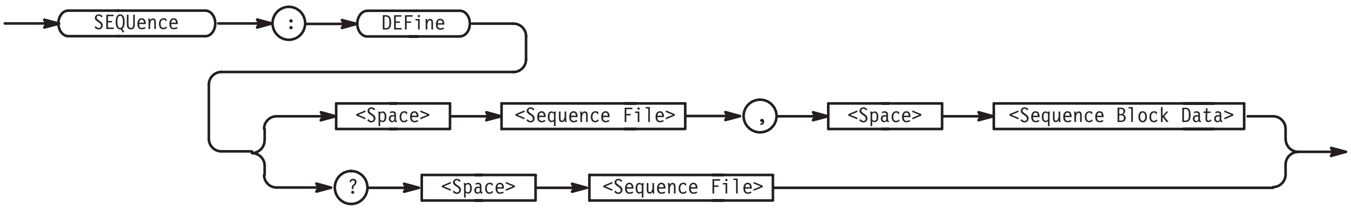
The `SEQUence:DEFine` command writes sequence data to the specified file. The `SEQUence:DEFine?` query returns sequence data that is written in the specified file.

Group WAVEFORM

Related Commands None

Syntax

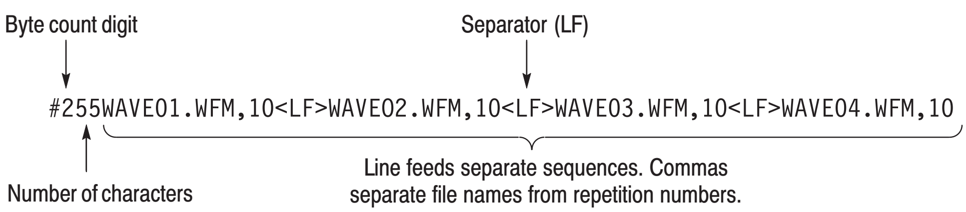
```
SEQUence:DEFine <Sequence File>, <Sequence Block Data>

SEQUence:DEFine? <Sequence File>
```



Arguments

```
<Sequence File>::=<string>
<Sequence Block Data>::=<Arbitrary Block>
```

where `<Sequence Block Data>` must be written in ASCII code and each sequence is separated by Line Feed (LF) code. The file name and repetition number are separated by a comma.



Examples

```
:SEQUENCE:DEFINE "SQWAVE.SEQ",
#255WAVE01.WFM,10<LF>WAVE02.WFM,10<LF>WAVE0
3.WFM,10<LF>WAVE04.WFM,10
```

writes sequence data to the file SQWAVE.SEQ.

SEQUence:EXPand

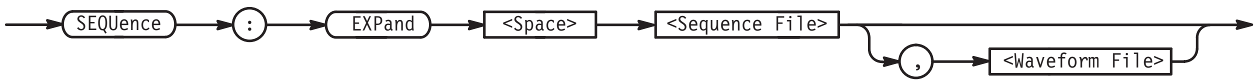
The `SEQUence:EXPand` command generates a waveform by accessing a sequence that was written to a specified sequence file, and transferring the data to a waveform file.

Group WAVEFORM

Related Commands SEQUence:DEFine

Syntax

```
SEQUence:EXPand <Sequence File>[,<Waveform File>]
```



Arguments

```
<Sequence File>::=<string> Sequence File Name
<Waveform File>::=<string> Waveform File Name
```

The sequence file and waveform file must be in the internal memory. If you do not specify a `<Waveform File>`, a waveform file with the sequence file name and the ".WFM" suffix, is generated. If the new waveform file has the same name as an existing file, an error will be detected. After developing a waveform file from a sequence, the number of waveform points contained within the new waveform file is equal to the sum of each unique waveform within the sequence, multiplied by the number of repetitions for each unique waveform.

Examples

```
:SEQUence:EXPand "SQWAVE.SEQ"
```

develops the sequence file "SQWAVE.SEQ" into a waveform file. In this example, the resulting waveform file name will default to "SQWAVE.WFM."

```
:SEQUence:EXPand "SQWAVE.SEQ","SWAVE01.WFM"
```

develops the sequence file "SQWAVE.SEQ" into the user-named waveform file "SWAVE01.WFM."

*SRE (?)

The *SRE common command sets the bits of the SRER (Service Request Enable Register). The *SRE? common query returns the contents of SRER.

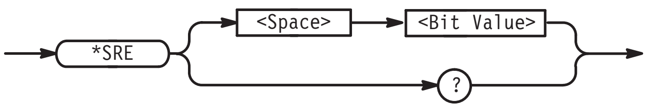
The power-on default for the SRER is all bits reset if the power-on status flag is TRUE. If this flag is set to FALSE, the SRER maintains its value through a power cycle. See *Status and Events* for more details on the SRER.

Group STATUS and EVENT

Related Commands *CLS, DESE, *ESE, *ESR?, EVENT?, EVMsg?, EVQty?, *STB?

Syntax *SRE <Bit Value>

*SRE?



Arguments <Bit Value>::=<NR1>
where the argument must be decimal number from 0 to 255. The SRER bits are set in binary bit according to the decimal number.

Examples *SRE 48
sets the SRER to 48 (binary 00110000), which sets the ESB and MAV bits.

*SRE?
might return 32 which indicates that the SRER contains the binary number 00100000.

STARt

The `STARt` command generates a trigger event to start the output of a waveform or a sequence.

Group MODE

Related Commands `RUNNing?`, `STOP`, `*TRG`

Syntax `STARt`

Arguments None

Examples `:START`
generates a trigger event.

*STB?

The `*STB?` common query returns the value of the SBR (Status Byte Register). At this time, bit 6 of the SBR is read as a MSS (Master Status Summary) bit. Refer to *Status and Events* for more details on the SBR.

Group STATUS and EVENT

Related Commands `*CLS`, `DESE`, `*ESE`, `*ESR`, `EVENT?`, `EVMsg?`, `EVQty?`, `*SRE`

Syntax `*STB?`

Arguments None

Returns `<NR1>`
which is a decimal number.

Examples `*STB?`
might return `96`, which indicates that the SBR contains the binary number 01100000.

STOP

The STOP command terminates waveform output. When the mode is *not* set to continuous, it also resets the sequence pointer to output the waveform from the top of the sequence with next trigger event.

Group MODE

Related Commands RUNNing?, STARt, *TRG

Syntax STOP

Arguments None

Examples :STOP
stops the output of a waveform.

*TRG

The *TRG common command generates a trigger event. This command is equivalent to the STARt command.

Group MODE

Related Commands RUNNing?, STARt, STOP

Syntax *TRG

Arguments None

Examples *TRG
generates a trigger event.

TRIGger?

The TRIGger? query returns all of the currently specified settings related to the trigger function.

Group MODE

Related Commands RUNNing?, STARt, STOP

Syntax TRIGger?

Arguments None

Examples :TRIGGER?
might return :TRIGGER:IMPEDANCE HIGH;LEVEL 1.400;
POLARITY POSITIVE;SLOPE POSITIVE;INPUT INTERNAL;OUTPUT OFF

TRIGger:IMPedance (?)

The TRIGger:IMPedance command selects high impedance (1 MΩ) or low impedance (50 Ω) for the external trigger input connector.

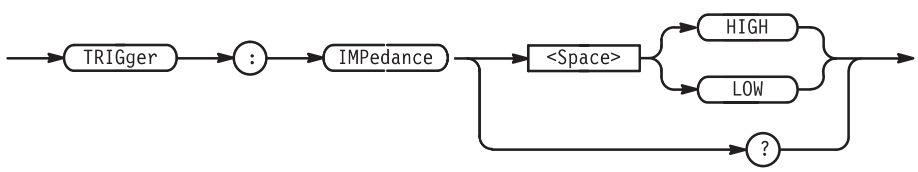
The TRIGger:IMPedance? query returns currently selected impedance.

Group MODE

Related Commands TRIGger:LEVel, TRIGger:POLarity, TRIGger:SLOPe

Syntax TRIGger:IMPedance { HIGH | LOW }

TRIGger:IMPedance?



Arguments

`HIGH` — selects high impedance: 1 MΩ

`LOW` — selects low impedance: 50 Ω

Examples

`:TRIGGER:IMPEDANCE LOW`
selects low impedance.

TRIGger:INPut (?)

The `TRIGger:INPut` command selects the trigger source. The `TRIGger:INPut?` query returns the currently selected trigger source.

Group

MODE

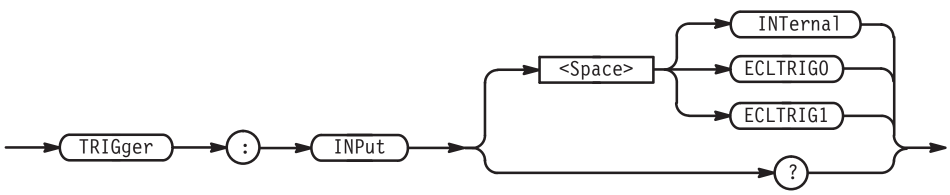
Related Commands

`TRIGger:OUTPut`, `CLOCk:SOURce`

Syntax

`TRIGger:INPut { INTernal | ECLTRIG0 | ECLTRIG1 }`

`TRIGger:INPut?`



Arguments

`INTernal` — selects the internal trigger signal as the trigger

`ECLTRIG0` — selects the VXIbus signal ECLTRIG0 as the trigger

`ECLTRIG1` — selects the VXIbus signal ECLTRIG1 as the trigger

Examples

`:TRIGGER:INPUT INTERNAL`
selects the internal trigger signal as the trigger. In this example, the module becomes the trigger master for synchronous operation.

TRIGger:LEVel (?)

The `TRIGger:LEVel` command sets the level on the external trigger at which the trigger event is generated. The `TRIGger:LEVel?` query returns the level currently set.

Group

MODE

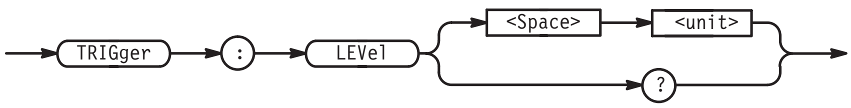
Related Commands

`TRIGger:IMPedance`, `TRIGger:POLarity`, `TRIGger:SLOPe`

Syntax

```
TRIGger:LEVel <unit>
```

```
TRIGger:LEVel?
```



Arguments

`<Level>::=<NR2>[<unit>]`
where `<unit>::={V | mV}` with a range of –5.0 V to 5.0 V, in 0.1 V steps.

Examples

`:TRIGGER:LEVEL 200mV`
sets the level to 200 mV.

TRIGger:OUTPut (?)

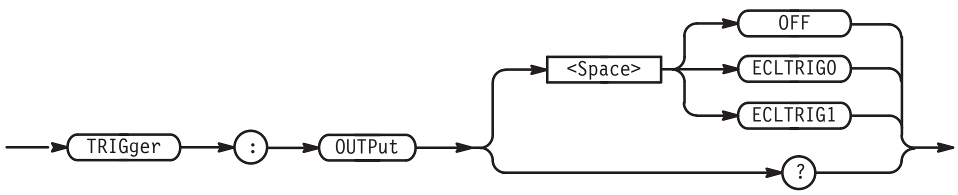
The TRIGger:OUTPut command determines whether or not the waveform generator outputs a trigger signal onto the VXIbus. The TRIGger:OUTPut? query returns the current setting of the trigger output.

Group MODE

Related Commands TRIGger:INPut, CLOCk:SOURce

Syntax TRIGger:OUTPut { OFF | ECLTRIG0 | ECLTRIG1 }

TRIGger:OUTPut?



Arguments OFF — no trigger is output to VXIbus

ECLTRIG0 — outputs a trigger on the ECLTRIG0 line of the VXIbus

ECLTRIG1 — outputs a trigger on the ECLTRIG1 line of the VXIbus

Examples :TRIGGER:OUTPUT OFF
sets the trigger output off so that a trigger signal is not sent to the VXIbus. In this example, the module becomes a trigger slave for synchronous operation.

TRIGger:POLarity (?)

The TRIGger:POLarity command selects polarity of the external trigger signal which generates the trigger event. The TRIGger:POLarity? query returns the currently selected polarity.

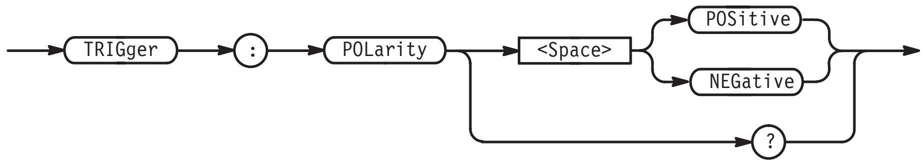
The polarity parameter is valid only when the mode is set to gated mode.

Group MODE

Related Commands TRIGger:IMPedance, TRIGger:LEVel, TRIGger:SLOPe

Syntax TRIGger:POLarity { POSitive | NEGative }

TRIGger:POLarity?



Arguments POSitive — selects positive polarity

NEGative — selects negative polarity

Examples :TRIGGER:POLARITY NEGATIVE
selects negative polarity.

TRIGger:SLOPe (?)

The TRIGger:SLOPe command selects the rising or falling edge of the external signal which generates the trigger event. The TRIGger:SLOPe? query returns status indicating which slope is currently selected.

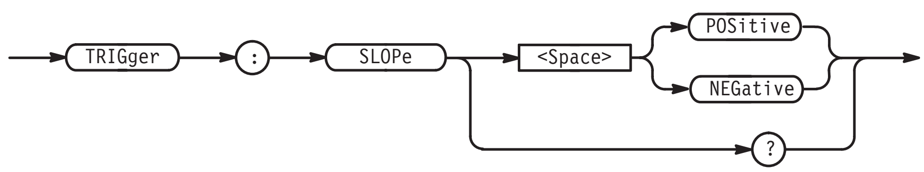
The slope parameter is valid only when modes other than gated or continuous are selected.

Group MODE

Related Commands TRIGger:IMPedance, TRIGger:LEVel, TRIGger:POLarity

Syntax TRIGger:SLOPe { POSitive | NEGative }

TRIGger:SLOPe?



Arguments POSitive — selects rising edge

NEGative — selects falling edge

Examples :TRIGGER:SLOPE POSITIVE
selects rising edge for trigger.

*TST?

The *TST? common query performs the self-test and returns the results. If an error is detected during self-test, execution is immediately stopped.

***NOTE**. The waveform generator does not respond to any commands or queries issued during the self-test. The self-test takes up to 90 seconds to complete.*

Group CALIBRATION and DIAGNOSTIC

Related Commands DIAG:SELect, DIAG:STATe, DIAG:RESUlt?

Syntax *TST?

Arguments None

Returns <Result>
where <Result>::=<NR1> and <NR1> is one of following arguments.

0	terminated without error
100	detected an error in the CPU unit
600	detected an error in the setup-related unit
700	detected an error in the waveform memory

Examples *TST?
might return 100 to indicate that errors were detected in the CPU unit.

VERBose (?)

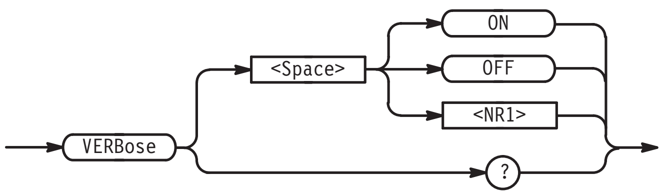
The VERBose command selects the long headers or the short headers to be returned with response messages. Longer response headers enhance readability for other programmers; shorter response headers provide faster bus transfer speed.

Group SYSTEM

Related Commands HEADer

Syntax VERBose { ON | OFF | <NR1> }

VERBose?



Arguments ON or non-zero value — selects long response header

OFF or zero value — selects short response header

Returns Responses are decimal numbers (<NR1>) and are defined as follows:

1	long header is currently selected
0	short header is currently selected

Examples :VERBOSE ON
sets long header for query responses.

:VERBOSE?
might return :VERBOSE 1, which indicates that the long response header is currently selected.

*WAI

The *WAI common command prevents the waveform generator from executing any further commands or queries until all pending operations are completed.

Group SYNCHRONIZATION

Related Commands *OPC

Syntax *WAI

Arguments None

Returns None

Examples *WAI
prevents the execution of any commands or queries until all pending operations complete.

WAVeform (?)

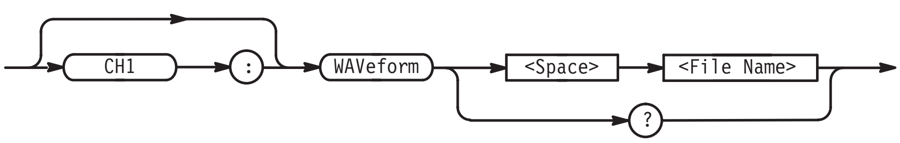
The WAVeform command selects a waveform or a sequence for output. The WAVeform? query returns the currently specified waveform or sequence file.

Group SETUP

Related Commands AMPLitude, FILTer, OFFSet, OPERation

Syntax [CH1:]WAVeform <File Name>

[CH1:]WAVeform?



Arguments

<File Name>::=<string>
where <string> is a waveform file name or sequence file name.

Examples

WAVEFORM "SOUARE.WFM"
selects the waveform in the waveform file SOUARE.WFM as the waveform output.

WAVFrm?

The WAVFrm? query transmits waveform preamble and the waveform data. This query is equivalent to the WFMPre? query, followed by the CURVe? query.

Group

WAVEFORM

Related Commands

CURVe?, DATA:SOURce, DATA:ENCDG, WFMPre?

Syntax

WAVFrm?

Arguments

None

Returns

Returns the settings as a sequence of commands, suitable for sending as set commands to restore a setup (see *Examples* below)

Examples

:WAVFRM?
might return the following response:
:WFMPRE:ENCDG BIN;BN_FMT RP;BYT_NR 2;BIT_NR 12;BYT_OR MSB;CRVCHK NONE;WFID "WAVEFORM.WFM, 1000 points, clock: 100.0MHz, amplitude: 1.000V, offset: 0.000V";NR_PT 1000;PT_FMT Y;XUNIT "S"; XINCR 1.0000E-08;PT_OFF 0;XZERO 0.000;YUNIT "V";YMULT 2.442E-04; YZERO 0.000;YOFF 2.047E+03;:CURVE #42000<DAB><DAB> ... <DAB>

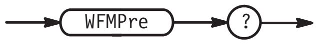
WFMPre?

The `WFMPre?` query returns all settings for the waveform preamble.

Group WAVEFORM

Related Commands All WFMPRE sub-group commands, `DATA:SOURce`

Syntax `WFMPre?`



Arguments None

Returns Returns the settings as a sequence of commands, suitable for sending as set commands to restore a setup (see *Examples* below)

Examples `:WFMPRE?` might return as follows:

```
:WFMPRE:ENCDG BIN;BN_FMT RP;BYT_NR 2;BIT_NR 12;BYT_OR MSB;CRVCHK
NONE;WFID "WAVEFORM.WFM, 1000 points, clock: 100.0MHz, amplitude:
1.000V, offset: 0.000V";NR_PT 1000;PT_FMT Y;XUNIT "S"; XINCR
1.0000E-08;PT_OFF 0;XZERO 0.000;YUNIT "V";YMULT 2.442E-04; YZERO
0.000;YOFF 2.047E+03
```

WFMPre:BIT_NR (?)

The `WFMPre:BIT_NR` command specifies the number of bits of precision for each binary data point. The `WFMPre:BIT_NR?` query returns the bits of precision currently specified.

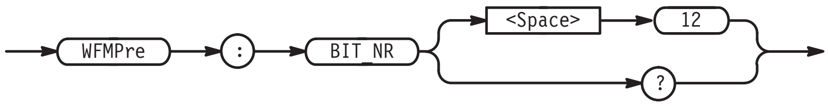
Group WAVEFORM

Related Commands `WFMPre:BN_FMT`, `WFMPre:BYT_NR`, `WFMPre:PT_FMT`, `WFMPre:BYT_OR`, `WFMPre:ENCDG`, `DATA:ENCDG`

Syntax

WFMPre:BIT_NR 12

WFMPre:BIT_NR?



Arguments

12
Any argument other than 12 (the default) is ignored.

Examples

:WFMPRE:BIT_NR?
might return :WFMPRE:BIT_NR 12

WFMPre:BN_FMT (?)

The WFMPre:BN_FMT command specifies format of binary data. The WFMPre:BN_FMT? query returns the binary data format currently specified.

Group

WAVEFORM

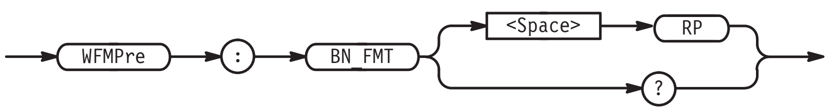
Related Commands

WFMPre:BYT_NR, WFMPre:BIT_NR, WFMPre:BYT_OR, WFMPre:ENCDG, DATA:ENCDG

Syntax

WFMPre:BN_FMT RP

WFMPre:BN_FMT?



Arguments

RP — binary unsigned integer code

Any argument other than RP (the default) is ignored.

Examples

:WFMPRE:BN_FMT?
might return :WFMPRE:BN_FMT RP

WFMPre:BYT_NR (?)

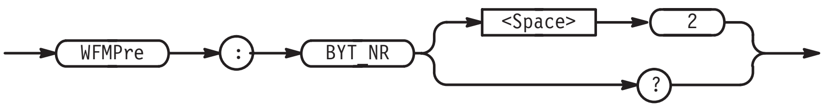
The WFMPre:BYT_NR command specifies data field width (byte length) for each binary data point. The WFMPre:BYT_NR? query returns the data field width currently specified.

Group WAVEFORM

Related Commands WFMPre:BN_FMT, WFMPre:BIT_NR, WFMPre:BYT_OR, WFMPre:ENCDG, DATA:ENCDG

Syntax WFMPre:BYT_NR 2

WFMPre:BYT_NR?



Arguments 2

Any argument other than 2 (the default) is ignored.

Examples WFMPRE:BYT_NR
might return :WFMPRE:BYT_NR 2.

WFMPre:BYT_OR (?)

The WFMPre:BYT_OR command specifies which byte of the binary data is sent first when the data field width of the binary data is defined to be two bytes. The WFMPre:BYT_OR? query returns the binary data byte order currently specified.

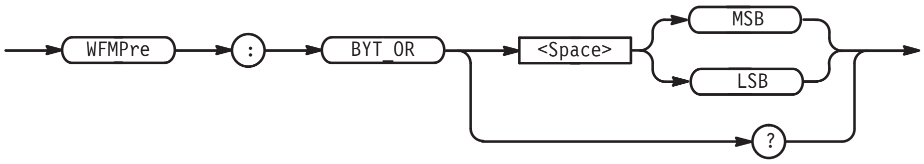
Group WAVEFORM

Related Commands WFMPre:BN_FMT, WFMPre:BYT_NR, WFMPre:BIT_NR, WFMPre:ENCDG, DATA:ENCDG

Syntax

WFMPre:BYT_OR { MSB | LSB }

WFMPre:BYT_OR?



Arguments

MSB — sends upper byte first, then lower byte for each data word

LSB — sends lower byte first, then upper byte for each data word

Examples

:WFMPRE:BYT_OR?
might return :WFMPRE:BYT_OR MSB.

WFMPre:CRVCHK (?)

The WFMPre:CRVCHK command specifies the error check method for binary data. The WFMPre:CRVCHK? query returns the error check method currently specified.

Group

WAVEFORM

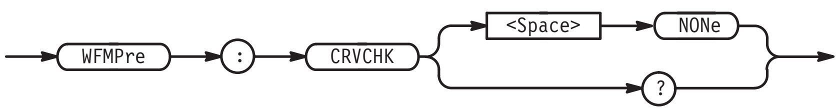
Related Commands

WFRPre:ENCDG, DATA:ENCDG

Syntax

WFMPre:CRVCHK NONe

WFMPre:CRVCHK?



Arguments

NONe — no error checking; all binary block data represent data

Any argument other than NONe (the default) is ignored.

Examples

:WFMPRE:CRVCHK?
might return :WFMPRE:CRVCHK NONE.

WFMPre:ENCDG (?)

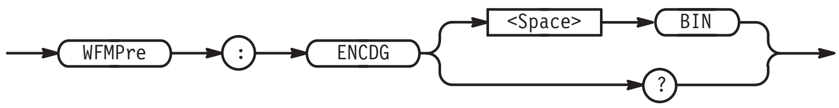
The WFMPre:ENCDG command sets the encoding type for the waveform transmitted with the CURVe command. The WFMPre:ENCDG? query returns the encoding type currently set.

Group WAVEFORM

Related Commands DATA:ENCDG

Syntax WFMPre:ENCDG BIN

WFMPre:ENCDG?



Arguments BIN — specifies binary encoding type

Any argument other than BIN (the default) is ignored.

Examples :WFMPre:ENCDG?
might return :WFMPRE:ENCDG BIN.

WFMPre:NR_PT (?)

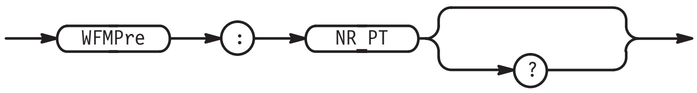
The WFMPre:NR_PT command sets the size of the waveform in terms of sets of points. The waveform generator sets the size of the waveform automatically. This parameter cannot be set by the user. The WFMPre:NR_PT? query returns the waveform size currently set.

Group WAVEFORM

Related Commands DATA:SOURce, DATA:DESTination

Syntax
WFMPre:NR_PT

WFMPre:NR_PT?



Arguments
None; the waveform generator sets the size of the waveform automatically

Examples
WFMPre:NR_PT?
might return :WFMPRE:NR_PT 131072.

WFMPre:PT_FMT (?)

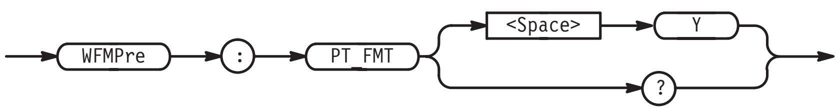
The WFMPre:PT_FMT command selects the data point format of the waveform. The WFMPre:PT_FMT? query returns the data point format currently selected.

Group
WAVEFORM

Related Commands
WFMPre:PT_OFF, WFMPre:XINCR, WFMPre:XMULT, WFMPre:XZERO, WFMPre:XOFF, WFMPre:YMULT, WFMPre:YZERO, WFMPre:YOFF

Syntax
WFMPre:PT_FMT Y

WFMPre:PT_FMT?



Arguments
Y — explicitly transmits Y values; absolute X and Y component values are calculated for each data point using the the transmission sequence yn, yn+1, yn+2 ...
where
Xn = <XZERO-value> + <XINCR-value> (n – <PT_OFF-value>)
and
Yn = <YZERO-value> + <YMULT-value> (yn – <YOFF-value>)

Any argument other than Y (the default) is ignored.

Examples
:WFMPRE:PT_FMT?
might return :WFMPRE:PT_FMT Y.

WFMPre:PT_OFF (?)

The `WFMPre:PT_OFF` command defines the X axis point offset value. The `WFMPre:PT_OFF?` query returns the X axis point offset value currently set.

Group

WAVEFORM

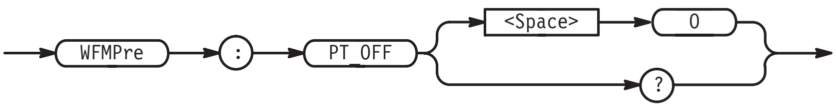
Related Commands

`WFMPre:PT_FMT`, `WFMPre:XINCR`, `WFMPre:XZERO`

Syntax

```
WFMPre:PT_OFF 0
```

```
WFMPre:PT_OFF?
```



Arguments

`0`

Any argument other than 0 (the default) is ignored.

Examples

`:WFMPRE:PT_OFF?`
might return `:WFMPRE:PT_OFF 0`.

WFMPre:WFID (?)

The `WFMPre:WFID` command sets comment and/or additional information as a waveform ID for the waveform preamble.

Group

WAVEFORM

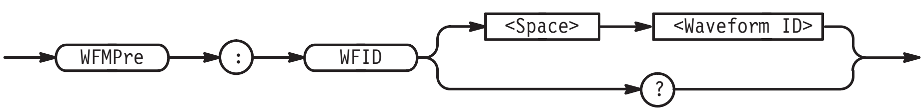
Related Commands

None

Syntax

```
WFMPre:WFID <Waveform ID>
```

```
WFMPre:WFID?
```



Arguments

<Waveform ID> is automatically set by the waveform generator, and arguments are ignored on input.

Examples

:WFMPRE:WFID?
might return the following response:
:WFMPRE:WFID "WAVEFORM.WFM, 1000 points, clock: 100.0MHz,amplitude: 1.000V, offset: 0.000V".

WFMPre:XINCR (?)

The WFMPre:XINCR command defines the X axis increment value. The WFMPre:XINCR? query returns the X axis increment value.

Group

WAVEFORM

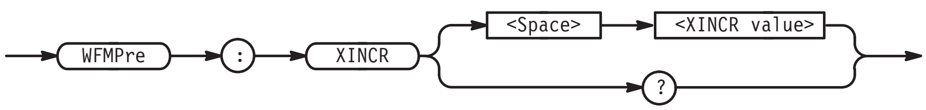
Related Commands

WFMPre:PT_FMT, WFMPre:PT_OFF, WFMPre:XZERO

Syntax

WFMPre:XINCR <XINCR value>

WFMPre:XINCR?



Arguments

<XINCR-value>::=<NR3>
where <NR3> is a decimal number that ranges from 4E–9 seconds to 1E–1 seconds.

Examples

:WFMPRE:XINCR 0.01
sets the X axis increment value to 0.01 second.

WFMPre:XUNIT (?)

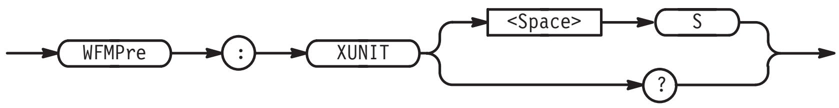
The `WFMPre:XUNIT` command defines the appropriate representation of the data unit for the X axis. The `WFMPre:XUNIT?` query returns the representation for the X axis data unit currently defined.

Group WAVEFORM

Related Commands `WFMPre:PT_OFF`, `WFMPRe:XINCR`, `WFMPre:XZERO`

Syntax `WFMPre:XUNIT S`

`WFMPre:XUNIT?`



Arguments `S`

Any argument other than S (the default) is ignored.

Examples `:WFMPRE:XUNIT?`
might return `:WFMPRE:XUNIT "S"`.

WFMPre:XZERO (?)

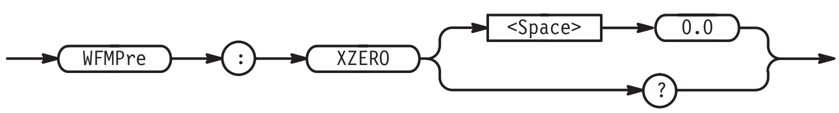
The `WFMPre:XZERO` command defines the X axis origin value.
The `WFMPre:XZERO?` query returns the X axis origin value currently defined.

Group WAVEFORM

Related Commands `WFMPre:PT_OFF`, `WFMPre:XUNIT`, `WFMPre:XINCR`

Syntax `WFMPre:XZERO 0.0`

`WFMPre:XZERO?`



Arguments 0.0

Any argument other than 0.0 (the default) is ignored.

Examples :WFMPRE:XZERO?
might return :WFMPRE:PT_OFF 0.0.

WFMPre:YMULT (?)

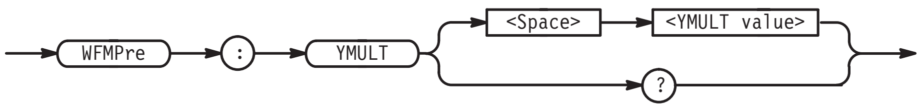
The WFMPre:YMULT command defines multiplier value of the data for the Y axis. The WFMPre:YMULT? query returns the Y axis multiplier value currently defined.

Group WAVEFORM

Related Commands WFMPre:YOFF, WFMPre:YZERO, WFMPre:YUNIT

Syntax WFMPre:YMULT <YMULT value>

WFMPre:YMULT?



Arguments <YMULT-value>::=<NR3>

Examples :WFMPRE:YMULT 0.0012
sets the multiplier value to 0.0012 V.

WFMPre:YOFF (?)

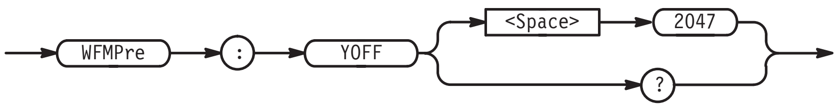
The `WFMPre:YOFF` command defines the Y axis offset value.
The `WFMPre:YOFF?` query returns the Y axis offset value currently defined.

Group WAVEFORM

Related Commands `WFMPre:YMULT`, `WFMPre:YZERO`, `WFMPre:YUNIT`

Syntax `WFMPre:YOFF 2047`

`WFMPre:YOFF?`



Arguments `2047`

Any argument other than 2047 (the default) is ignored.

Examples `:WFMPRE:YOFF?`
might return `:WFMPRE:YOFF 2.047E+03`

WFMPre:YUNIT (?)

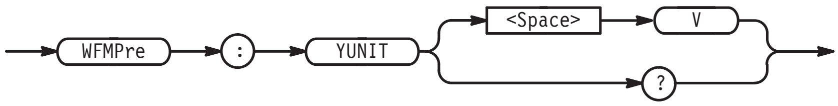
The `WFMPre:YUNIT` command defines the appropriate representation of the data unit for the Y axis. The `WFMPre:YUNIT?` query returns the representation for the Y axis data unit currently defined.

Group WAVEFORM

Related Commands `WFMPre:YMULT`, `WFMPre:YZERO`, `WFMPre:YOFF`

Syntax WFMPre:YUNIT V

WFMPre:YUNIT?



Arguments V

Any argument other than V (the default) is ignored.

Examples :WFMPRE:YUNIT?
might return :WFMPRE:YUNIT "V".

WFMPre:YZERO (?)

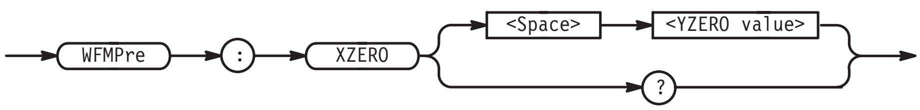
The WFMPre:YZERO command defines the Y axis origin value. The WFMPre:YZERO? query returns the Y axis origin value currently defined.

Group WAVEFORM

Related Commands WFMPre:PT_OFF, WFMPre:YMULT

Syntax WFMPre:YZERO <YZERO value>

WFMPre:YZERO?



Arguments <YZERO-value>::=<NR2>
where <NR2> is a decimal number that ranges from –2.500 to 2.500 in steps as small as 0.005. The unit volts is assumed.

Examples :WFMPRE:YZERO 0.225
sets the Y axis origin value to 0.225 V.

Status and Events

Status and Event Reporting

This section describes how the VX4792 Arbitrary Waveform Generator reports its status and internal events. The section describes the elements that comprise the status and events reporting system and explains how status and events are handled.

The status and event reporting system reports certain significant events that occur within the waveform generator. The system consists of five registers plus two queues. Four of the registers and one of the queues are compatible with IEEE Std. 488.2–1987; the other register and queue are specific to Tektronix products.

Registers

The registers fall into two functional groups: status registers and enable registers.

- Status registers store information about the status of the waveform generator. They include the Standard Event Status Register (SESR) and the Status Byte Register (SBR).
- Enable registers determine whether certain events are reported to the Status Registers and the Event Queue. They include the Device Event Status Enable Register (DESER), the Event Status Enable Register (ESER), and the Service Request Enable Register (SRER).

Status Registers

The Standard Event Status Register (SESR) and the Status Byte Register (SBR) record certain types of events that may occur while the waveform generator is in use. The IEEE Std 488.2-1987 defines these registers.

Each bit in a Status Register records a particular type of event, such as an execution error or service request. When an event of a given type occurs, the waveform generator sets the bit that represents that type of event to a value of one. (You can disable bits so that they ignore events and remain at zero. See the *Enable Registers* section on page 4–4.) Reading the status registers tells you what types of events have occurred.

Standard Event Status Register (SESR). The SESR is shown in Figure 4–1. It records eight types of events that can occur within the waveform generator. Use the *ESR? query to read the SESR register. Reading the register clears the bits of the register so that the register can accumulate information about new events.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 4–1: The Standard Event Status Register (SESR)

Table 4-1: SESR Bit Functions

Bit	Function
7 (MSB)	PON (Power On) — indicates that the waveform generator was powered on.
6	URQ (User Request) — indicates an event occurred, and because of that event, the waveform generator needs attention from the operator.
5	CME (Command Error) — indicates that an error occurred while the waveform generator was parsing a command or query. Command error messages are listed in Table B-2 on page B-2.
4	EXE (Execution Error) — indicates that an error occurred while the waveform generator was executing a command or query. An execution error occurs for either of the following reasons: ■ A value designated for the argument is not within the range allowed by the waveform generator, is not valid for the command, or is incorrect in some other way ■ Execution took place improperly under conditions that are different from those which should have been requested Execution error messages are listed in Table B-3 on page B-3.
3	DDE (Device Dependent Error) — indicates that a device-specific error occurred. Device error messages are listed in Table B-4 on page B-5.
2	QYE (Query Error) — indicates that an error occurred when attempting to read the output queue. Such an error occurs for one of the following two reasons: ■ An attempt was made to retrieve a message from the output queue even though it is empty or pending ■ Output queue message was cleared while it was being retrieved from the output queue
1	RQC (Request Control) — the waveform generator does not use this bit. Request Control (RQC) is used to show that an instrument has requested to transfer bus control back to the controller. (This is the usage prescribed by the IEEE Std. 488.1.)
0 (LSB)	OPC (Operation Complete) — indicates that the operation is complete. This bit is set when all pending operations complete following an `*OPC` command.

Status Byte Register (SBR). The SBR is shown in Figure 4–2. It records whether or not the following events have occurred:

- Output is available in the Output Queue
- The waveform generator has requested service
- The SESR has recorded any events

Use a Serial Poll or the *STB? query to read the contents of the SBR. The bits in the SBR are set and cleared depending on the contents of the SESR, the Event Status Enable Register (ESER), and the Output Queue. When you use a Serial Poll to obtain the SBR, bit 6 is the RQS bit. When you use the *STB? query to obtain the SBR, bit 6 is the MSS bit. Reading the SBR does not clear the bits, including the MSS bit.

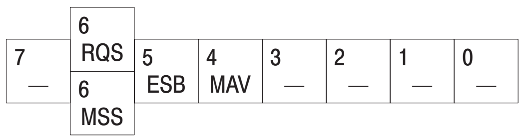


Figure 4–2: The Status Byte Register (SBR)

Table 4–2: SBR Bit Functions

Bit	Function
7 (MSB)	Not used (must be set to zero for waveform generator operation).
6	The RQS (Request Service) bit, when obtained from a serial poll, shows that the waveform generator requests service from the GPIB controller (that is, the SRQ line is asserted on the GPIB). This bit is cleared when the serial poll completes.
6	The MSS (Master Status Summary) bit, when obtained from *STB? query, summarizes the ESB and MAV bits in the SBR. (It shows that status is present and enabled in the SESR, or a message is available at the Output Queue, or both.)
5	The ESB (Event Status Bit) shows that status is enabled and present in the SESR.
4	The MAV (Message Available) bit shows that output is available in the Output Queue.
3 – 0	Not used (must be set to zero for waveform generator operation).

Enable Registers

You use the DESER (Device Event Status Enable Register), the ESER (Event Status Enable Register), and the SRER (Service Request Enable Register) to select which events are reported to the Status Registers and the Event Queue. Each of these Enable Registers acts as a filter to a Status Register (the DESER also acts as a filter to the Event Queue) and can allow or prevent information from being recorded in the register or queue.

Each bit in an Enable Register corresponds to a bit in the Status Register it controls. In order for an event to be reported to its bit in the Status Register, the corresponding bit in the Enable Register must be set to one. If the bit in the Enable Register is set to zero, the event is not recorded.

Various commands set the bits in the Enable Registers. The Enable Registers and the commands used to set them are described below.

Device Event Status Enable Register (DESER). The DESER is shown in Figure 4–3. This register controls which events are reported to the SESR and the Event Queue. The bits in the DESER correspond to those in the SESR, as was described earlier.

Use the DESE command to enable and disable the bits in the DESER. Use the DESE? query to read the DESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 4–3: The Device Event Status Enable Register (DESER)

Event Status Enable Register (ESER). The ESER is shown in Figure 4–4. It controls which events are allowed to be summarized by the Event Status Bit (ESB) in the SBR.

Use the *ESE command to set the bits in the ESER. Use the *ESE? query to read the ESER.

7	6	5	4	3	2	1	0
PON	URQ	CME	EXE	DDE	QYE	RQC	OPC

Figure 4–4: The Event Status Enable Register (ESER)

Service Request Enable Register (SRER). The SRER is shown in Figure 4–5. It controls which bits in the SBR generate a Service Request and are summarized by the Master Status Summary (MSS) bit.

Use the *SRE command to set the SRER. Use the *SRE? query to read it. The RQS bit remains set to one until either the Status Byte Register is read with a Serial Poll or the MSS bit changes back to a zero.

7	6	5	4	3	2	1	0
—	—	ESB	MAV	—	—	—	—

Figure 4–5: The Service Request Enable Register (SRER)

Queues

The status and event reporting system contains two queues: the Output Queue and the Event Queue.

Output Queue

The Output Queue is a FIFO (First In First Out) queue that holds response messages until they are requested. When a message is put in the queue, the MAV bit of the Status Byte Register (SBR) is set.

The Output Queue empties each time the waveform generator receives a new command or query. Therefore, the controller must read the output queue before it sends the next command or query command. Otherwise the controller will lose responses to earlier queries. If a command or query command is given without first reading the Output Queue, an error results (typically 410, see Table B–5 on page B–5) and the Output Queue is emptied.

Event Queue

The Event Queue is a FIFO (First In, First Out) queue which can hold up to 20 instrument-generated events. When the number of events exceeds 20, the 20th event is replaced by event code 350, "Queue overflow".

To read out the contents from the Event Queue, perform the following steps:

1. Send *ESR? — to read out the contents of SESR. When the contents of SESR are read out, SESR is cleared allowing you to take out events from the Event Queue.

2. Send one of the following queries:

- ALLEv? — to read out and return all events made available by *ESR?. The ALLEv? query returns both the event code and message text.
- EVENT? — to read out and return the oldest event of those made available by *ESR?. The EVENT? query returns only the event code.
- EVMsg? — to read out and return the oldest event of those made available by *ESR?. The EVMsg? query returns both the event code and message text.

Reading the SESR erases any events that were made available by previous *ESR? reads, unless they were not read from the Event Queue. Events that occur after an *ESR? read are put in the Event Queue but are not available until *ESR? is used again.

Processing Sequence

Figure 4–6 shows the status and event processing that is summarized below.

1. An event occurs which causes the DESR to be checked. Based on the state of the DESR, the following actions occur:

- If the control bit for that event is set in the DESER, then the SESR bit that corresponds to this event is set to 1.
- The set control bit allows the event to be placed into the Event Queue. Placing the event in the Event Queue sets the MAV bit in the SBR to 1.
- If the control bit for that event is also set in the ESER, then the ESB bit of SBR is also set to 1.

2. When either bit of SBR has been set to 1 and the corresponding control bit of SRER is also set, then the MSS bit of SBR is set and a service request is generated for use with GPIB interface operation.

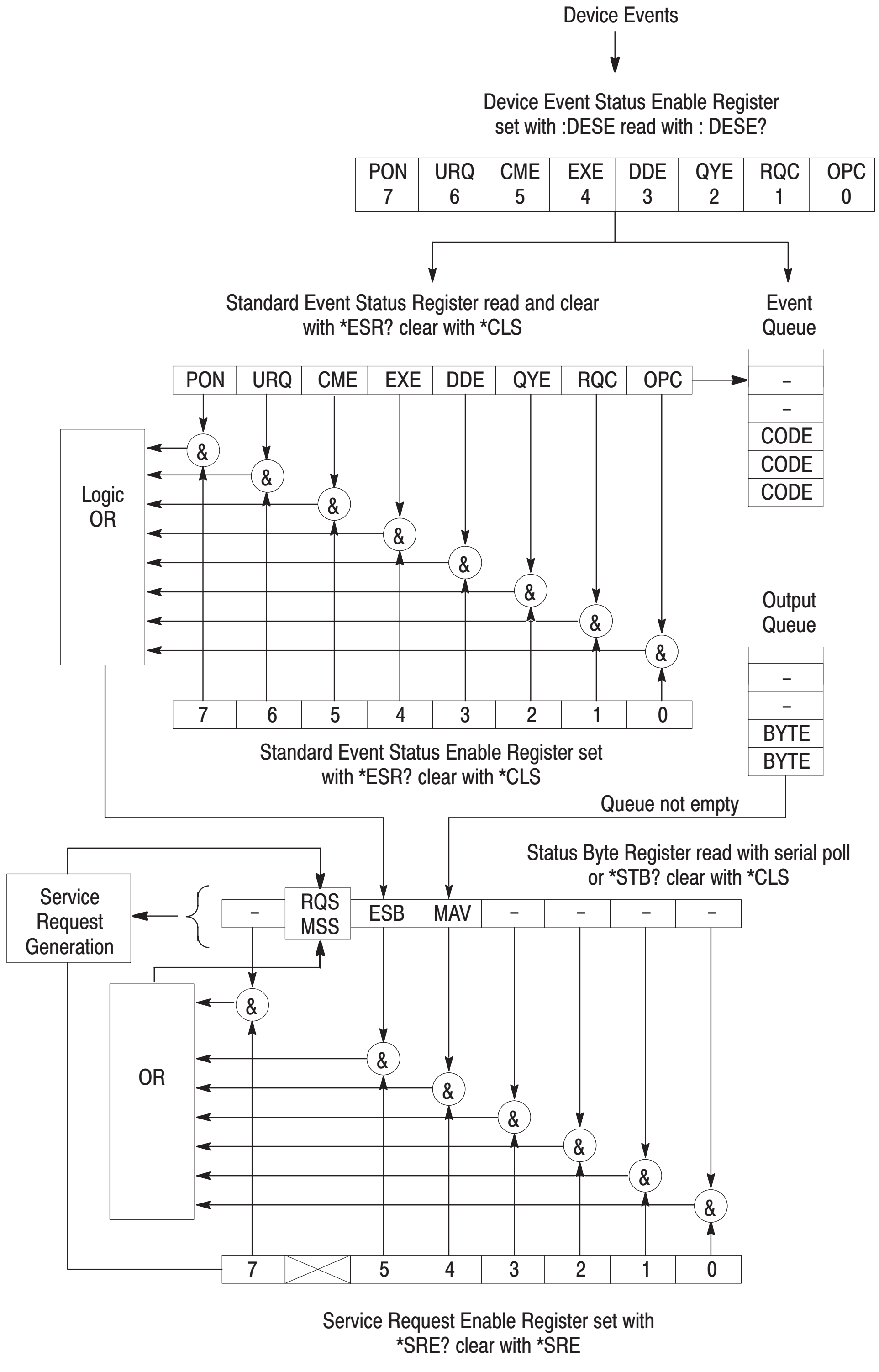


Figure 4–6: Status and Event Handling Process Overview

Appendices

Appendix A: ASCII & GPIB Code Chart

B7 B6 B5 BITS / B4 B3 B2 B1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
	CONTROL		NUMBERS SYMBOLS		UPPER CASE		LOWER CASE	
0 0 0 0	0 **NUL** 0 0	20 **DLE** 10 16	40 LA0 **SP** 20 32	60 LA16 **0** 30 48	100 TA0 **@** 40 64	120 TA16 **P** 50 80	140 SA0 **`** 60 96	160 SA16 **p** 70 112
0 0 0 1	1 GTL **SOH** 1 1	21 LL0 **DC1** 11 17	41 LA1 **!** 21 33	61 LA17 **1** 31 49	101 TA1 **A** 41 65	121 TA17 **Q** 51 81	141 SA1 **a** 61 97	161 SA17 **q** 71 113
0 0 1 0	2 **STX** 2 2	22 **DC2** 12 18	42 LA2 **"** 22 34	62 LA18 **2** 32 50	102 TA2 **B** 42 66	122 TA18 **R** 52 82	142 SA2 **b** 62 98	162 SA18 **r** 72 114
0 0 1 1	3 **ETX** 3 3	23 **DC3** 13 19	43 LA3 **#** 23 35	63 LA19 **3** 33 51	103 TA3 **C** 43 67	123 TA19 **S** 53 83	143 SA3 **c** 63 99	163 SA19 **s** 73 115
0 1 0 0	4 SDC **EOT** 4 4	24 DCL **DC4** 14 20	44 LA4 **$** 24 36	64 LA20 **4** 34 52	104 TA4 **D** 44 68	124 TA20 **T** 54 84	144 SA4 **d** 64 100	164 SA20 **t** 74 116
0 1 0 1	5 PPC **ENQ** 5 5	25 PPU **NAK** 15 21	45 LA5 **%** 25 37	65 LA21 **5** 35 53	105 TA5 **E** 45 69	125 TA21 **U** 55 85	145 SA5 **e** 65 101	165 SA21 **u** 75 117
0 1 1 0	6 **ACK** 6 6	26 **SYN** 16 22	46 LA6 **&** 26 38	66 LA22 **6** 36 54	106 TA6 **F** 46 70	126 TA22 **V** 56 86	146 SA6 **f** 66 102	166 SA22 **v** 76 118
0 1 1 1	7 **BEL** 7 7	27 **ETB** 17 23	47 LA7 **'** 27 39	67 LA23 **7** 37 55	107 TA7 **G** 47 71	127 TA23 **W** 57 87	147 SA7 **g** 67 103	167 SA23 **w** 77 119
1 0 0 0	10 GET **BS** 8 8	30 SPE **CAN** 18 24	50 LA8 **(** 28 40	70 LA24 **8** 38 56	110 TA8 **H** 48 72	130 TA24 **X** 58 88	150 SA8 **h** 68 104	170 SA24 **x** 78 120
1 0 0 1	11 TCT **HT** 9 9	31 SPD **EM** 19 25	51 LA9 **)** 29 41	71 LA25 **9** 39 57	111 TA9 **I** 49 73	131 TA25 **Y** 59 89	151 SA9 **i** 69 105	171 SA25 **y** 79 121
1 0 1 0	12 **LF** A 10	32 **SUB** 1A 26	52 LA10 ***** 2A 42	72 LA26 **:** 3A 58	112 TA10 **J** 4A 74	132 TA26 **Z** 5A 90	152 SA10 **j** 6A 106	172 SA26 **z** 7A 122
1 0 1 1	13 **VT** B 11	33 **ESC** 1B 27	53 LA11 **+** 2B 43	73 LA27 **;** 3B 59	113 TA11 **K** 4B 75	133 TA27 **[** 5B 91	153 SA11 **k** 6B 107	173 SA27 **{** 7B 123
1 1 0 0	14 **FF** C 12	34 **FS** 1C 28	54 LA12 **,** 2C 44	74 LA28 **<** 3C 60	114 TA12 **L** 4C 76	134 TA28 **** 5C 92	154 SA12 **l** 6C 108	174 SA28 **¦** 7C 124
1 1 0 1	15 **CR** D 13	35 **GS** 1D 29	55 LA13 **–** 2D 45	75 LA29 **=** 3D 61	115 TA13 **M** 4D 77	135 TA29 **]** 5D 93	155 SA13 **m** 6D 109	175 SA29 **}** 7D 125
1 1 1 0	16 **SO** E 14	36 **RS** 1E 30	56 LA14 **.** 2E 46	76 LA30 **>** 3E 62	116 TA14 **N** 4E 78	136 TA30 **^** 5E 94	156 SA14 **n** 6E 110	176 SA30 **~** 7E 126
1 1 1 1	17 **SI** F 15	37 **US** 1F 31	57 LA15 **/** 2F 47	77 UNL **?** 3F 63	117 TA15 **O** 4F 79	137 UNT **_** 5F 95	157 SA15 **o** 6F 111	177 **RUBOUT (DEL)** 7F 127
	ADDRESSED COMMANDS	UNIVERSAL COMMANDS	LISTEN ADDRESSES		TALK ADDRESSES		SECONDARY ADDRESSES OR COMMANDS	

KEY

octal → 5	PPC ←	GPIB code (with ATN asserted)
	ENQ ←	ASCII character
hex → 5	5 ←	decimal

Tektronix
REF: ANSI STD X3.4-1977
IEEE STD 488.1-1987
ISO STD 646-2973

Appendix B: Messages

Tables B–1 through B–8 list the status and event messages used in the status and event reporting system. You use the *ESR? query to make the messages available for dequeuing; you use the :EVENT?, EVMsg?, and ALLEv? queries to dequeue and return the messages. When using these queries, use the *ESR? query to make the events available for return. The messages return as shown below:

- The :EVENT? query returns the event code only.
- The EVMsg?, and ALLEv? queries return both the event code and event message in the following format:

  <event code>, “<event message ; secondary message>”

Most messages returned have both an event message, followed by a semicolon (;), and a second message which contains more detailed information. Although these secondary messages are not listed in this manual, you can use the EVMsg? and ALLEv? queries to display them.

Table B–1 lists the messages generated when the system has no events or status to report. These messages have no associated SESR bit.

Table B–1: Normal Condition

Code	Description
0	No events to report — queue empty
1	No events to report — new events pending *ESR?

Table B–2 lists the error messages generated due to improper command syntax. When these errors are encountered, check that the command is properly formatted and that it follows the syntax.

Table B–2: Command Errors (CME Bit:5)

Code	Description
100	Command error
101	Invalid character
102	Syntax error
103	Invalid separator
104	Data type error
105	GET not allowed
106	Invalid program data separator
108	Parameter not allowed
109	Missing parameter
110	Command header error
111	Header separator error
112	Program mnemonic too long
113	Undefined header
114	Header suffix out of range
118	Query not allowed
120	Numeric data error
121	Invalid character in number
123	Exponent too large
124	Too many digits
128	Numeric data not allowed
130	Suffix error
131	Invalid suffix
134	Suffix too large
138	Suffix not allowed
140	Character data error
141	Invalid character data
144	Character data too long
148	Character data not allowed
150	String data error
151	Invalid string data
152	String data too long
158	String data not allowed
160	Block data error
161	Invalid block data
168	Block data not allowed

Table B–2: Command Errors (CME Bit:5) (Cont.)

Code	Description
170	Expression error
171	Invalid expression
178	Expression data not allowed
180	Macro error
181	Invalid outside macro definition
183	Invalid inside macro definition
184	Macro parameter error

Table B–3 lists the execution errors that are detected during execution of a command.

Table B–3: Execution Errors (EXE Bit:4)

Code	Description
200	Execution error
201	Invalid while in local
202	Settings lost due to RTL
203	Invalid password
210	Trigger error
211	Trigger ignored
212	Armed ignored
213	Init ignored
214	Trigger deadlock
215	ARM deadlock
220	Parameter error
221	Settings conflict
222	Data out of range
223	Too much data
224	Illegal parameter value
225	Parameter under range
226	Parameter over range
227	Parameter rounded
230	Data corrupt or stale
231	Data questionable
240	Hardware error

Table B–3: Execution Errors (EXE Bit:4) (Cont.)

Code	Description
241	Hardware missing
250	Mass storage error
251	Missing mass storage
252	Missing media
253	Corrupt media
254	Media full
255	Directory full
256	File name not found
257	File name error
258	Media protected
260	Expression error
261	Math error in expression
262	Expression syntax error
263	Expression execution error
270	Macro error
271	Macro syntax
272	Macro execution error
273	Illegal macro label
274	Macro parameter error
275	Macro definition too long
276	Macro recursion error
277	Macro redefinition not allowed
278	Macro header not found
280	Program error
281	Cannot create program
282	Illegal program name
283	Illegal variable name
284	Program currently running
285	Program syntax error
286	Program run time error

Table B–4 lists the internal errors that can occur during operation of the waveform generator. These errors may indicate that the waveform generator needs repair.

Table B–4: Internal Errors (DDE Bit:3)

Code	Description
300	Device-specific error
310	System error
311	Memory error
312	PUD memory lost
313	Calibration memory lost
314	Save/recall memory lost
315	Configuration memory lost
330	Self-test failed
350	Queue overflow (does not affect the DDE bit)

Table B–5 lists the system event messages. These messages are generated whenever certain system conditions occur.

Table B–5: System Event

Code	Description
401	Power on
402	Operation complete
403	User request
404	Power fail
405	Request control
410	Query INTERRUPTED
420	Query UNTERMINATED
430	Query DEADLOCKED
440	Query UNTERMINATED after indefinite response

Table B–6 lists warning messages that do not interrupt the flow of command execution. These messages indicate that you may get unexpected results.

Table B–6: Warnings (EXE Bit:4)

Code	Description
500	Execution warning
501	Equation compile has aborted

Table B–7 lists internal errors that indicate an internal fault in the waveform generator.

Table B–7: Internal Warnings (DDE Bit:3)

Code	Description
600	Internal warning

Table B–8 lists status messages that are specific to the waveform generator. These messages appear when an operation starts, ends, or is in process. These messages have no associated SESR bit.

Table B–8: Device-Specific Messages

Code	Description
2000	File error
2001	Directory not empty
2002	Too many files
2003	File locked
2004	File already exists
2005	File already opened
2006	Invalid file type
2007	File type mismatch
2008	Internal memory full
2009	Invalid file format
2010	Comment error
2012	Invalid data in comment string
2020	Waveform error
2021	Waveform request is valid
2022	Too much curve data

Table B–8: Device-Specific Messages (Cont.)

Code	Description
2023	Curve data byte count error
2024	Waveform load error
2025	Internal waveform memory full
2026	Waveform size invalid
2030	Marker error
2031	Marker request is invalid
2032	Too much marker data
2040	Equation error
2042	Too many equations
2043	Equation too long
2044	Invalid equation syntax
2046	Equation compile error
2050	Sequence error
2052	Too much sequence data
2053	Invalid sequence repeat count
2054	Invalid sequence syntax
2055	Sequence load error
2056	Internal sequence memory full
2057	Recursive sequence
2058	Sequence in subsequence
2059	Sequence incomplete
2060	Autostep error
2062	Too much autostep data
2070	Date error
2071	Invalid date syntax
2072	Invalid date value
2080	Time error
2081	Invalid time syntax
2082	Invalid time value

Appendix C: Default Settings

Table C–1 lists the status of commands that are affected by power-up or the *RST universal command.

Table C–1: Factory Initialized Settings

Command Group	Header	Default Settings
Calibration & Diagnostic Commands	DIAG:SELect	ALL
Calibration & Diagnostic Commands	SELFcal:SELect	ALL
Mode Commands	MODE	CONTINUOUS
Mode Commands	TRIGger:IMPedance	HIGH
Mode Commands	TRIGger:INPut	INTERNAL
Mode Commands	TRIGger:LEVel	1.400
Mode Commands	TRIGger:OUTPut	OFF
Mode Commands	TRIGger:POLarity	POSITIVE
Mode Commands	TRIGger:SLOPe	POSITIVE
Output Commands	OUTPut:STATe	0
Output Commands	OUTPut:SYNC	END
Setup Commands	CLOCk:FREQuency	1.000E+08
Setup Commands	CLOCk:SOURce	INTERNAL
Setup Commands	OPERation	NORMAL
Setup Commands	AMPLitude	1.000
Setup Commands	FILTer	THRU
Setup Commands	OFFSet	0.000
Setup Commands	WAVeform	" "
Status & Event Commands	DESE	256
Status & Event Commands	*ESE	0
Status & Event Commands	*SRE	0
System Commands	HEADer	1
System Commands	VERBose	1
Waveform Commands	DATA:DESTination	"GPIB.WFM"
Waveform Commands	DATA:ENCDG	RPBINARY
Waveform Commands	DATA:SOURce	"CH1"
Waveform Commands	WFMPre:ENCDG	BIN
Waveform Commands	WFMPre:BN_FMT	RP
Waveform Commands	WFMPre:BYT_NR	2

Table C–1: Factory Initialized Settings (Cont.)

Command Group	Header	Default Settings
Waveform Commands	WFMPre:BIT_NR	12
Waveform Commands	WFMPre:BYT_OR	MSB
Waveform Commands	WFMPre:CRVCHK	NONE
Waveform Commands	WFMPre:WFID	" "
Waveform Commands	WFMPre:NR_PT	0
Waveform Commands	WFMPre:PT_FMT	Y
Waveform Commands	WFMPre:XUNIT	"S"
Waveform Commands	WFMPre:XINCR	1.000E–08
Waveform Commands	WFMPre:PT_OFF	0
Waveform Commands	WFMPre:XZERO	0.000
Waveform Commands	WFMPre:YUNIT	"V"
Waveform Commands	WFMPre:YMULT	2.442E–04
Waveform Commands	WFMPre:YZERO	0.000
Waveform Commands	WFMPre:YOFF	2.047E+03

Appendix D: Sample Waveform Library

Table D–1 and Table D–2 list the waveform samples found on the sample waveform library disk that comes with the waveform generator as a standard accessory. This appendix provides a listing of the equations and waveforms for each file.

Table D–1: Waveform Samples

Sample Number	File Name	Description
1	SIN.EQC SIN.PAT	Sine wave signal
2	TRI.EQC TRI.PAT	Triangle wave signal
3	RAMP.EQC RAMP.PAT	Ramp wave signal
4	SQU.EQC SQU.PAT	Square wave signal
5	GAUSS_P.EQC GAUSS_P.PAT	Gaussian pulse
6	LORENT_P.EQC LORENT_P.PAT	Lorentz pulse
7	SINX_P.EQC SINX_P.PAT	Sin(x)/x pulses
8	SQU_SIN.EQC SQU_SIN.PAT	Squared sine pulse
9	D_EXP.EQC D_EXP.PAT	Rising, falling exponential pulse
10	NYQUIST.EQC NYQUIST.PAT	Nyquist Pulse
11	LIN_SWP.EQC LIN_SWP.PAT	Linear Frequency Sweep
12	LOG_SWP.EQC LOG_SWP.PAT	Logarithmic Frequency Sweep
13	AM.EQC AM.PAT	Amplitude modulated signal
14	FM.EQC FM.PAT	Frequency modulated signal
15	DMP_SIN.EQC DMP_SIN.PAT	Damped sine wave

Table D–1: Waveform Samples (Cont.)

Sample Number	File Name	Description
16	PWM_P.EQC PWM_P.PAT	Pulsewidth modulated signal
17	PRBS15.PAT PRBS15.SQC	NRZ pseudo-random pulse
18	MDISK_W.EQC MDISK_W.PAT	Electromagnetic disk signal 1
19	MDISK_RD.PAT	Magnetic disk read signal 2

Table D–2: NTSC Directory

Sample Number	File Name	Description
1	$CB1-2H.SQC	Colorbar composite
2	$CB1-(Y).SQC	Colorbar luminance
3	$CB1-(C).SQC	Colorbar chrominance
4	$M-BURST.SQC	Multiburst
5	$RAMP.SQC	Ramp signal
6	$SWEEP10.SQC	Sweep signal
7	$SMPTE.SQC	SMPTE colorbar
8	$IYQB.SQC	YIQB signal
9	$RBLU.SQC	Bluebar signal

Sine Wave Signal

The Sine Wave Signal waveform has the following characteristics:

- Signal frequency: 1 MHz
- Waveform points: 200
- Clock frequency: 200 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,1μs)<LF>sin(2*pi*1e6*t)

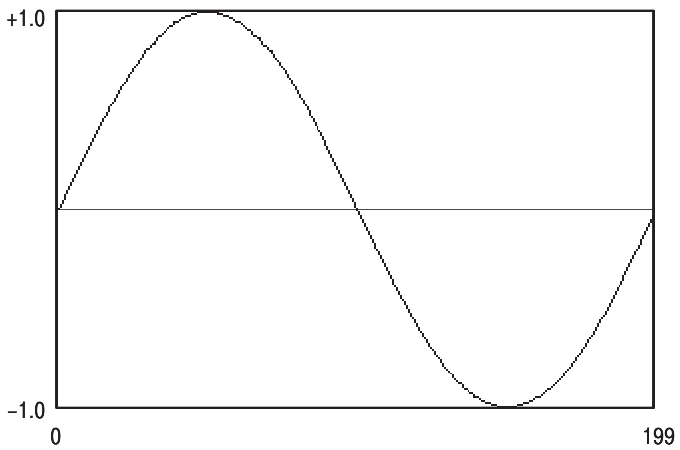


Figure D–1: Sine Wave Signal

Triangle Wave Signal

The Triangle Wave Signal waveform has the following characteristics:

- Signal frequency: 1 MHz
- Waveform points: 200
- Clock frequency: 200 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,1µs)<LF>range(0,0.25µs)<LF>x<LF>
range(0.25µs,0.75µs)<LF>1–2*x<LF>range(0.75µs,1µs)<LF>–1+x

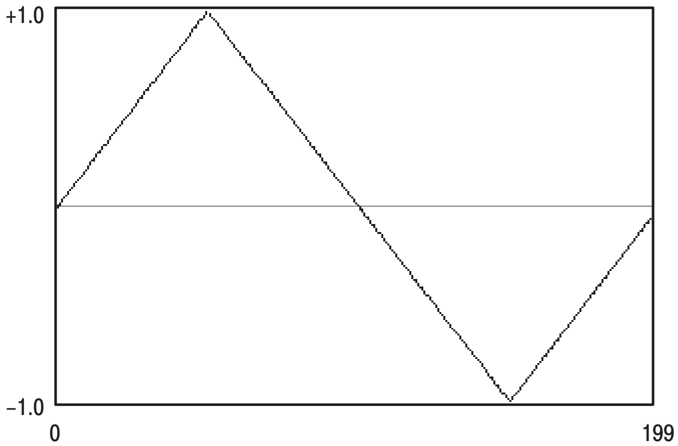


Figure D–2: Triangle Wave Signal

Ramp Signal

The Ramp Signal waveform has the following characteristics:

- Signal frequency: 1 MHz
- Waveform points: 200
- Clock frequency: 200 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,1μs)<LF>–1+2*x

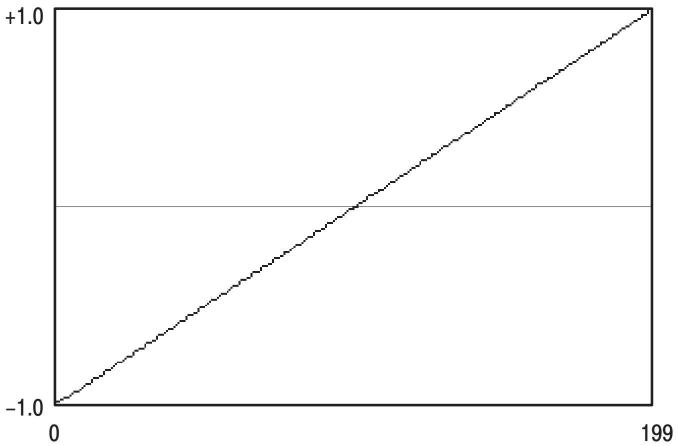


Figure D–3: Ramp Signal

Square Wave Signal

The Square Wave Signal waveform has the following characteristics:

- Signal frequency: 1 MHz
- Waveform points: 200
- Clock frequency: 200 MHz
- Output parameters: Filter 50 MHz
- Equation: range(0,1µs)<LF>range(0,0.5µs)<LF>1<LF> range(0.5µs,1µs)<LF>–1

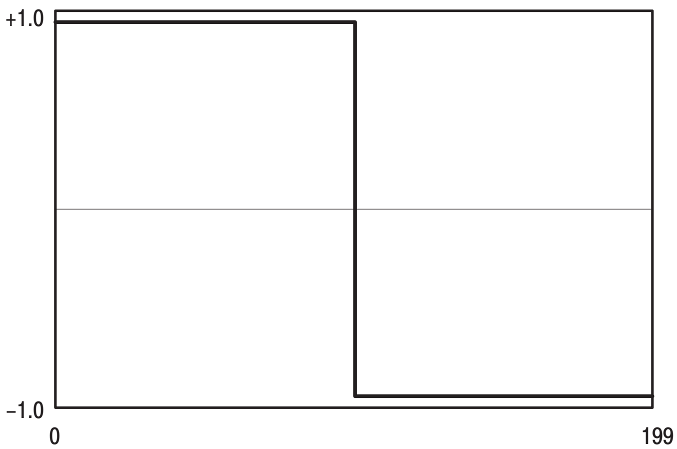


Figure D–4: Square Wave Signal

Gaussian Pulse

The Gaussian Pulse waveform has the following characteristics:

- Waveform points: 256
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,2.56μs)<LF>k0=0.3e–6<LF>k1=1.28e–6<LF> exp(–ln(2)*((2*(t–k1/k0)^2))

  Pulse Width (k0) is 0.3 μs

  Peak Location (k1) is 1.28 μs

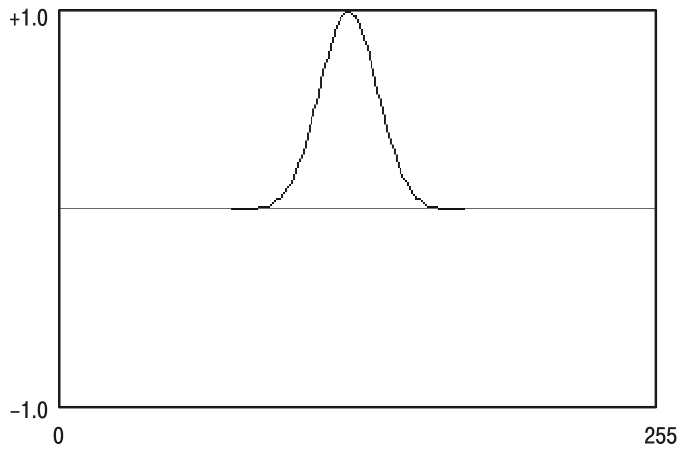


Figure D–5: Gaussian Pulse

Lorentz Pulse

The Lorentz Pulse waveform has the following characteristics:

- Waveform points: 256
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,2.56μs)<LF>k0=0.3e–6<LF>k1=1.28e–6<LF> 1/(1+(2*(t–k1)/k0)^2)

  Pulse Width (k0) is 0.3 μs

  Peak Location (k1) is 1.28 μs

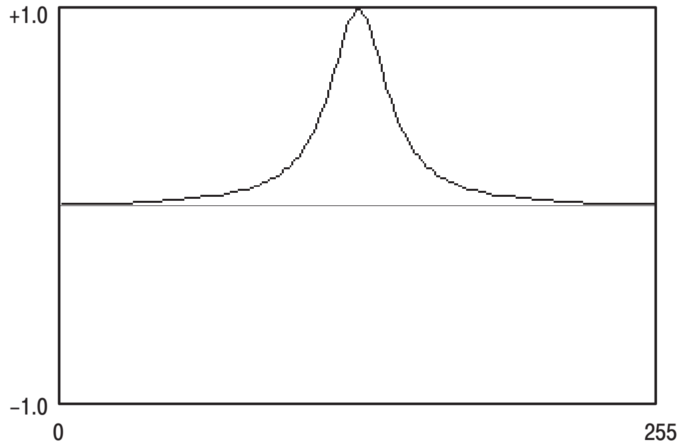


Figure D–6: Lorentz Pulse

Sin(x)/x Pulse

The Sin(x)/x Pulse waveform has the following characteristics:

***NOTE**. 326 cycles are needed to utilize the 12-bit vertical resolution.*

- Waveform points: 4000
- Clock frequency: 100 MHz
- Output parameters: Filter Through
- Equation: range(0,40μs)<LF>k0=2.5e6<LF>k1=20e–6<LF>
  sin(2*pi*k0*(t–k1))/(2*pi*k0*(t–k1))

  Sine Frequency (k0) is 2.5 MHz

  Peak Location (k1) is 20 μs

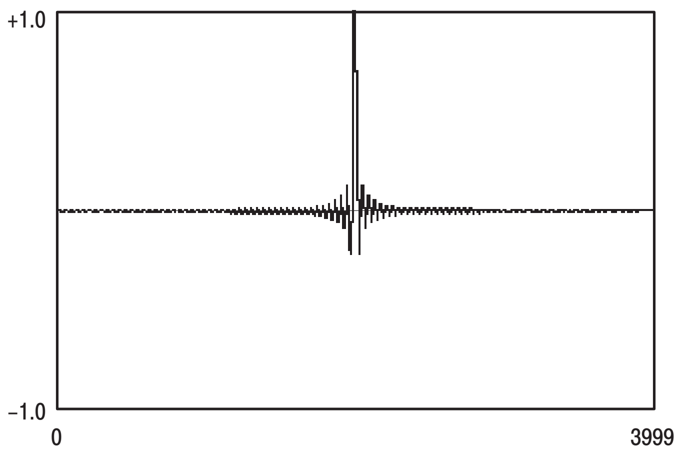


Figure D–7: Sin(x)/x Pulse

Squared Sine Pulse

The Squared Sin Pulse waveform has the following characteristics:

- Waveform points: 1000
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,3μs)<LF>0<LF>range(3μs,7μs)<LF>(cos(2*pi*(x–0.5))+1)/2<LF>range(7μs,10μs)<LF>0

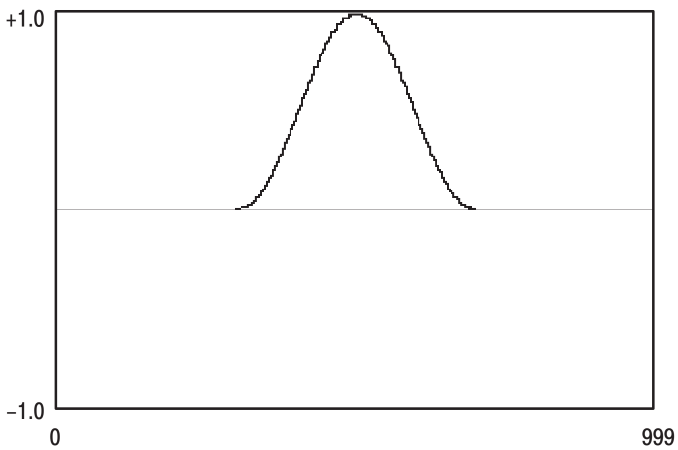


Figure D–8: Squared Sine Pulse

Rising, Falling Exponential Pulse

The Rising, Falling Exponential Pulse waveform has the following characteristics:

- Waveform points: 10,000
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,100μs)<LF>k1=1e–6<LF>k2=10e–6<LF> exp(–t/k2)–exp(–t/k1)<LF>norm()

  Rising Time Constant (k1) is 1 μs

  Falling Time Constant (k2) is 10 μs

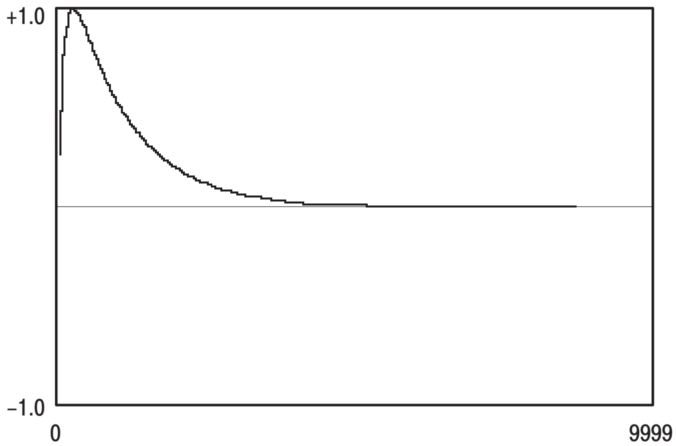


Figure D–9: Rising, Falling Exponential Pulse

Nyquist Pulse

The Nyquist Pulse waveform has the following characteristics:

- Waveform points: 1000
- Clock frequency: 100 MHz
- Output parameters: Filter 50 MHz
- Equation: range(0,10μs)<LF>k0=200e–9<LF>k1=5e–6<LF>k2=0.5<LF>
  cos(pi*k2*(t–k1)/k0)/(1–(2*k2*(t–k1)/k0)^2)<LF>
  v*sin(pi*(t–k1)/k0)/(pi*(t–k1)/k0)

  Date Period (k0) is 200 ns

  Peal Location (k1) is 5 μs

  Excess Bandwidth Factor (k2) is 0.5 [0 to 1]

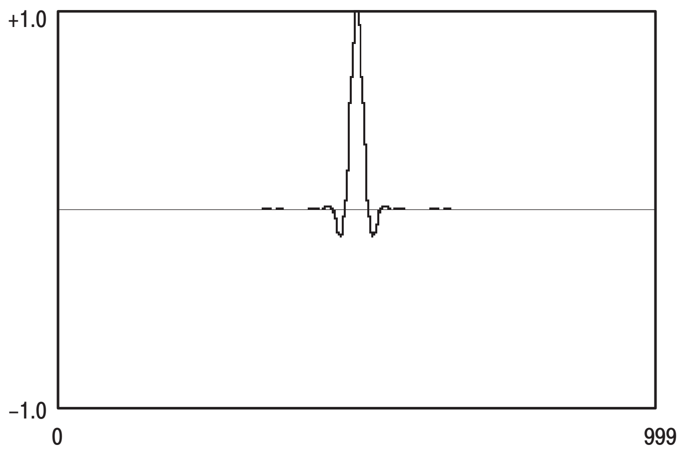


Figure D–10: Nyquist Pulse Waveform

Linear Frequency Sweep

The Linear Frequency Sweep waveform has the following characteristics:

- Waveform points: 16,000
- Clock frequency: 10 MHz
- Output parameters: Filter 50 MHz
- Equation: range(0,1.6ms)<LF>k0=1.6e–3<LF>k1=5e3<LF>
  k2=50e3<LF>sin(2*pi*k1*t+2*pi*(k2–k1)*(t^2)/2/k0)

  Sweep Period (k0) is 1.6 ms

  Start Frequency (k1) is 5 kHz

  End Frequency (k2) is 5 kHz

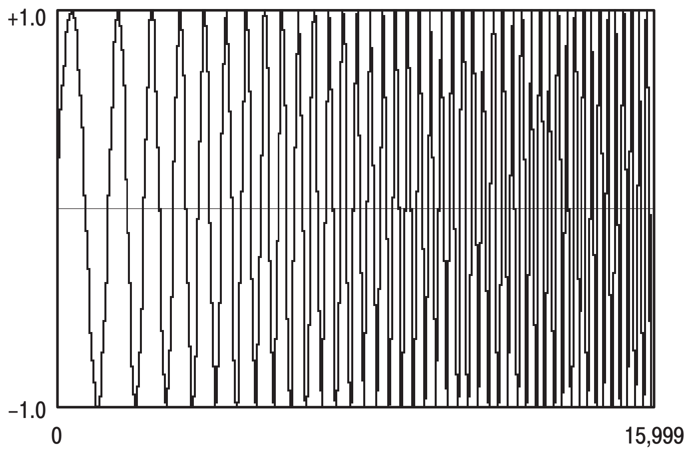


Figure D–11: Linear Frequency Sweep Signal

Logarithmic Frequency Sweep

The Logarithmic Frequency Sweep waveform has the following characteristics:

- Waveform points: 22,000
- Clock frequency: 100 MHz
- Output parameters: Filter 50 MHz
- Equation: range(0,2.2ms)<LF>k0=2.2e–3<LF>k1=5e3<LF>k2=50e3<LF>k3=ln(k2/k1)<LF>sin(2*pi*k1*k0/k3*(exp(k3*x)–1))

  Sweep Period (k0) is 2.2 ms

  Start Frequency (k1) is 5 kHz

  End Frequency (k2) is 5 kHz

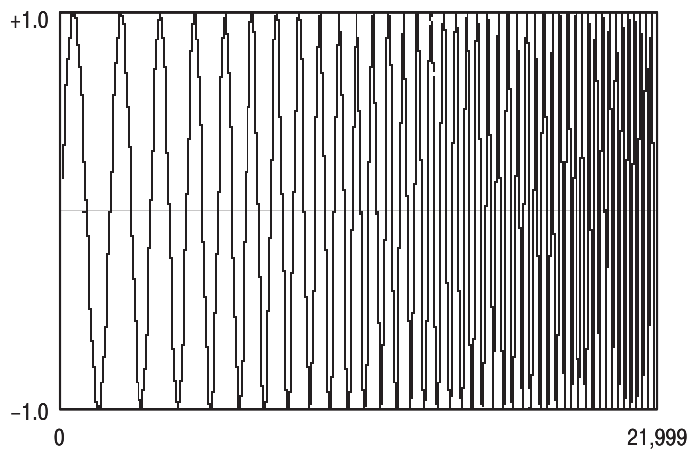


Figure D–12: Logarithmic Frequency Sweep Signal

Amplitude Modulated Signal

The Amplitude Modulated signal has the following characteristics:

- Waveform points: 20,000
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,0.2ms)<LF>k0=5e3<LF>k1=5e6<LF>k2=0.5<LF>
(1+k2*cos(2*pi*k0*t))*cos(2*pi*k2*t)

  Modulated Signal Frequency (k0) is 5 kHz

  Carrier Signal Frequency (k1) is 5 MHz

  Modulation Factor (k2) is 50% [0.5]

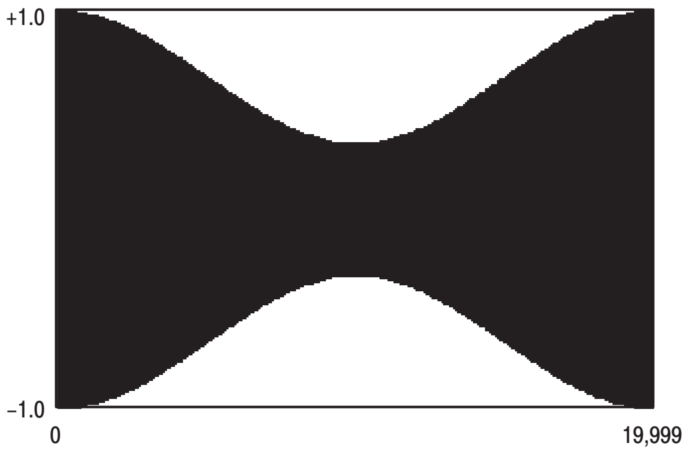


Figure D–13: Amplitude Modulated Signal

Frequency Modulated Signal

The Frequency Modulated signal has the following characteristics:

- Waveform points: 2000
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,20μs)<LF>k0=50e3<LF>k1=2.5e6<LF>k2=2e6<LF> sin(2*pi*k1*t+k2/k0*sin(2*pi*k0*t))

  Modulated Signal Frequency (k0) is 50 kHz

  Carrier Signal Frequency (k1) is 2.5 MHz

  Frequency Deviation (k2) is 2 MHz

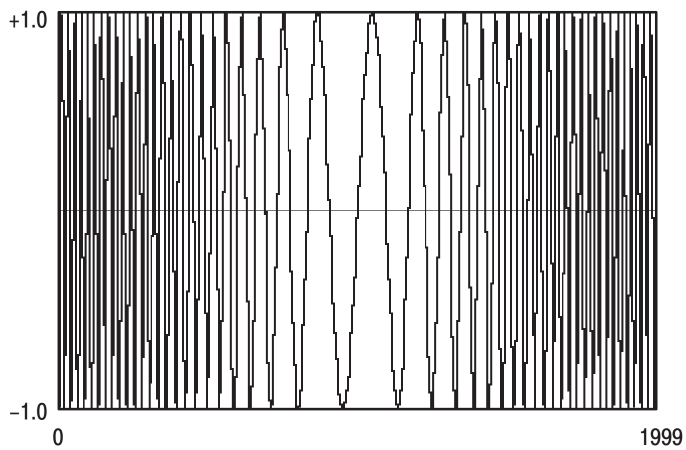


Figure D–14: Frequency Modulated Signal

Damped Sine Wave

The Damped Sine Wave waveform has the following characteristics:

- Waveform points: 4000
- Clock frequency: 100 MHz
- Output parameters: Filter 20 MHz
- Equation: range(0,40μs)<LF>k0=2e–3<LF>k1=12.66e–12<LF> k2=k0*k1<LF>k3=6e–6<LF>exp(–r/k3)*sin(1–sqrt(k2)*t)

  Inductance (k0) is 2 mh

  Capacitance (k1) is 12.6 pF

  Damping Time Constant (k3) is 6 μs

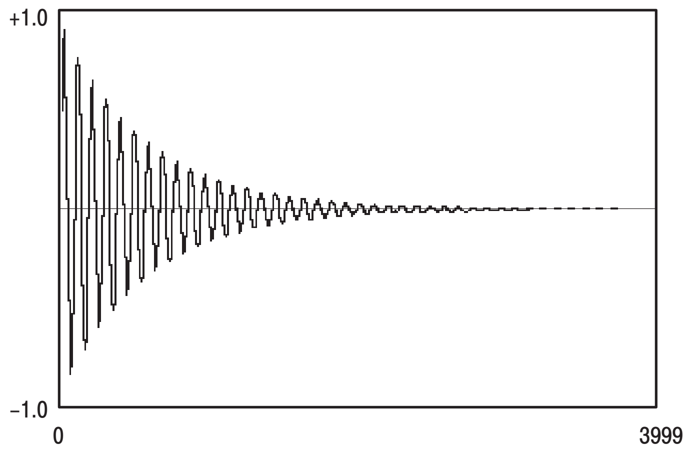


Figure D–15: Damped Sine Wave

Pulse Width Modulated Signal

To Pulse Width Modulated Signal has the following characteristics:

- Waveform points: 1024
- Clock frequency: 1 MHz
- Output parameters: Filter Through
- Equation: range(0,1024μs)<LF>–1<LF>range(0,16μs)<LF>v+2<LF> range(32μs,50μs)<LF>v+2<LF>range(64μs,84μs)<LF>v+2<LF> range(96μs,118μs)<LF>v+2<LF>range(128μs,152μs)<LF>v+2<LF> range(160μs,186μs)<LF>v+2<LF>range(192μs,220μs)<LF>v+2<LF> range(224μs,254μs)<LF>v+2<LF>range(256μs,318μs)<LF>v+2<LF> range(320μs,348μs)<LF>v+2<LF>range(352μs,378μs)<LF>v+2<LF> range(384μs,408μs)<LF>v+2<LF>range(416μs,438μs)<LF>v+2<LF> range(448μs,468μs)<LF>v+2<LF>range(480μs,498μs)<LF>v+2<LF> range(512μs,528μs)<LF>v+2<LF>range(544μs,558μs)<LF>v+2<LF> range(576μs,589μs)<LF>v+2<LF>range(608μs,618μs)<LF>v+2<LF> range(640μs,648μs)<LF>v+2<LF>range(672μs,678μs)<LF>v+2<LF> range(704μs,708μs)<LF>v+2<LF>range(736μs,738μs)<LF>v+2<LF> range(800μs,802μs)<LF>v+2<LF>range(832μs,836μs)<LF>v+2<LF> range(864μs,870μs)<LF>v+2<LF>range(896μs,904μs)<LF>v+2<LF> range(928μs,938μs)<LF>v+2<LF>range(960μs,972μs)<LF>v+2<LF> range(992μs,1006μs)

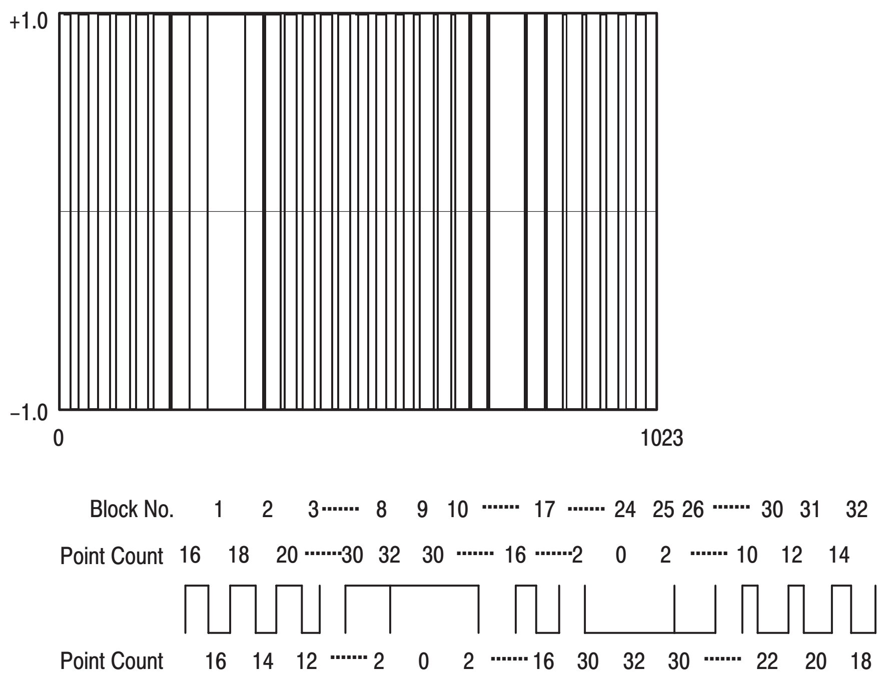


Figure D–16: Pulse Width Modulated Signal

NRZ Pseudo-Random (PRBS) Signal

The NRZ Pseudo-Random Signal stored in waveform file PRBS15.PAT (Figure D–17) is a 32 Kbit (15 step) M-series pseudo-random signal generated with the shift register shown in Figure D–18. All 1s are set in the initial register and the data is changed every two samples.

- Waveform points: 32,767
- Clock frequency: 100 MHz
- Output parameters: Filter Through

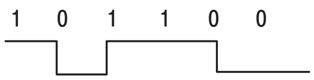


Figure D–17: NRZ Pseudo-random Signal

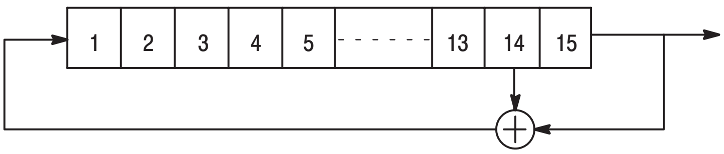


Figure D–18: Pseudo-random Signals Generated with Shift Register

The NRZ Pseudo-Random Signal stored in sequence file PRBS15.SQC (Figure D–19) is produced by repeating the PRBS15.PAT file eight times. To display the sequence, use the `SEQUence:EXPand` command (page 3–70).

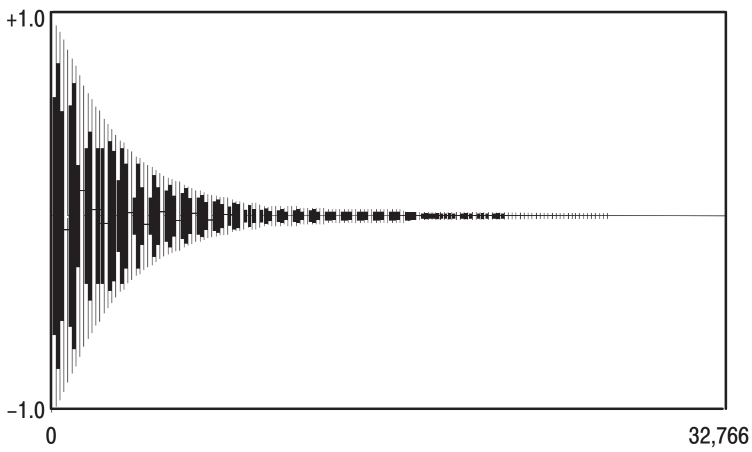


Figure D–19: NRZ Pseudo-random Signal

Electromagnetic Disk Signal 1

The Electromagnetic Disk Signal 1 waveform has the following characteristics:

- Waveform points: 200
- Clock frequency: 100 MHz
- Output parameters: Filter 50 MHz
- Equation: range(0,0.08μs)<LF>k1=1/3<LF>k2=1/9<LF>
  sin(2*pi*x)–sin(3*2*pi*x)+sin(5*2*pi*x)<LF>norm()

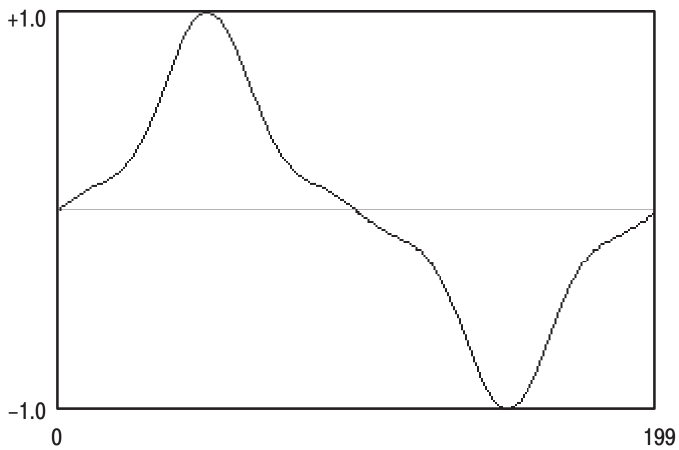


Figure D–20: Electromagnetic Disk Signal 1 Waveform

Electromagnetic Disk Signal 2

The Electromagnetic Disk Signal 2 waveform has the following characteristics:

- Waveform points: 768
- Clock frequency: 100 MHz
- Output parameters: Filter 50 MHz
- The impulse response waveform is Gaussian waveform (GAUSS_P.PAT)

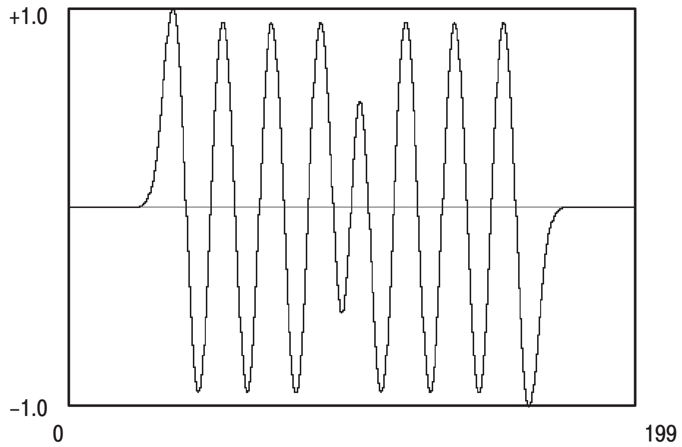


Figure D–21: Electromagnetic Disk Signal 2 Waveform

NTSC Directory

The NTSC Directory contains nine sequence files that produce television test signals. The sequence files have the following characteristics:

- Waveform points: One horizontal line is 3640 points
- Output waveform points: One frame is 382,200 points
- Clock frequency: 57.27 MHz (3.579545 MHz × 16)
- Output parameters: Filter 20 MHz

See Table D–2 on page D–2 for a summary of the sequence file names.

***NOTE**. The SMPTE Colorbar signal is produced from the Colorbar Composite, IYQB, and Bluebar sequence files.*

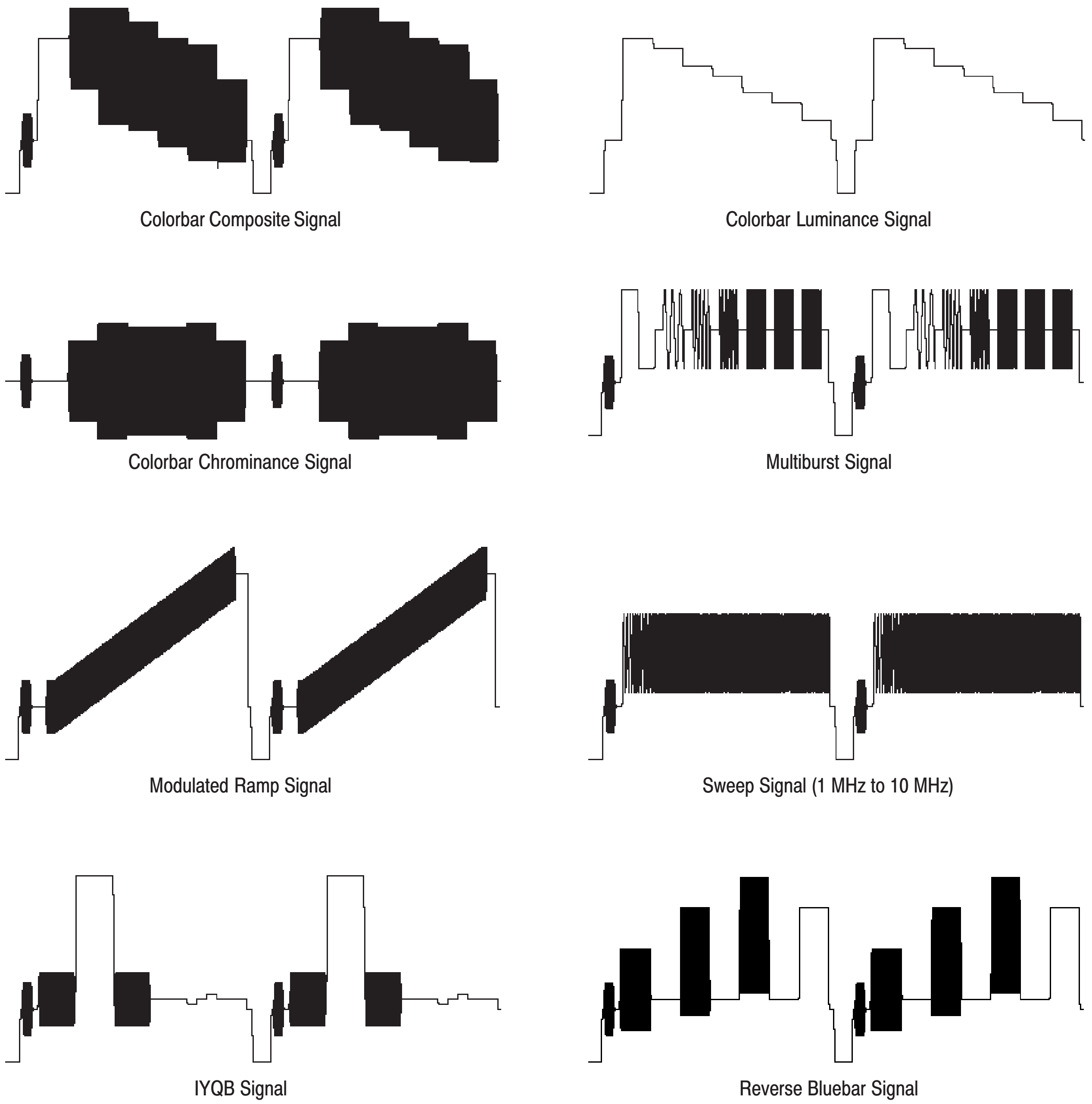


Figure D–22: NTSC Directory Signals

Appendix E: Specifications

Introduction

This section contains a collection of tables that describe the electrical, environmental, and mechanical characteristics of the VX4792 Arbitrary Waveform Generator. The specification tables contain up to three columns: Characteristic, Performance Requirement, and Supplemental Information. Each column is described below.

Characteristic column. Provides the name of each characteristic.

Performance Requirement column. Describes the limits of the warranted characteristics. Limits within this column are typically verified by performing tests located within the *Performance Verification* procedure (*Appendix F*).

Supplemental Information column. Explains performance requirements, or provides typical performance information.

Performance Conditions

The performance requirements found in this section are only valid under the following conditions:

- The waveform generator must be operating as an integral part of an appropriately configured mainframe
- The operating conditions must comply with those specified for the system mainframe in which the waveform generator is installed
- The test environment must comply with the limits described in Table E–2 Environmental Characteristics
- The instrument must have been calibrated at an ambient temperature between +20° C to +30° C following a 20 minute warm up period
- All tolerance limits apply after a 20 minute warm up period

Table E–1: Electrical Characteristics

Characteristics	Performance Requirement		Supplemental Information
Power Requirements	I_{Pm}*1	I_{Dm}*2	Typical Current
+5 V	7.8 A	1.20 A	7.1 A
–5.2 V	4.4 A	0.12 A	4.0 A
–2 V	2.2 A	0.06 A	2.0 A
+12 V	0.4 A	0.02 A	0.3 A
–12 V	0.3 A	0.03 A	0.2 A
+24 V	0.3 A	0.04 A	0.3 A
–24 V	0.2 A	0.08 A	0.2 A
VXI Interface			
VXIbus Revision Level			VXI System Specification Revision 1.3
Interface Type			Message Based
Protocols			Word Serial (WSP)
VXI Bus			ECLTRG0-1 (Sync. Trigger) LBUS0-11 (ECL Level)
Arbitrary Waveforms			
Waveform Memory			
Memory Length			
Waveform Data	256 Kwords X 12 Bits		
Marker 1 Data	256 Kwords X 1 Bit		
Marker 2 Data	256 Kwords X 1 Bit		
Waveform	64 to 256 K Points		Multiple of 8
Sequence Memory	8 Kwords		32 bits per word
Scan Counter	1 to 64 K (16 Bits)		
Sequence Memory	1 to 64 K (16 Bits)		
Rate Clock			
Frequency Range	10 Hz to 250 MHz		
Accuracy			
+10° C to +40° C	0.01%		
+15° C to +30° C	0.005%		
Resolution	0.1% to 0.01%		

*1 **Peak module current**

*2 **Dynamic module current**

Table E–1: Electrical Characteristics (Cont.)

Characteristics	Performance Requirement	Supplemental Information
Main Output	Measured with 50 Ω Load	
Amplitude		
Program Range	50 mV_{p-p} to 5 V_{p-p}	
Program Steps	1 mV	
Resolution	1/4096	12 bits
DC Accuracy	50 mV_{p-p} to 5 V_{p-p}	
50 mV to 500 mV	± (0.5% of Amplitude + 5 mV)	No offset at 1 MHz clock
501 mV to 5 V	± (1% of Amplitude + 25 mV)	No offset at 1 MHz clock
Offset		
Program Range	–2.5 V to +2.5 V	
Program Steps	5 mV	
Accuracy	± (1% of Offset + 5 mV)	Waveform is 0 VDC and amplitude is 50 mV
Pulse Response		
+15° C to +30° C		
Flatness	≤ 3% After 20 ns From Edges	
Aberration	≤ (7% + 10 mV)	
+10° C to +40° C		
Rise/Fall Time	< 4.2 ns	
Flatness	≤ 5% After 20 ns From Edges	
Aberration	≤ (9% + 10 mV)	
Impedance		50 Ω typical
Harmonic Distortion		Measured at 250 MHz clock, 0.5 V amplitude, 5000 points for sine data, no offset, and no filter
2nd Harmonics		≤ –40 dBc
3rd Harmonics		≤ –50 dBc
Skew Between Modules	≤ 4 ns	Filter selection is THROUGH

Table E–1: Electrical Characteristics (Cont.)

Characteristics	Performance Requirement	Supplemental Information
Operating Mode		The trigger source can be an external signal, a VXI trigger from the resource manager, or a trigger command from the resource manager
Continuous	Generates a waveform or sequence continuously	
Triggered	Generates a waveform or sequence once when triggered	
Burst	Generates a waveform or sequence for a specified burst count time when triggered	
Gated	Generates a waveform or sequence continuously while the external trigger signal is true, or after receiving a `STARt` or `TRIGger` command	
Waveform Advance	Generates waveforms defined in the sequence file continuously; a trigger signal advances to the next waveform within the sequence file	
Autostep	Generates waveforms or sequences defined in the autostep file once when triggered; a trigger signal advances to the next waveform or sequence within the autostep file, which may include new output parameters (for example, new amplitude or offset settings)	
Arithmetic Operation		
AM		
Sensitivity	2 V_{p-p} (± 5%) Signal Produces 100% Modulation	100% modulation at +1 V 50% modulation at 0 V 0% modulation at –1 V
Frequency Response	DC to 4 MHz	–3 dB

Table E–1: Electrical Characteristics (Cont.)

Characteristics	Performance Requirement	Supplemental Information
Filters		
3 dB Cutoff Frequency		
1 MHz		1 MHz ±20%
5 MHz		5 MHz ±20%
20 MHz		20 MHz ±20%
50 MHz		50 MHz ±20%
Delay		
1 MHz		390 ns typical
5 MHz		78 ns typical
20 MHz		18 ns typical
50 MHz		11 ns typical
Characteristics		Low pass filter with Bessel characteristics
Auxiliary Outputs		
Sync		
Amplitude	> 1.2 V into 50 Ω > 2.4 V into open circuit	
Impedance		50 Ω typical
Duration		100 ns ±20%
Sync→Signal Delay		10 ns typical
Marker 1		
Amplitude	2.5V (+5%/–10%) into 50 Ω 5V (+5%/–10%) into open circuit	
Rise/Fall Time	< 8 ns	
Impedance		50 Ω typical
Marker→Signal Delay		10 ns typical
Marker 2		
Amplitude	2.5V (+5%/–10%) into 50 Ω 5V (+5%/–10%) into open circuit	
Rise/Fall Time	< 8 ns	
Impedance		50 Ω typical
Marker→Signal Delay		10 ns typical

Table E–1: Electrical Characteristics (Cont.)

Characteristics	Performance Requirement	Supplemental Information
Clock		
Amplitude	1 V ± 0.3 V into 50 Ω	
Impedance		50 Ω typical
Auxiliary Inputs		
Trigger		
Threshold		
Range	–5 V to +5 V	
Resolution	0.1 V	
Accuracy	± (5% of Level + 0.1 V)	
Pulse Width	15 ns Minimum	
Input Swing	0.2 V Minimum	
Input Voltage	10 V_{p-p} with 1 MΩ Impedance 5 V_{RMS} with 50 Ω Impedance	
Impedance		1 MΩ with 30 pF maximum capacitance, or 50 Ω
Trigger→Signal Delay		
Internal Clock		50 ns typical
External Clock		50 ns + 1 clock cycle is typical
Trigger Holdoff	< 1µs Maximum (Excluding Auto Step mode)	
AM		
Range	2 V_{p-p} (–1 V to +1 V) for 100% Modulation	± 5%
Input Voltage	± 5 V Maximum	
Impedance		10 kΩ typical
Clock		
Threshold Level	0.5 V ± 0.1 V	
Pulse Width	2 ns Minimum	
Input Swing	0.8 V Minimum	
Input Voltage	± 2 V Maximum	
Impedance		50 Ω typical
Frequency Range	Up to 250 MHz	

Table E–2: Environmental Characteristics

Characteristics	Performance Requirement
Temperature	
Operating	0° C to +40° C
Nonoperating	–20° C to +70° C
Humidity	
0° C to +30° C	95% Relative Humidity
+30° C to +40° C	75% Relative Humidity
Altitude	
Operating	3.05 km (10,000 feet) Derate maximum operating temperature –1° C for every 300 m (1,000 feet) above 1.5 km (5,000 feet)
Nonoperating	4.57 km (15,000 feet)
Vibration	Withstands 0.33 mm (0.013 inches)$_{p\text{-}p}$, 5 Hz to 55 Hz sinewave, 15 minutes each axis, 10 minutes each axis at resonance or 55 Hz
Shock	Withstands 30 g (half sine), 11 ms duration, 3 shocks in each direction along 3 major axes, 18 total shocks
Packaged Product Vibration and Shock	Qualified under National Safe Transit Association Preshipment Test Procedures 1A-B-1 and 1A-B-2

Table E–3: Mechanical Characteristics

Characteristics	Performance Requirement
Module Size	D-size, Double wide (requires two slots in the mainframe)
Overall Dimensions	
Height	262 mm (10.3 in)
Width	61 mm (2.4 in)
Depth	368 mm (14.5 in)
Weight	
Net	3 kg (6.7 lbs)
Shipping	3.5 kg (7.7 lbs)
Cooling	5 l/s @ 0.5 mm H_2O for a 10° C rise (average of two slots) The temperature rises are generally different depending on the left slot part or the right slot part in this module. When using the VX1410 as a mainframe, they are specified to; 15 ° C rise in the left slot part and 7 ° C rise in the right slot part.

Table E–4: Certifications and compliances [*3, *4, *5]

EC Declaration of Conformity	Meets intent of Directive 89/336/EEC for Electromagnetic Compatibility. Compliance was demonstrated to the following specifications as listed in the Official Journal of the European Communities: EN 55011 — Class A Radiated and Conducted Emissions EN 50081-1 Emissions: EN 60555-2 — AC Power Line Harmonic Emissions EN 50082-1 Immunity: IEC 801-2 — Electrostatic Discharge Immunity IEC 801-3 — RF Electromagnetic Field Immunity IEC 801-4 — Electrical Fast Transient/Burst Immunity IEC 801-5 — Power Line Surge Immunity
FCC Compliance	Emissions comply with FCC Code of Federal Regulations 47, Part 15, Subpart B, Class A Limits
Certifications	Canadian Standards Association certified to Standard CAN/CSA-C22.2 No. 231.

[*3] **High-quality double shielded cables with low impedance must be used to ensure compliance to the listed standards.**

[*4] **Low Voltage Directive 73/23/ECC does not apply to this product.**

[*5] **This product complies when installed into a Tektronix VX4521 Resource Manager with a starting serial number of B020284.**

Appendix F: Performance Verification

This section contains a series of procedures to verify that the VX4792 Arbitrary Waveform Generator performs as warranted. The procedures are arranged in nine logical groupings, presented in the following order:

- Operating Mode Checks
- External AM Operation Check
- Clock Frequency and Amplitude Checks
- Gain and Offset Accuracy Checks
- Pulse Response Check
- SYNC OUTPUT and MARKER OUTPUT Amplitude Checks
- External Trigger Level Accuracy Check
- External CLOCK IN Check
- Synchronous Operation Check

Before you perform the performance verification procedures, you must complete the *System Setup* located on page F–8.

Levels of Testing

Two levels of performance testing are available to test the waveform generator. The test you perform depends upon your requirements. The tests are described below.

Functional Check Procedure. This procedure, located on page 1–11, tests the installed module to verify that it is operating properly. Once the waveform generator is installed into a VXIbus mainframe with a suitable computer interface, no additional test equipment is needed to perform the functional check.

The *Functional Check Procedure* is short, requires no external equipment, and performs extensive functional and accuracy testing. Use this check to quickly determine if the waveform generator is suitable for putting into service, such as when it is first received.

Performance Verification Procedures. This group of procedures, which begin on page F–11 with *Operating Mode Checks*, provides a more extensive test to verify performance of the waveform generator. The procedures involve direct checking of warranted specifications. The procedures require more time than the *Functional Check Procedure*, and external test equipment is required (see *Equipment Required* on page F–3). Use the performance verification procedures to verify that the waveform generator performance is within warranted limits.

Test Interval

To ensure accurate operation of the waveform generator, check the performance every 2000 hours, or once every 12 months if you use the waveform generator intermittently.

Conventions

Throughout the procedures within this section, the following conventions apply:

General Format. Each test procedure uses the following general format:

- Title of Test
- Equipment Required
- Prerequisites
- Procedure

Step Sequence. Each procedure consists of steps, substeps, and subparts. The steps are arranged in the following sequence:

1. First Step.
   a. First Substep.
   b. Second Substep.
      - First subpart.
      - Second subpart.
2. Second Step.

Command Entry. When instructed to enter commands on the keyboard of your computer, the text to be entered is formatted as shown in the following example:

1. To change the waveform generator controls, type:

   `IBWRT "AMPL 1V" <RETURN>`

   where `<RETURN>` is a carriage return. You must type every character shown.

Prerequisites

The tests in this section comprise an extensive, valid confirmation of performance and functionality, when the following requirements are met:

- The waveform generator must be operating as an integral part of an appropriately configured mainframe
- The operating conditions must comply with those specified for the system mainframe in which the waveform generator is installed
- The test environment must comply with the limits described in Table E–2 *Environmental Characteristics*
- The waveform generator must have been calibrated at an ambient temperature between +20° C to +30° C following a 20 minute warm up period
- The waveform generator must be operating for at least 20 minutes before the tests are performed

Equipment Required

The performance verification procedures have been written using the test equipment listed in Table F–1. If alternate equipment is used, the equipment must meet or exceed the minimum requirements shown in Table F–1.

Table F–1: Test Equipment Requirements

Item Description	Minimum Requirements	Recommended Instrument
Oscilloscope	Bandwidth: 350 MHz Two channels[1]	Tektronix TDS500 Series Digitizing Oscilloscope or 2400 Series Digitizing Oscilloscope
Function Generator	Output voltage: –5 V to +5 V	Tektronix FG 5010 Programmable Function Generator[2]
Frequency Counter	Frequency range: 10 Hz to 250 MHz Accuracy: ±10 ppm	Tektronix DC 5010 Programmable Universal Counter/Timer[2]
Digital Multimeter (DMM)	DC volts range: 0.05 V to +5 V Accuracy: ±0.1%	Fluke 8842A
TM 5000 Series Power Module	6 plug-in slots	Tektronix TM 5006A Power Module
VXIbus Mainframe	C-size	Tektronix VX1410
VXIbus Resource Manager	C-size slot 0 resource manager	Tektronix VX4521
Personal Computer	IBM PC/AT compatible	IBM PC, PC XT, PC AT, or PC-compatible computer

1 **Four channels are recommended when performing the Synchronous Operation Checks.**

2 **Requires a TM 5000 Series Power Module.**

Table F–1: Test Equipment Requirements (Cont.)

Item Description	Minimum Requirements	Recommended Instrument
GPIB Interface		National Instruments GPIB-PC2A
GPIB Cable		Tektronix Part Number 012-0630-06
Performance Check Disk		Tektronix Part Number 063-1766-XX
Precision Termination	Impedance: 50 Ω, 0.1% Connectors: BNC	Tektronix Part Number 011-0129-00
Adapter	Connectors: BNC female-to-female	Tektronix Part Number 103-0028-00
Adapter	Connectors: BNC female-to-dual banana	Tektronix Part Number 103-0090-00
BNC Dual Input (TEE) Adapter	Connectors: BNC	Tektronix Part Number 103-0030-00
BNC Cable (4 Required)	Impedance: 50 Ω Connectors: BNC Length: 43 inches	Tektronix Part Number 012-0057-01

Performance Check Disk Waveform Files

The VX4792 Performance Check Disk supplied with your waveform generator contains files that are used during the *Performance Verification* procedure. Table F–2 provides a detailed listing of the contents of each waveform file (.PAT suffix). Waveform files define the shape, points, clock frequency, and amplitude of test waveforms. Table F–3 provides a detailed listing of the contents of each command file (.CMD suffix). Command files are ASCII files that define instrument settings.

Table F–2: VX4792 Performance Check Disk Waveform File Summary

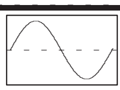
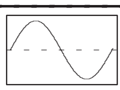
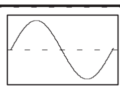
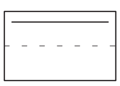
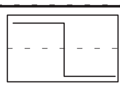
File Name	Wave Shape	Points	Clock	Amplitude	Where Used
AST_1.PAT		1000	250 MHz	3 V	Autostep Mode
AST_2.PAT		200	150 MHz	1.5 V	Autostep Mode
AST_3.PAT		200	25 MHz	0.5 V	Autostep Mode
C1000.PAT		1000	100 MHz	1 V	Clock Amplitude DC Amplitude Accuracy External AM Operation
CLK64.PAT		64	64 MHz	1 V	Pulse Response Synchronous Operation

Table F-2: VX4792 Performance Check Disk Waveform File Summary (Cont.)

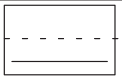
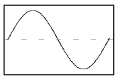
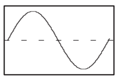
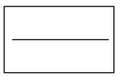
File Name	Wave Shape	Points	Clock	Amplitude	Where Used
MC1000.PAT		1000	250 MHz	1 V	DC Amplitude Accuracy
SIN1000.PAT		1000	100 MHz	1 V	Burst Mode Clock Frequency Accuracy Cont Mode External Trigger Level Accuracy Gate Mode Triggered Mode Waveform Advance Mode
SIN200.PAT		200	100 MHz	1 V	Waveform Advance Mode
Z1000.PAT		1000	250 MHz	1 V	Offset Accuracy

Table F-3: VX4792 Performance Check Disk Command File Summary

File Name	Contents	Where Used

AM.CMD	`*rst`
	`wave "c1000.wfm"`
	`ampl 5v`
	`offset 0v`
	`output:state on`
	`operation eam`
		External AM Operation

AUTO1.CMD	`*rst`
	`autostep:define "mode_ast.ast" ,#0ast_1.wfm`
	`ast_2.wfm`
	`ast_3.wfm`
		Autostep Mode

AUTO2.CMD	`mode astep, "mode_ast.ast"`
	`output:sync start`
	`output:state on`
		Autostep Mode

BURST.CMD	`*rst`
	`mode burst,3`
	`wave "sin1000.wfm"`
	`output:state on`
		Burst Mode

CLKAMP.CMD	`*rst`
	`wave "c1000.wfm"`
	`output:state on`
	`clock:freq 1mhz`
		Clock Amplitude

CONT.CMD	`wave "sine1000.wfm"` `output:state on`	Cont Mode

Table F-3: VX4792 Performance Check Disk Command File Summary (Cont.)

File Name	Contents	Where Used

EXT_CLK.CMD	`*rst`
	`wave "sin1000.wfm"`
	`clock:source ext`
	`output:state on`
		External CLOCK INPUT

FREQ.CMD	`*rst`
	`wave "sin1000.wfm"`
	`clock:freq 250mhz`
	`output:state on`
		Clock Frequency Accuracy

GAIN1.CMD	`*rst`
	`wave "c1000.wfm"`
	`ampl 500mv`
	`output:state on`
		DC Amplitude Accuracy

GAIN2.CMD	`*rst`
	`wave "mc1000.wfm"`
	`ampl 500mv`
	`output:state on`
		DC Amplitude Accuracy

GAIN3.CMD	`*rst`
	`wave "c1000.wfm"`
	`ampl 5v`
	`output:state on`
		DC Amplitude Accuracy

GAIN4.CMD	`*rst`
	`wave "mc1000.wfm"`
	`ampl 5v`
	`output:state on`
		DC Amplitude Accuracy

GATE.CMD	`*rst`
	`mode gate`
	`trig:level 1v`
	`trig:pol pos`
	`trig:imp high`
	`wave "sin1000.wfm"`
	`output:state on`
		Gate Mode

MARK.CMD	`*rst`
	`wave "clk64.wfm"`
	`clock:freq 1mhz`
	`marker1:point 0,on`
	`marker2:point 0,on`
	`output:state on`
		SYNC and MARKER OUTPUT Amplitude

MASTER.CMD	`wave "clk64.wfm"`
	`clock:source int`
	`clock:freq 250mhz`
	`trig:output ecltrg0`
	`trig:input ecltrg0`
	`mode wadv`
	`output:state on`
		Synchronous Operation

Table F-3: VX4792 Performance Check Disk Command File Summary (Cont.)

File Name	Contents	Where Used

OFFSET1.CMD	`*rst`
	`wave "z1000.wfm"`
	`ampl 50mv`
	`offset 1.25v`
	`clock:freq 1mhz`
	`output:state on`
		Offset Accuracy

OFFSET2.CMD	`*rst`
	`wave "z1000.wfm"`
	`ampl 50mv`
	`offset -1.25v`
	`clock:freq 1mhz`
	`output:state on`
		Offset Accuracy

PULSE.CMD	`*rst`
	`wave "clk64.wfm"`
	`ampl 0.5`
	`output:state on`
		Pulse Response

SLAVE.CMD	`wave "clk64.wfm"`
	`clock:source lbus`
	`clock:freq 250mhz`
	`trig:input ecltrg0`
	`trig:output off`
	`output:state on`
	`mode wadv`
		Synchronous Operation

TRIG.CMD	`*rst`
	`mode triggered`
	`trig:pol pos`
	`trig:level 1v`
	`trig:imp low`
	`wave "sin1000.wfm"`
	`output:state on`
		Triggered Mode

TRG_LEV.CMD	`*rst`
	`mode gate`
	`trig:pol pos`
	`trig:level 1v`
	`trig:imp high`
	`wave "sin1000.wfm"`
	`clock:freq 100mhz`
	`output:state on`
		External Trigger Level Accuracy

TRG_POL.CMD	`trig:pol neg` `trig:level -1v`	External Trigger Level Accuracy

Table F–3: VX4792 Performance Check Disk Command File Summary (Cont.)

File Name	Contents	Where Used

WADV1.CMD	`*rst`
	`mode wadv`
	`trig:slop pos`
	`trig:level 1v`
	`trig:imp low`
	`sequence:define "mode_adv.seq",#0sin1000.wfm,10`
	`sin200.wfm,10`
		Waveform Advance Mode Mode

WADV2.CMD	`wave "mode_adv.seq"` `output:state on`	Waveform Advance Mode

System Setup

Before you perform the performance verification procedures, you must complete the following tasks to configure your system. Figure F–2 shows the basic system configuration.

1. Install the GPIB Interface circuit board and software into the IBM PC-compatible computer (refer to the GPIB Interface user documentation for instructions).
2. Configure the VXIbus mainframe for the instrument modules to be installed. See page 1–6 for Tektronix mainframes; otherwise refer to the user manual for the mainframe for instructions.
3. Setup the Slot 0 Resource Manager for operation within this system (Figure F–1):
   - Set the logical address to 1
   - Set the configuration switch for secondary addressing (Switch #1 CLOSED)
   - Set the CLK 10 switch for internal operation
   - Set the interrupt handler switch to 7

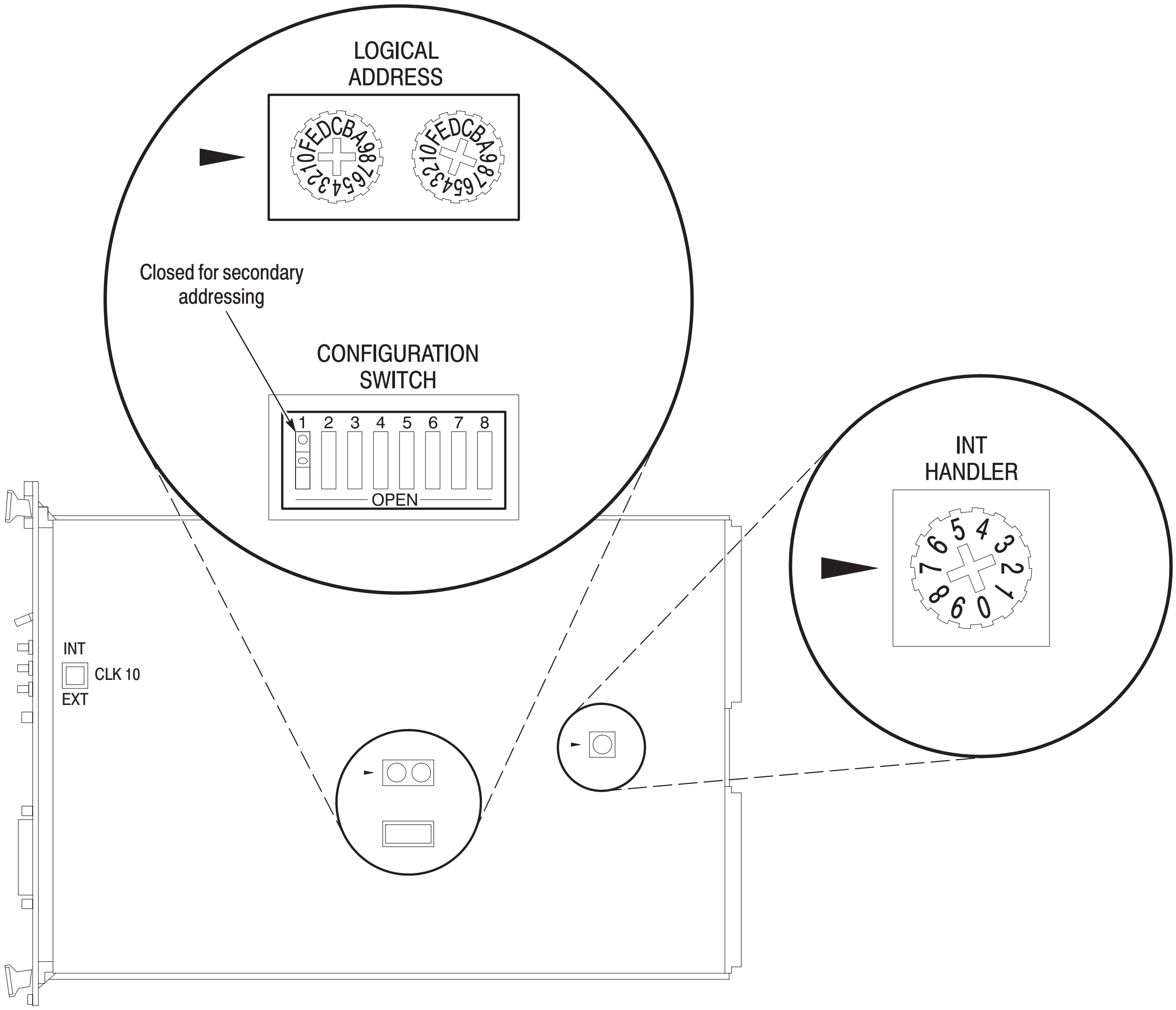


Figure F-1: Setting Switches on the Resource Manager

4. Install the Slot 0 Resource Manager into Slot 0 of the mainframe (refer to the resource manager user manual for instructions).
5. Install the VX4792 Arbitrary Waveform Generator into the slot next to the Slot 0 Resource Manager (see page 1–3 for instructions).
6. Install the GPIB cable between the computer and Slot 0 Resource Manager. Your system should match the configuration shown in Figure F–2.

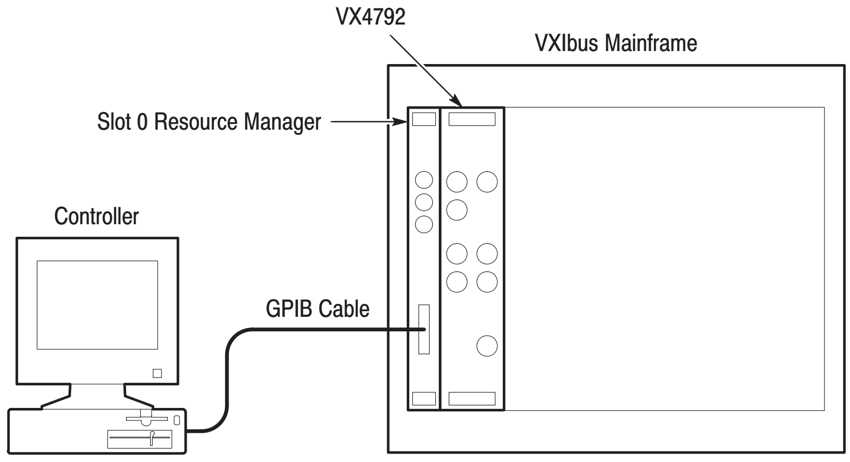


Figure F-2: Basic System Configuration

7. Load the Performance Check Disk onto the hard disk of the computer at the directory that contains IBIC.

8. On the computer, change the directory to the directory where IBCONF is located.

9. To invoke IBCONF, type:

   ```
   IBCONF <RETURN>
   ```

   Change the Device Characteristics Parameters for DEV1 as shown in Table F–4.

Table F-4: VX4792 Device Characteristics Parameters

Parameter	Setting for VX4792 (DEV1)
Primary GPIB Address	1
Secondary GPIB Address	97
Timeout Setting	T10s
EOS Byte	0AH
Terminate Read on EOS	NO
Set EOI with EOS on Write	NO
Type of Compare on EOS	7-Bit
Set EOI with Last Byte of Write	YES

10. Reboot your computer if necessary.

11. On the computer, change the directory to access IBIC.

12. To invoke IBIC and select device 1 (the VX4792 Arbitrary Waveform Generator), type the following commands:

```
IBIC <RETURN>
```

```
IBFIND DEV1 <RETURN>
```

Now your computer will display the `DEV1:` prompt.

13. Perform the *Functional Check Procedure* located on page 1–11. This procedure verifies that your system is operational, and runs a self-calibration to optimize waveform generator performance.

Operating Mode Checks

This procedure checks the operation of the Continuous, Triggered, Burst, Gated, Waveform Advance, and Autostep modes.

Check Continuous Mode

This procedure checks the operation of the Continuous mode.

Electrical Characteristic Checked: Operating mode, Continuous, on page E–4.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install the test setup and set test equipment controls (see Figure F–3):*

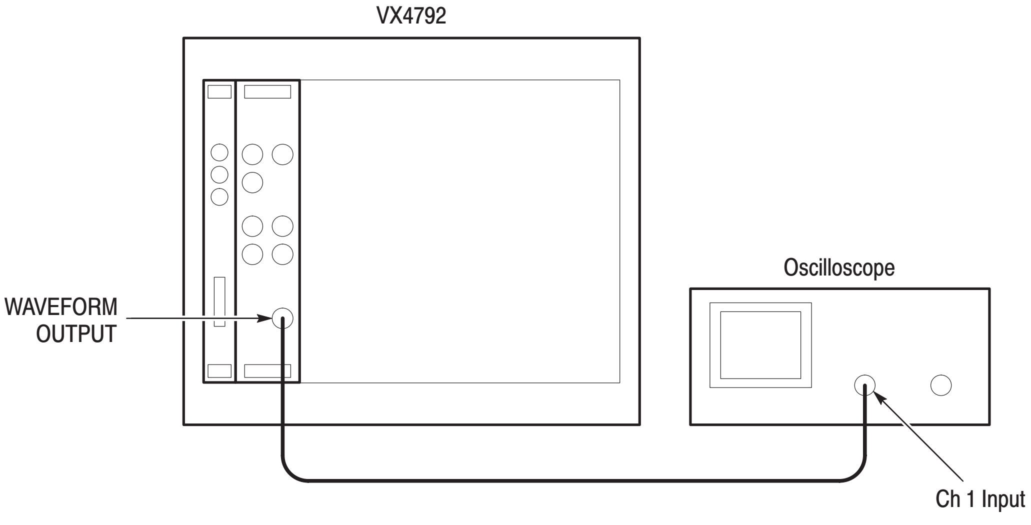


Figure F–3: Cont Mode Initial Test Setup

a. *Connect the oscilloscope:* Connect the WAVEFORM output connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

b. *Set the oscilloscope controls:*

Vertical	CH1
CH1 coupling	DC
CH1 scale	0.2 V/div
CH1 input impedance:	50 Ω
Horizontal	
Sweep	5 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	–100 mV
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRT "*RST" <RETURN>

IBWRTF SIN1000.PAT <RETURN>

IBWRTF CONT.CMD <RETURN>
```

3. *Check against limits*: Check that the oscilloscope displays a five-cycle sine wave that has an amplitude of five divisions.

4. *End procedure:* Disconnect the oscilloscope.

Check Triggered Mode

This procedure checks the operation of the Triggered mode.

Electrical Characteristic Checked: Operating mode, Triggered, on page E–4.

Equipment Required: Two 50 Ω coaxial cables, a function generator, and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install the test setup and set test equipment controls (see Figure F–4):*

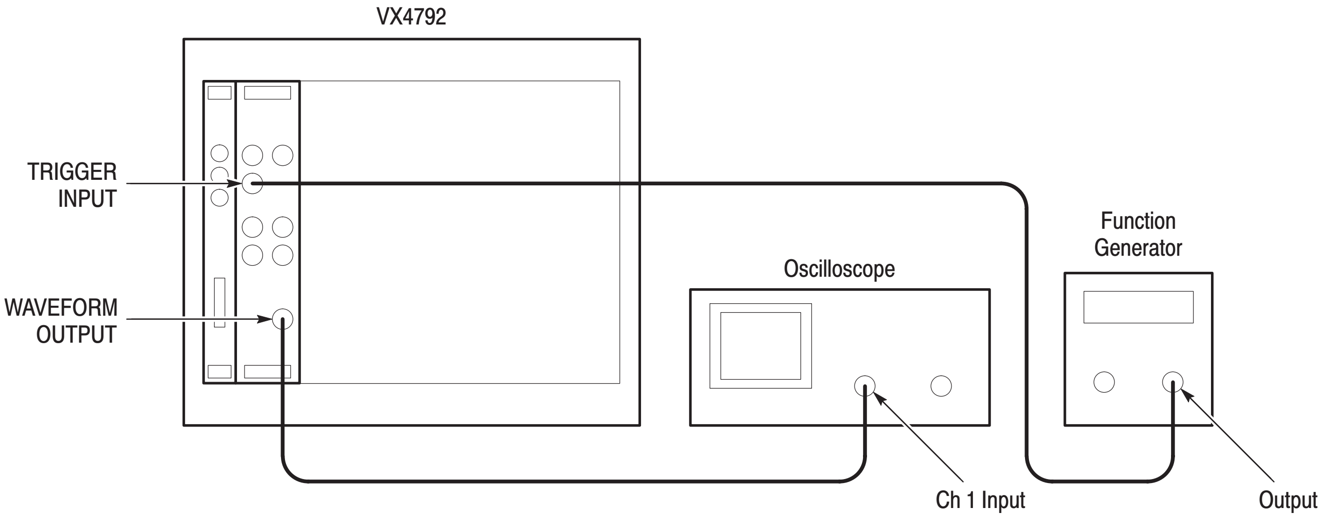


Figure F-4: Triggered Mode Initial Test Setup

a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

b. *Connect the function generator:* Connect the TRIGGER INPUT connector though a coaxial cable to the function generator output connector.

c. *Set the oscilloscope controls:*

Vertical:	CH1
CH1 coupling:	DC
CH1 scale	0.2 V/div
CH1 input impedance:	50 Ω
Horizontal	
Sweep	10 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	–100 mV
Mode	Auto

d. *Set the function generator controls:*

Function	Square
Mode	Continuous
Parameter	
Frequency	100 Hz
Amplitude	4 V
Offset	2 V
Output	Off

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF TRIG.CMD <RETURN>
```

***NOTE**. Watch the CRT of the oscilloscope when triggering manually. The oscilloscope sweeps one time for each* START *command.*

3. *To check triggered mode with manual triggering, type:*

```
IBWRT "START" <RETURN>
```

Check that the oscilloscope displays a one-cycle sine wave.

4. *Check triggered mode with external triggering:*

a. *Enable function generator output:* Turn on the function generator output.

b. *Check triggering:* Check that for each trigger supplied by the function generator, the oscilloscope displays a one-cycle sine wave.

5. *End procedure:* Turn off the function generator output, and disconnect the function generator and oscilloscope.

Check Burst Mode

This procedure checks the operation of the Burst mode.

Electrical Characteristic Checked: Operating mode, Burst, on page E–4.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–5):*

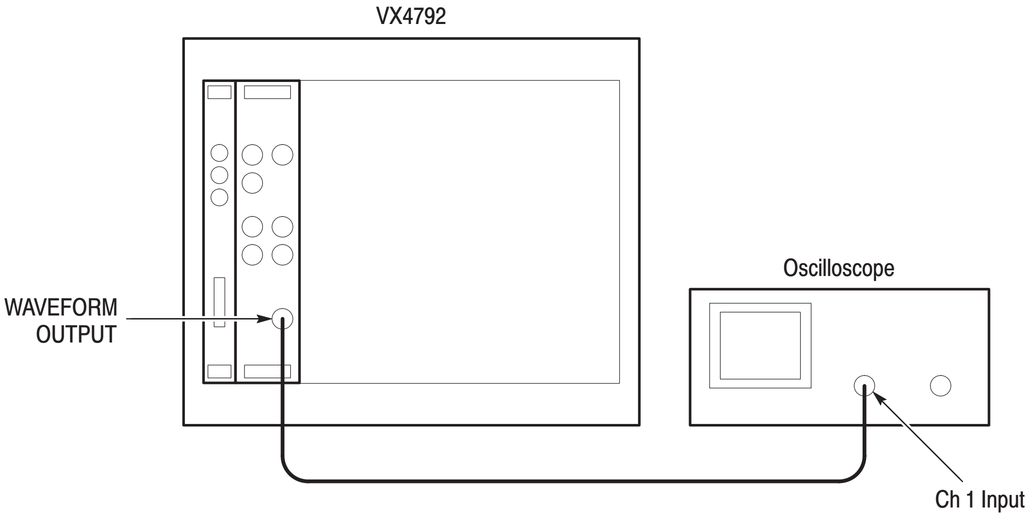


Figure F–5: Burst Mode Initial Test Setup

a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

b. *Set the oscilloscope controls:*

Vertical:	CH1
CH1 coupling:	DC
CH1 scale	0.2 V/div
CH1 input impedance:	50 Ω
Horizontal	
Sweep	10 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	–100 mV
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF BURST.CMD <RETURN>
```

***NOTE**. Watch the CRT of the oscilloscope when triggering manually. The oscilloscope sweeps one time for each* START *command.*

3. *To check burst count, type*:

```
IBWRT "START" <RETURN>
```

Check that the oscilloscope displays three cycles of the sine wave.

4. *End procedure:* Disconnect the oscilloscope.

Check Gated Mode

This procedure checks the operation of the Gated mode.

Electrical Characteristic Checked: Operating mode, Gated, on page E–4.

Equipment Required: Four 50 Ω coaxial cables, a 50 Ω precision termination, a BNC female-to-female adapter, a dual input adapter (TEE), a function generator, and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–6):*

   a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

   b. *Connect the function generator:* Connect the function generator output to both the TRIGGER INPUT and the oscilloscope CH2 input through coaxial cables, a precision termination, and a dual input adapter.

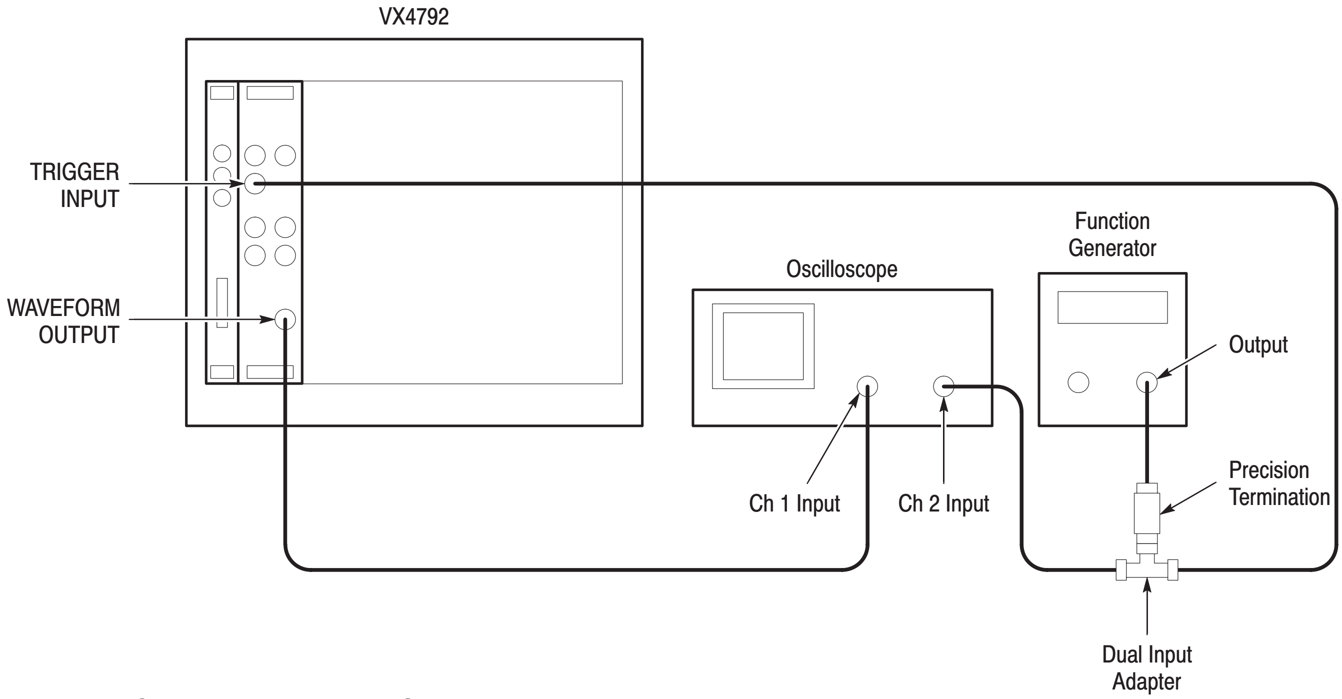


Figure F-6: Gated Mode Initial Test Setup

c. *Set oscilloscope controls:*

Vertical	CH1, CH2
CH1, CH2 coupling	DC
CH1 scale	0.5 V/div
CH2 scale	1 V/div
CH1 input impedance	50 Ω
CH2 input impedance	1 MΩ
Horizontal	
Sweep	20 μs/div
Trigger	
Source	CH2
Coupling	DC
Slope	Positive
Level	500 mV
Mode	Auto

d. *Set function generator controls:*

Function	Square
Mode	Continuous
Parameter	
Frequency	10 kHz
Amplitude	4.0 V
Offset	2.0 V
Output	Off

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

`IBWRTF GATE.CMD <RETURN>`

3. *To check gated mode with manual trigger:*

a. Type:

`IBWRT "START" <RETURN>`

Check that the oscilloscope continuously displays a sine wave.

b. Type:

`IBWRT "STOP" <RETURN>`

Check that the oscilloscope continuously displays a line.

4. *Check gated mode with gate signal:*

a. *Apply gate signal:* Turn function generator output on.

b. *Check gated mode with positive gate signal:* Check that the oscilloscope displays a sine wave when the gate signal is high (Figure F–7).

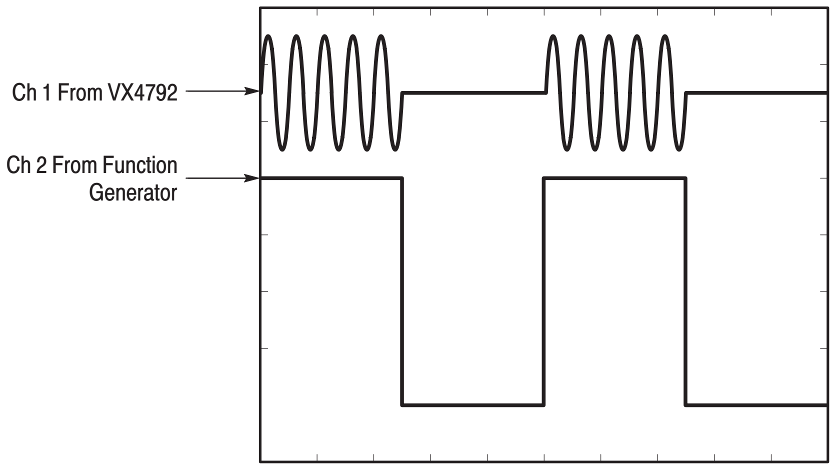


Figure F–7: Waveform Output Signal with Positive Gate Signal

c. *To change the trigger polarity to negative, type:*

`IBWRT "TRIG:POL NEG" <RETURN>`

d. *Check gated mode with a negative gate signal:* Check that the oscilloscope displays a sine wave while the gate signal is low (Figure F–8).

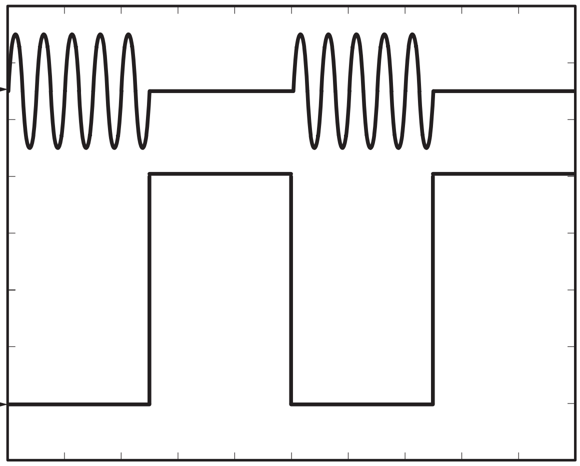


Figure F–8: Waveform Output Signal with Negative Gate Signal

5. *End procedure:* Turn the function generator output off and disconnect the function generator.

Check Waveform Advance Mode

This procedure checks the operation of Waveform Advance mode.

Electrical Characteristic Checked: Operating mode, Waveform Advance, on page E–4.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–9):*

   a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

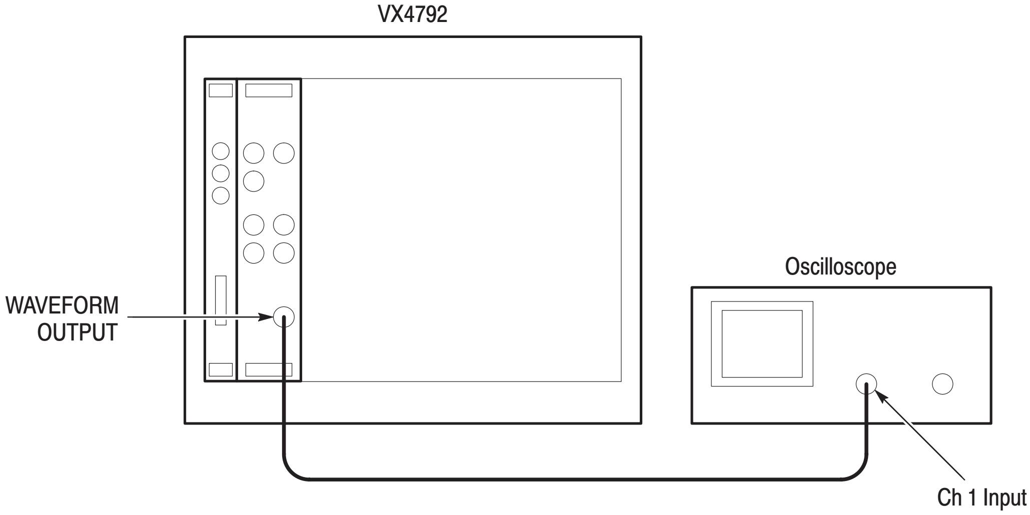


Figure F-9: Waveform Advance Mode Initial Test Setup

b. *Set oscilloscope controls:*

Vertical	CH1
CH1 coupling	DC
CH1	0.2 V/div
CH1 input impedance	50 Ω
Horizontal	
Sweep	5 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	0 V
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF SIN200.PAT <RETURN>
```

```
IBWRTF WADV1.CMD <RETURN>
```

```
IBWRTF WADV2.CMD <RETURN>
```

3. *To check waveform advance, repeatedly type:*

```
IBWRT "START" <RETURN>
```

Check that the oscilloscope displays a continuous sine wave that switches between two frequencies each time you send the `START` command.

4. *End procedure:* Disconnect the oscilloscope.

Check Autostep Mode

This procedure checks the operation of the Autostep mode.

Electrical Characteristic Checked: Operating mode, Autostep, on page E–4.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–10):*

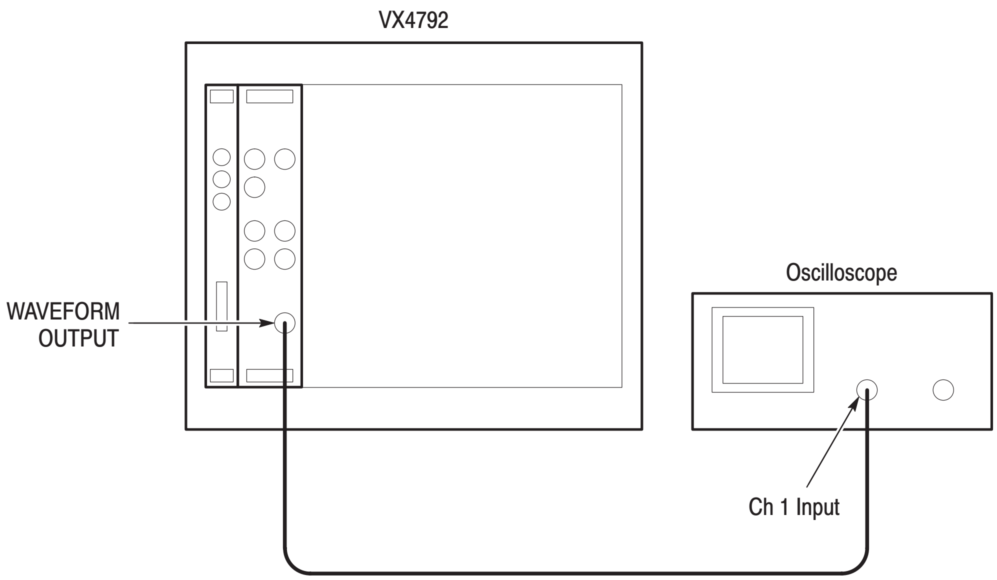


Figure F–10: Autostep Mode Initial Test Setup

a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT through a coaxial cable to the oscilloscope CH1 vertical input.

b. *Set the oscilloscope controls:*

Vertical	CH1
CH1 coupling	DC
CH1 scale	0.5 V/div
CH1 input impedance	50 Ω
Horizontal	
Sweep	2 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	100 mV
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   ```
   IBWRTF AST_1.PAT <RETURN>
   IBWRTF AST_2.PAT <RETURN>
   IBWRTF AST_3.PAT <RETURN>
   IBWRTF AUTO1.CMD <RETURN>
   IBWRTF AUTO2.CMD <RETURN>
   ```

***NOTE**. Watch the CRT of the oscilloscope when triggering manually. The oscilloscope sweeps one time for each* `START` *command.*

3. *To check autostep mode, repeatedly type:*

   ```
   IBWRT "START" <RETURN>
   ```

   Check that the oscilloscope momentarily displays a sine wave that switches frequency and amplitude each time you enter the `START` command.

4. *End procedure:* Disconnect the oscilloscope.

External AM Operation Checks

This procedure checks operation of external AM mode.

Electrical Characteristic Checked: Auxiliary Inputs, AM, page E–6.

Equipment Required: Two 50 Ω coaxial cables, a 50 Ω terminator, a function generator, and a digital multimeter (DMM).

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Set function generator controls:*

Function	Square
Mode	Continuous
Parameter	
Frequency	1 kHz
Amplitude	0 V
Offset	1 V
Output	On

2. *Set DMM controls*:

Mode	VDC
Range	Auto
Inputs	Front

3. *Adjust function generator offset for +1.00 V:*

   a. Connect the function generator output to the DMM.

   b. Adjust the function generator offset for a +1.00 V DMM display.

   c. Disconnect the DMM from the function generator output.

4. *Install test setup (see Figure F–11):*

   a. *Connect DMM:* Connect the WAVEFORM OUTPUT through a coaxial cable, the 50 Ω terminator, and BNC-to-dual banana adapter to the DMM INPUT connector.

   b. *Connect function generator:* Connect the AM INPUT through a coaxial cable to the function generator output.

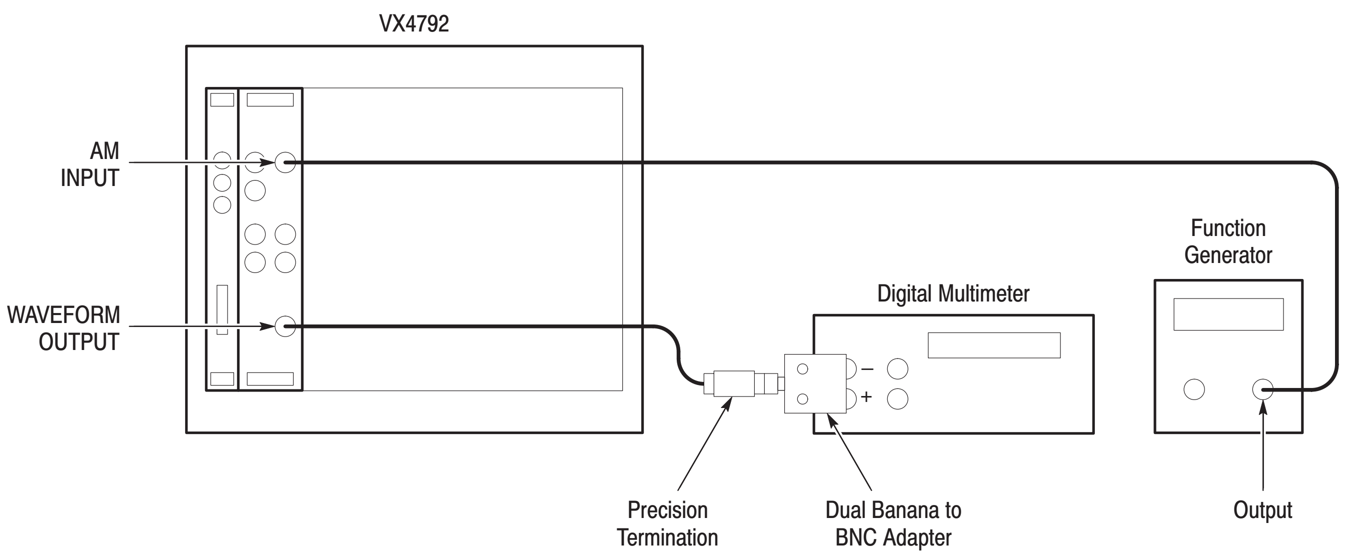


Figure F–11: External AM Operation Initial Test Setup

5. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF C1000.PAT <RETURN>

IBWRTF AM.CMD <RETURN>
```

6. *Check external AM operation with +1 V AM input:* Check that the DMM reading is between 2.375 V and 2.625 V (100% modulation).

7. *Adjust function generator offset for 0 V:*

   a. Disconnect the 50 Ω terminator from the DMM.

   b. Disconnect the coaxial cable from the function generator output.

   c. Connect the function generator output to the DMM.

   d. Adjust the function generator offset for a 0.00 V DMM display.

   e. Disconnect the DMM from the function generator output.

   f. Install the test setup (see Figure F–11 and step 4).

8. *Check external AM operation with 0 V AM input:* Check that the DMM reading is between 1.125 V and 1.375 V (50% modulation).

9. *Adjust function generator offset for –1.00 V:*

   a. Disconnect the 50 Ω terminator from the DMM.

   b. Disconnect the coaxial cable from the function generator output.

   c. Connect the function generator output to the DMM.

   d. Adjust the function generator offset for a –1.00 V DMM display.

   e. Disconnect the DMM from the function generator output.

   f. Install the test setup (see Figure F–11 and step 4).

10. *Check external AM operation with –1 V AM input:* Check that the DMM voltage reading is between –0.125 V and 0.125 V (0% modulation).

11. *End procedure:* Turn the function generator output off. Disconnect the function generator and DMM.

Clock Frequency and Amplitude Checks

These procedures check the accuracy of the clock frequency and the clock output amplitude.

Check Clock Frequency Accuracy

This procedure checks the CLOCK OUTPUT signal frequency accuracy.

Electrical Characteristic Checked: Clock Generator, Accuracy, on page E–2.

Equipment Required: A 50 Ω coaxial cable and a frequency counter.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–12):*

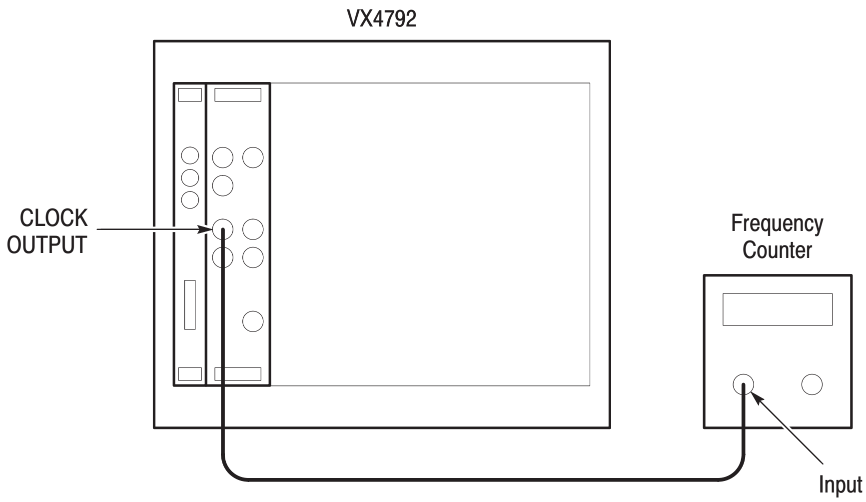


Figure F–12: Clock Frequency Accuracy Initial Test Setup

a. *Connect frequency counter:* Connect the CLOCK OUTPUT to the frequency counter input through a coaxial cable.

b. *Set frequency counter controls:*

CHANNEL A	
Termination	50 Ω
Slope	Negative
Attenuation	X5
Coupling	DC
FREQ A	

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   `IBWRTF FREQ.CMD <RETURN>`

3. *Check clock frequency accuracy:*

   a. *Check clock frequency accuracy at current clock frequency setting:* Check that the frequency counter reading is between 249.9875 MHz and 250.0125 MHz.

   b. *To check clock frequency accuracy for different clock frequency settings:*

      - Type:

        `IBWRT "CLOCK:FREQ <FREQ FROM TABLE>" <RETURN>`

        where `<FREQ FROM TABLE>` is a frequency command from Table F–5.

      - Change the clock frequency to each value shown in Table F–5. Check that the frequency counter reading is between the test limits shown for each clock frequency.

Table F–5: Clock Frequency Settings

Frequency Command	Clock Frequency	Test Limits
100MHz	100 MHz	99.995 MHz to 100.005 MHz
1MHz	1 MHz	0.99995 MHz to 1.00005 MHz
1kHz	1 kHz	0.99995 kHz to 1.00005 kHz
10Hz	10 Hz	9.9995 Hz to 10.0005 Hz

4. *End procedure:* Disconnect the frequency counter.

Check Clock Amplitude

This procedure checks the CLOCK OUTPUT signal amplitude.

Electrical Characteristic Checked: Auxiliary Outputs, Clock, Amplitude, on page E–6.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–13):*

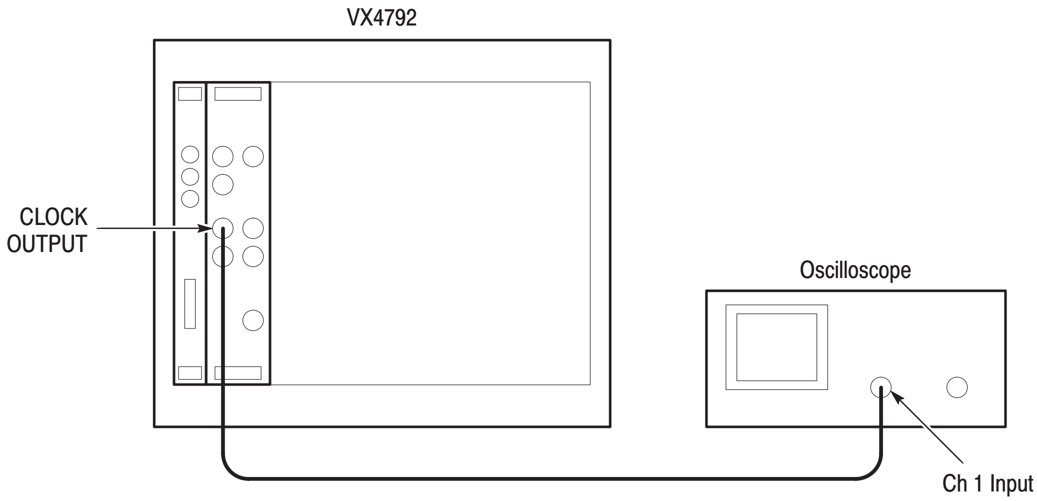


Figure F–13: Clock Amplitude Initial Test Setup

a. *Connect oscilloscope:* Connect the CLOCK OUTPUT through a coaxial cable to the oscilloscope CH1 vertical input.

b. *Set oscilloscope controls:*

Vertical	CH1
Coupling	DC
Scale	200 mV/div
Input impedance	50 Ω
Horizontal	
Sweep	500 ns/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	500 mV
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF CLKAMP.CMD <RETURN>
```

3. *Check clock amplitude accuracy:* Check that the pulse displayed on the oscilloscope has an amplitude between 0.7 V and 1.2 V_{p-p}.

4. *End procedure:* Disconnect the oscilloscope.

Amplitude and Offset Accuracy Checks

These procedures check the accuracy of the output DC amplitude and offset parameters.

NOTE. *The amplitude and offset accuracy checks are structured as a continuous test. After completing the* Check DC Amplitude Accuracy *test, you will use the current control settings, and the next step in the sequence file, to perform the* Check Offset Accuracy *test.*

Check DC Amplitude Accuracy

This procedure checks the DC amplitude accuracy.

Electrical Characteristic Checked: Main Outputs, Offset, DC Accuracy, on page E–3.

Equipment Required: A 50 Ω coaxial cable, a BNC-to-dual banana adapter, and a digital multimeter (DMM).

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set controls (see Figure F–14):*

   a. *Connect DMM:* Connect the WAVEFORM OUTPUT through a 50 Ω coaxial cable, a 50 Ω terminator, and a BNC-to-dual banana adapter to the DMM INPUT connector.

   b. *Set DMM controls:*

Mode	VDC
Range	Auto
Input	Front

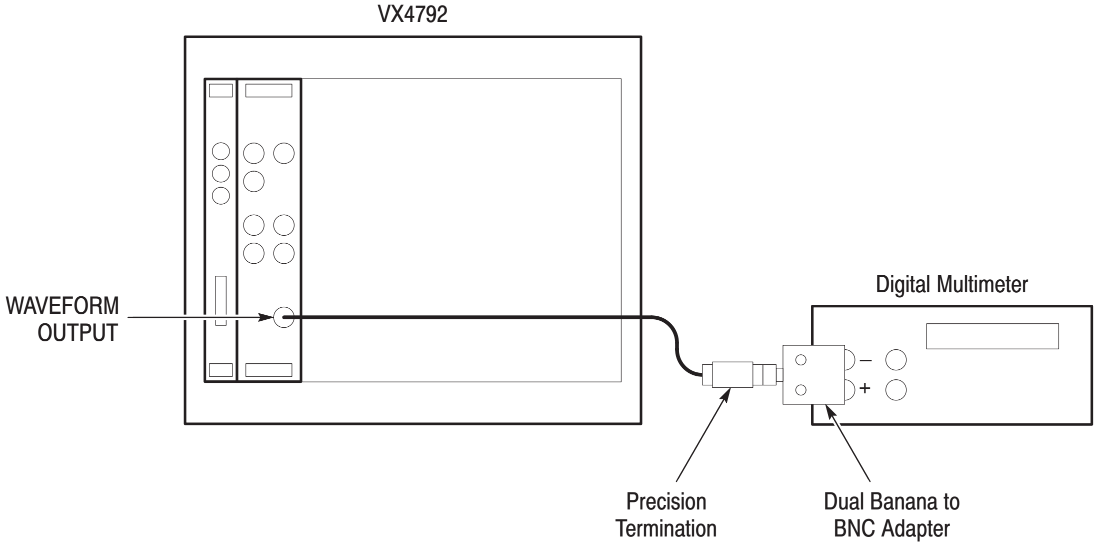


Figure F-14: Gain Accuracy Initial Test Setup

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   ```
   IBWRTF MC1000.PAT <RETURN>
   ```

   ```
   IBWRTF GAIN1.CMD <RETURN>
   ```

3. *Check gain accuracy:* Check the DMM for between +0.24375 V and +0.25625 V.
4. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   ```
   IBWRTF GAIN2.CMD <RETURN>
   ```

5. *Check gain accuracy:* Check the DMM for between –0.24375 V and –0.25625 V.
6. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   ```
   IBWRTF GAIN3.CMD <RETURN>
   ```

7. *Check gain accuracy:* Check the DMM for between +2.45 V and +2.55 V.
8. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

   ```
   IBWRTF GAIN4.CMD <RETURN>
   ```

9. *Check gain accuracy:* Check the DMM for between –2.45 V and –2.55 V.
10. *End procedure:* Retain the test setup and control settings.

Check Offset Accuracy

This procedure checks the output offset accuracy.

Electrical Characteristic Checked: Main Outputs, Offset, Accuracy, on page E–3.

Equipment Required: A 50 Ω coaxial cable, 50 Ω termination, BNC-to-dual banana adapter, and a digital multimeter (DMM).

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Use the test setup and test equipment settings from previous check.*

2. *To check offset accuracy:*

  a. Type:

```
IBWRTF Z1000.PAT <RETURN>
```

```
IBWRTF OFFSET1.CMD <RETURN>
```

  b. Check that the DMM voltage reading is between +1.2325 V and +1.2675 V.

  c. Type:

```
IBWRTF OFFSET2.CMD <RETURN>
```

  d. Check that the DMM voltage reading is between –1.2675 V and –1.2325 V.

3. *End procedure:* Disconnect the DMM and the termination.

Pulse Response Check

This procedure checks the pulse response characteristics of the VX4792 Arbitrary Waveform Generator output waveforms at amplitudes of 0.5 V and 1 V.

Electrical Characteristic Checked: Main Outputs, Pulse Response, on page E–3.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–15):*

  a. *Connect the oscilloscope:* Connect the WAVEFORM OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

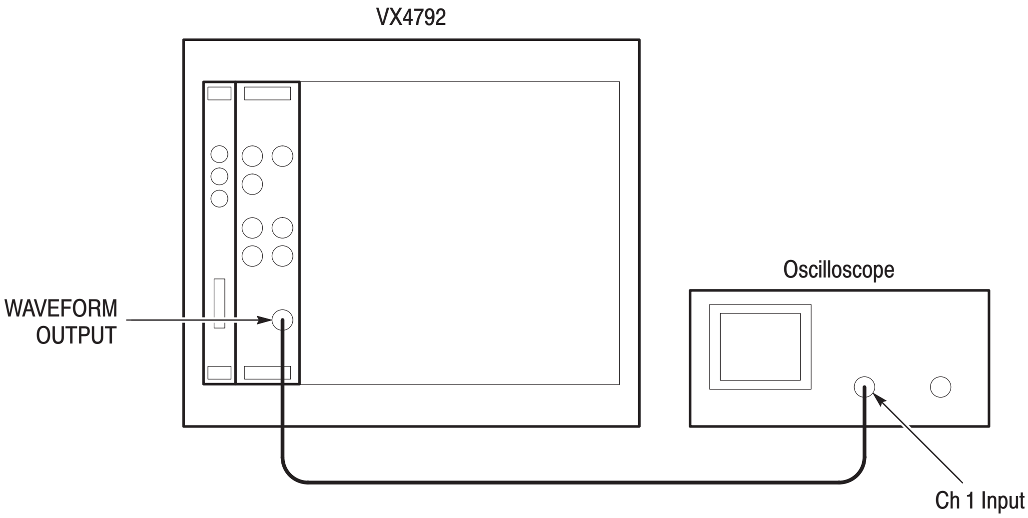


Figure F–15: Pulse Response Initial Test Setup

b. *Set oscilloscope controls:*

Vertical	CH1
Coupling	DC
Scale	0.1 V/div
Input impedance	50 Ω
Horizontal	
Sweep	2 ns/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	0 V
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF CLK64.PAT <RETURN>
```

```
IBWRTF PULSE.CMD <RETURN>
```

3. *Check pulse response at 0.5 V amplitude:*

a. *Check rise time:* Check that the rise time of the waveform displayed on the oscilloscope from the 10% point to the 90% point is 4 ns or less.

b. *Check aberrations:* Check that the aberrations of the displayed waveform are within 0.45 div.

c. *Check flatness:* Set the oscilloscope sweep for 5 ns/div. Check that the flatness of the displayed waveform is within 0.15 div after 20 ns from the rising edge.

d. *Change the oscilloscope controls:*

Horizontal	
Sweep	2 ns/div
Trigger	
Slope	Negative

e. *Check fall time:* Check that the fall time of the displayed waveform from the 10% point to the 90% point is 4.2 ns or less.

4. *Check pulse response at 1 V amplitude:*

a. *Change the oscilloscope controls:*

Vertical	CH1
CH1 scale	0.2 V/div
Trigger	
Slope	Positive

b. *To change the VX4792 Arbitrary Waveform Generator controls, type:*

```
IBWRT "AMPL 1V" <RETURN>
```

c. *Check the pulse response:* Repeat substeps described in 3a through e, checking the pulse response characteristics shown in Table F–6.

Table F–6: Pulse Response Characteristics for 1 V Amplitude

Characteristics	Test Limits
Rise time	4 ns, maximum
Aberrations	0.4 div, maximum
Flatness	0.15 div, maximum
Fall time	4 ns, maximum

5. *End procedure:* Remove the connections.

SYNC OUTPUT and MARKER OUTPUT Amplitude Checks

These procedures check the amplitude of the SYNC OUTPUT and MARKER OUTPUT signals.

Electrical Characteristic Checked: Auxiliary Outputs, Sync, Amplitude, on page E–5; Auxiliary Outputs, Marker 1, Amplitude, on page E–5.

Equipment Required: A 50 Ω coaxial cable and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–16):*

   a. *Connect the oscilloscope:* Connect the SYNC OUTPUT connector through the coaxial cable to the CH1 vertical input connector on the oscilloscope.

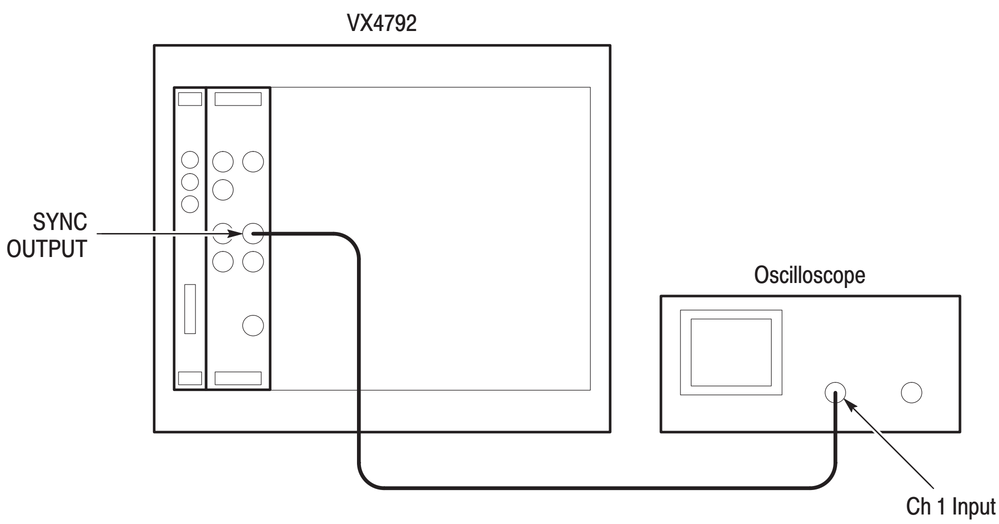


Figure F–16: SYNC OUTPUT and MARKER OUTPUT Amplitude Initial Test Setup

b. *Set oscilloscope controls:*

Vertical	CH1
CH1 Coupling	DC
CH1 Scale	200 mV/div
CH1 Input Impedance	50 Ω
Horizontal	
Sweep	50 ns/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	500 mV
Mode	Auto

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF MARK.CMD <RETURN>
```

3. *Check front-panel SYNC OUTPUT and MARKER 1 OUTPUT amplitude:*

a. *Check SYNC OUTPUT pulse amplitude:* Check that the pulse amplitude of the waveform displayed on the oscilloscope is 1.2 V_{p-p} or greater.

b. *Check MARKER OUTPUT pulse amplitude:*

- Move the coaxial cable from the SYNC OUTPUT connector to the MARKER 1 connector.
- Change the oscilloscope sweep to 1 μs/div.
- Check that the pulse amplitude of the displayed waveform is between 2.25V and 2.625V.

4. *Check MARKER 2 OUTPUT pulse amplitude:*

- Disconnect the coaxial cable from the MARKER 1 connector, and connect the cable to the MARKER 2 OUTPUT connector.
- Check that the pulse amplitude of the displayed waveform is between 2.25V and 2.625V.

5. *End procedure:* Disconnect the oscilloscope.

External Trigger Level Accuracy Check

This procedure checks the external trigger level accuracy of the VX4792 Arbitrary Waveform Generator.

Electrical Characteristic Checked: Auxiliary Inputs, Trigger, Accuracy, on page E–6.

Equipment Required: Two 50 Ω coaxial cables, a function generator, and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–17):*

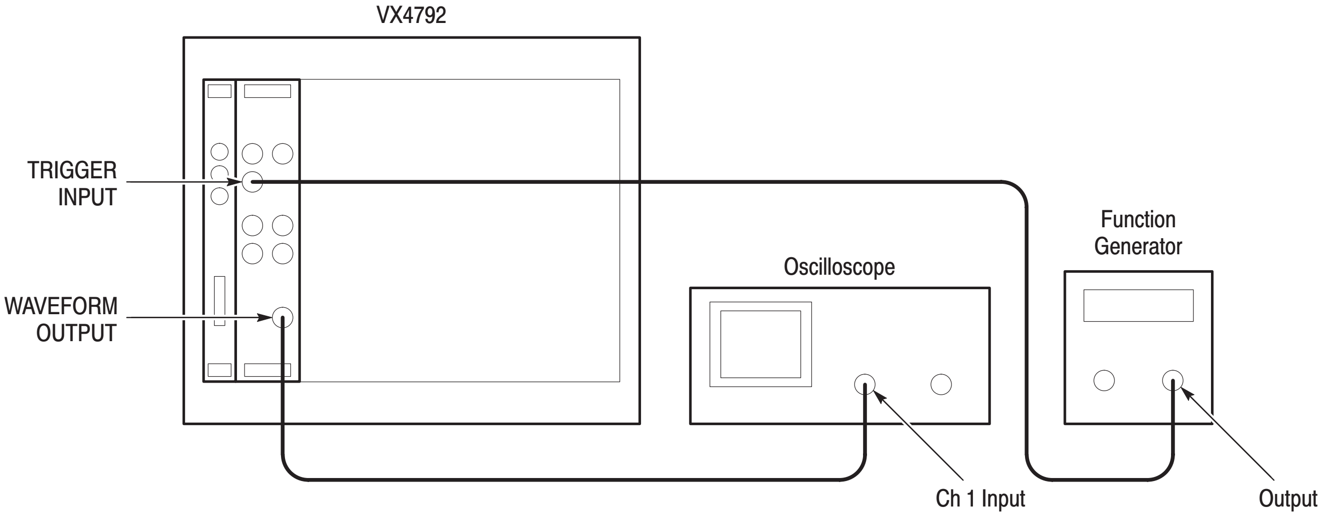


Figure F–17: External Trigger Level Accuracy Initial Test Setup

   a. *Connect oscilloscope:* Connect the WAVEFORM OUTPUT through a coaxial cable to the oscilloscope CH1 vertical input.

   b. *Connect function generator:* Connect the TRIGGER INPUT through a coaxial cable to the function generator output.

   c. *Set oscilloscope controls:*

Vertical	CH1
CH1 Coupling	DC
CH1 Scale	0.2 V/div
CH1 Input Impedance	50 Ω

Horizontal	
Sweep	50 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	0 V
Mode	Auto

d. *Set function generator controls:*

Function	Square
Mode	Continuous
Parameter	
Frequency	1 kHz
Amplitude	0 V
Offset	0.6 V
Output	Off

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF TRG_LEV.CMD <RETURN>
```

3. *Check external trigger high level:*

a. *Enable function generator output:* Turn on the function generator output.

b. *Check external trigger level accuracy:*

- Gradually increase the function generator offset level until a waveform is displayed on the oscilloscope.
- Check that that the function generator offset level is between +0.85 V and +1.15 V, when the waveform is first displayed.

4. *Check external trigger low level:*

a. *Change the function generator controls:*

Parameter	
Offset	–0.6 V

b. *To set the VX4792 Arbitrary Waveform Generator controls, type:*

```
IBWRTF TRG_POL.CMD <RETURN>
```

c. *Check external trigger level accuracy:*

- Gradually decrease the function generator offset level until a waveform is displayed on the oscilloscope.
- Check that that the function generator offset level is between –1.15 V and –0.85 V, when the waveform is first displayed.

5. *End procedure:* Turn off the function generator output and disconnect the function generator.

External CLOCK INPUT Check

This procedure checks the VX4792 Arbitrary Waveform Generator response to an external CLOCK INPUT signal.

Electrical Characteristic Checked: Auxiliary Inputs, Clock, Threshold level, on page E–6.

Equipment Required: Two 50 Ω coaxial cables, a function generator, and an oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install test setup and set test equipment controls (see Figure F–18):*

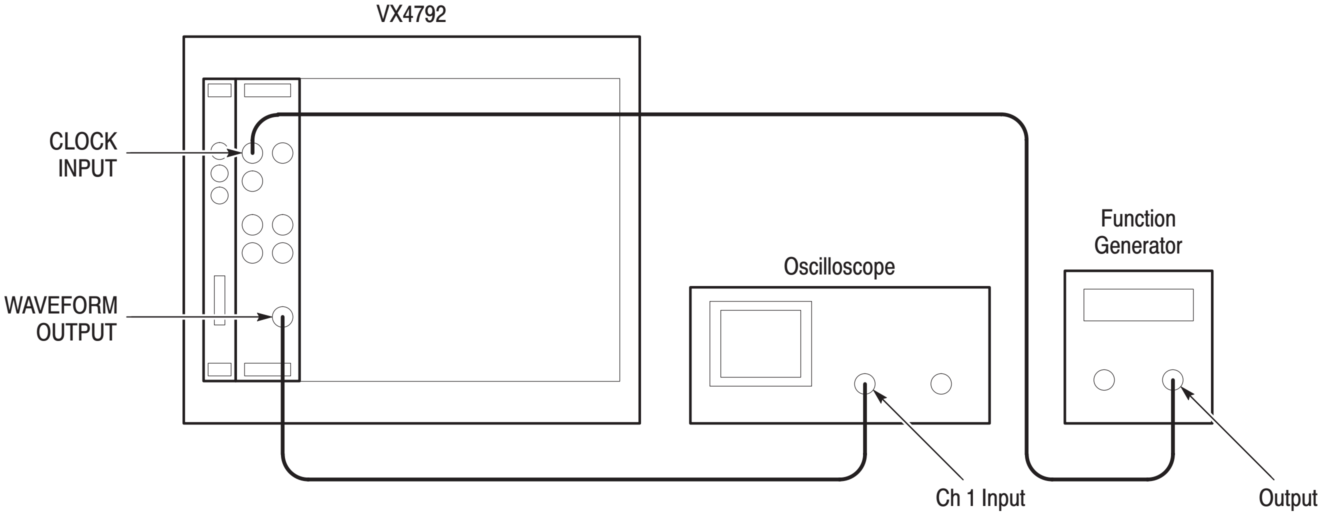


Figure F–18: External CLOCK INPUT Initial Test Setup

a. *Connect oscilloscope:* Connect the WAVEFORM OUTPUT through a coaxial cable to the oscilloscope CH1 vertical input.

b. *Connect function generator:* Connect the CLOCK INPUT through a coaxial cable to the function generator output.

c. *Set oscilloscope controls:*

Vertical	CH1
Coupling	DC
Scale	0.2 V/div
Input Impedance	50 Ω
Horizontal	
Sweep	500 μs/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	0 mV
Mode	Auto

d. *Set function generator controls:*

Function	Square
Mode	Continuous
Parameter	
Frequency	1 MHz
Amplitude	1.6 V
Offset	0.6 V
Output	Off

2. *To set the VX4792 Arbitrary Waveform Generator controls and select the waveform file, type:*

```
IBWRTF EXT_CLK.CMD <RETURN>
```

3. *Check the external CLOCK INPUT threshold level:*

a. *Enable function generator output:* Turn on function generator output.

b. *Check the level:* Check that the waveform displayed on the oscilloscope has an amplitude of five divisions and a stable display of five cycles.

4. *Turn off equipment output and disconnect test setup:*

a. *Disable function generator output:* Turn off function generator output.

b. *Remove connections:* Disconnect all connections to the waveform generator unless you will perform the *Synchronous Operation Check*.

Synchronous Operation Check

***NOTE**. The following procedure can only be performed when two or more VX4792 Arbitrary Waveform Generator modules are installed into the same mainframe. The procedure provides steps for testing six modules. To test fewer than six modules, ignore the setup information for the extra modules.*

To avoid signal delays due to the test cables, use only coaxial cables of the same length and type.

This procedure checks the skew between multiple VX4792 Arbitrary Waveform Generator modules within the same mainframe.

Electrical Characteristic Checked: Main Outputs, Skew Between Modules, Threshold level, on page E–3.

Equipment Required: Four 50 Ω coaxial cables (same length) and a 4-trace oscilloscope.

Prerequisites: The instrument must meet the prerequisites listed on page F–3.

Procedure:

1. *Install VX4792 Arbitrary Waveform Generator modules into mainframe:*

   a. With the VXIbus mainframe turned off, install the additional VX4792 Arbitrary Waveform Generator modules into adjacent slots (see page 1–3 for instructions). For example, if the first instrument module is installed into slots 1 and 2, install the next module into slots 3 and 4.

   b. Turn on the VXIbus mainframe.

2. *Install test setup and set test equipment controls (see Figure F–19):*

   a. *Connect oscilloscope to first VX4792 Arbitrary Waveform Generator module (slots 1 and 2):* Connect the WAVEFORM OUTPUT from the first module (slots 1 and 2) through a coaxial cable to the oscilloscope CH1 vertical input.

   b. *Connect oscilloscope to second VX4792 Arbitrary Waveform Generator module (slots 3 and 4):* Connect the WAVEFORM OUTPUT from the second module (slots 3 and 4) through a coaxial cable to the oscilloscope CH2 vertical input.

   c. *Connect oscilloscope to third VX4792 Arbitrary Waveform Generator module (slots 5 and 6):* Connect the WAVEFORM OUTPUT from the third module (slots 5 and 6) through a coaxial cable to the oscilloscope CH3 vertical input.

   d. *Connect oscilloscope to fourth VX4792 Arbitrary Waveform Generator module (slots 7 and 8):* Connect the WAVEFORM OUTPUT from the fourth module (slots 7 and 8) through a coaxial cable to the oscilloscope CH4 vertical input.

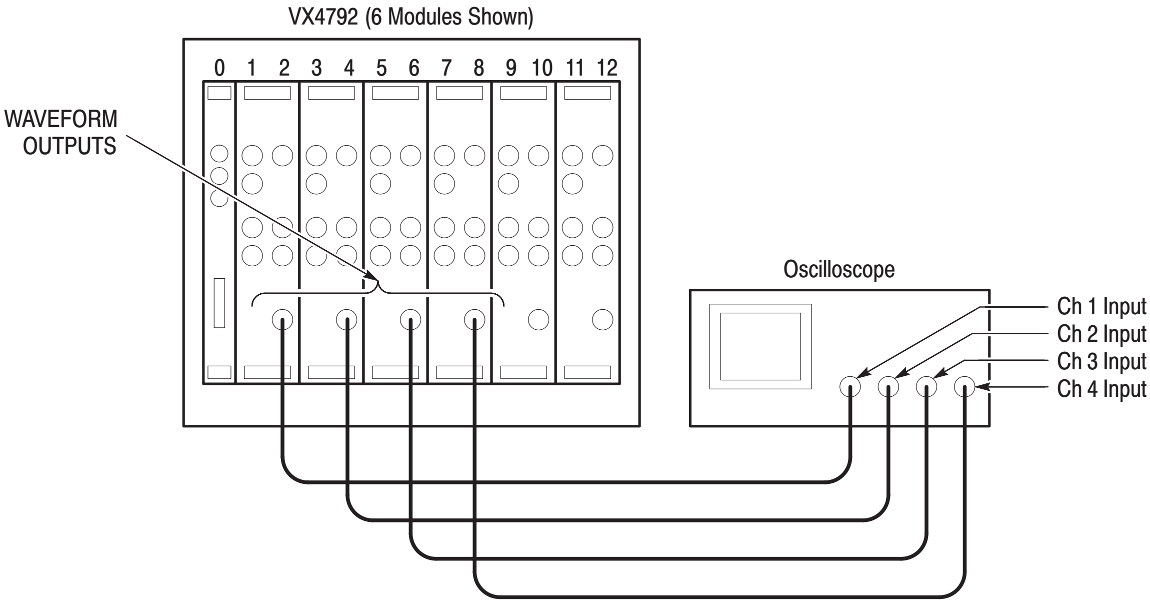


Figure F-19: Synchronous Operation Initial Test Setup

e. *Set oscilloscope controls:*

Vertical	CH1, CH2, CH3, CH4
Coupling (CH1–CH4)	DC
Scale (CH1–CH4)	0.2 V/div
Input Impedance (CH1–CH4)	50 Ω
Horizontal	
Sweep	2 ns/div
Trigger	
Source	CH1
Coupling	DC
Slope	Positive
Level	0 V
Mode	Auto

3. *To set the first and second VX4792 Arbitrary Waveform Generator controls, and select the waveform file, type:*

```
IBWRTF CLK64.PAT <RETURN>

IBWRTF MASTER.CMD <RETURN>

IBSAD 98 <RETURN>

IBWRTF CLK64.PAT <RETURN>

IBWRTF SLAVE.CMD <RETURN>
```

4. *Setup controls of additional VX4792 Arbitrary Waveform Generator modules (if installed):*

   a. *To set the third VX4792 Arbitrary Waveform Generator controls, type:*

   ```
   IBSAD 99 <RETURN>
   IBWRTF CLK64.PAT <RETURN>
   IBWRTF SLAVE.CMD <RETURN>
   ```

   b. *To set the fourth VX4792 Arbitrary Waveform Generator controls, type:*

   ```
   IBSAD 100 <RETURN>
   IBWRTF CLK64.PAT <RETURN>
   IBWRTF SLAVE.CMD <RETURN>
   ```

   c. *To set the fifth VX4792 Arbitrary Waveform Generator controls, type:*

   ```
   IBSAD 101 <RETURN>
   IBWRTF CLK64.PAT <RETURN>
   IBWRTF SLAVE.CMD <RETURN>
   ```

   d. *To set the sixth VX4792 Arbitrary Waveform Generator controls, type:*

   ```
   IBSAD 102 <RETURN>
   IBWRTF CLK64.PAT <RETURN>
   IBWRTF SLAVE.CMD <RETURN>
   ```

5. *To begin the test, type:*

   ```
   IBSAD 97 <RETURN>
   IBWRT "START" <RETURN>
   ```

6. *Check skew between the first, second, third and fourth modules:* Check that the skew (time variance) between the rising edges of the waveforms displayed on the oscilloscope is within 4 ns (see Figure F–20).

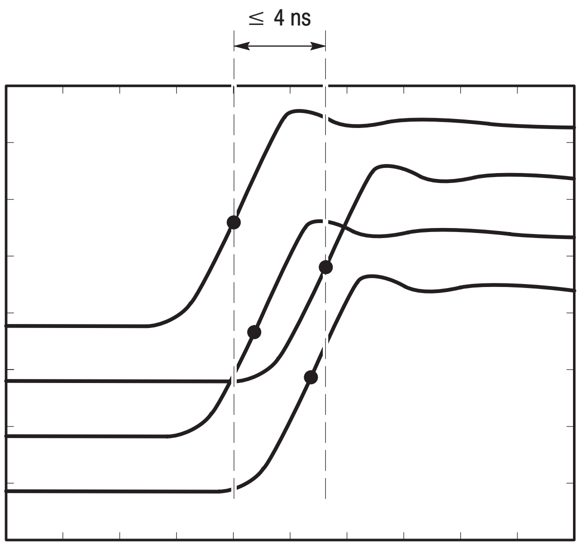


Figure F–20: Skew Between Modules

7. *Check skew between the first, second, third, and fifth modules (if installed):*

   a. Disconnect the coaxial cable from the WAVEFORM OUTPUT of the fourth module (slots 7 and 8).

   b. Connect the coaxial cable to the WAVEFORM OUTPUT of the fifth module (slots 9 and 10).

   c. Check that the skew (time variance) between the rising edges of the waveforms displayed on the oscilloscope is within 4 ns (see Figure F–20).

8. *Check skew between the first, second, third, and sixth modules (if installed):*

   a. Disconnect the coaxial cable from the WAVEFORM OUTPUT of the fifth module (slots 9 and 10).

   b. Connect the coaxial cable to the WAVEFORM OUTPUT of the sixth module (slots 11 and 12).

   c. Check that the skew (time variance) between the rising edges of the waveforms displayed on the oscilloscope is within 4 ns (see Figure F–20).

9. *Turn off equipment and disconnect test setup:*

   a. *Disable equipment:* Turn off all equipment.

   b. *Remove connections:* Disconnect all connections to the VX4792 Arbitrary Waveform Generator.

Appendix G: Functions

This appendix covers the following items:

- Differentiation
- Integration
- Random (rnd) function

Differentiation

The diff() function calculates the central deviation as the differential value. The equation below expresses the central deviation when the function f(x) is given at even intervals of Δx.

$$f'(x) \approx \frac{f(x + \Delta x) - f(x - \Delta x)}{(2\,\Delta x)}$$

In actual practice, when function f(x)is expressed by n values, the differential value $f'(x_i)$ at point x_i is given by the following equation:

$$f'(x_i) \approx n\frac{\{f(x_{i+1}) - f(x_{i-1})\}}{2}$$

Here, n is the number of waveform points and i is an integer in the range, i = 1, 2, ..., n.

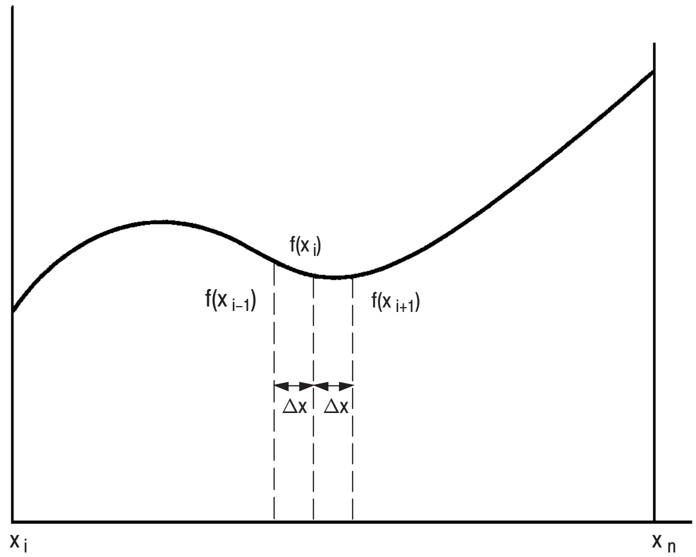


Figure G–1: Equation Differentiation

The values at the first and last points are obtained not from the center deviation, but from the following equations:

First point

$$f'(x_1) \approx \frac{n\{-3f(x_1) + 4f(x_2) - f(x_3)\}}{2}$$

Last point

$$f'(x_n) \approx \frac{n\{f(x_{n-2}) - 4f(x_{n-1}) + 3f(x_n)\}}{2}$$

Integration

The integ() function integrates numerically based on a trapezoidal formula. The trapezoidal formula is expressed with the following equation:

$$\int f(x)dx \approx \sum_{i=1}^{n} \frac{f(x_{i-1}) + f(x_i)}{2} \cdot \Delta x$$

$$= \Delta \frac{x}{2} \{f(x_1) + 2f(x_2) + 2f(x_3) + \ldots + 2f(x_{n-1}) + f(x_n)\}$$

Here, n is the number of waveform points and i is an integer in the range, i = 1, 2, ..., n.

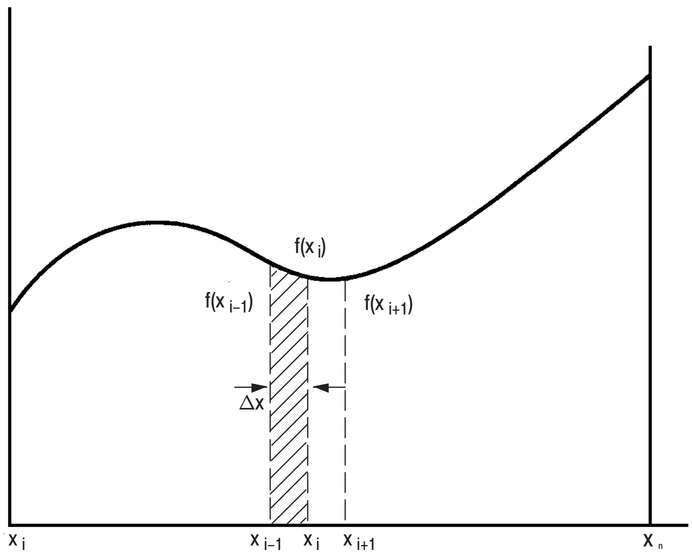


Figure G–2: Equation Integration

The integration is actually calculated with the following formula:

$$\int f(x)dx \approx 1/2\{f(x_1) + 2f(x_2) + 2f(x_3) + \ldots + 2f(x_{n-1}) + f(x_n)\}$$

However, the imaginary initial value $f(x_0)$ always takes a value of 0.

Random (rnd) Function

A random number generation algorithm uses a uniform distribution random generation routine and the central-limit theorem to derive Gaussian distribution random numbers.

Central-limit theorem: when the independent random variables X_1, X_2..., and X_n conform to an identical random distribution, the mean and variance of x = (X_1 + X_2 +... + X_n)/n are given as follows:

$$E(n) = \mu \qquad V(n) = \sigma^2/n$$

Even if the initial random distribution is not normal, if a reasonably large value for n is used, the arithmetical mean x of a considerably large number of variables will be close to the normal distribution.

In actuality, 12 is used for n, uniform random numbers are accumulated n times, and their arithmetical mean is derived as the ultimate Gaussian distribution random number.

The following algorithm is used to generate uniform distribution random numbers:

$$seed\ [n] = (253.0 \times seed\ [n\text{–}1] + 1.0)\ mod\ 16777216$$

$$ran = seed\ [n]\ /16777216$$

Glossary and Index

Glossary

ACFAIL~
A VXIBus backplane line that is asserted under the following conditions: by the VXIbus mainframe power supply when a power failure has occurred (either AC line source or power supply malfunction), or by the front panel ON/STANDBY switch when it is set to STANDBY.

AM
The amplitude modulation or waveform multiplication feature.

Arbitrary Waveform Generator (ARB)
Essentially a digital-to-analog converter with built-in memory, plus trigger and output mode settings.

A-Size Card
A VXIbus instrument module that is 100 mm X 160 mm X 20.32 mm (3.9 in X 6.3 in X 0.8 in) for a single-wide unit.

ASCII
Acronym for the American Standard Code for Information Interchange. Controllers transmit commands to the instrument using ASCII character encoding.

Autostep
Generates the predefined waveform in an autostep file; the next trigger advances the waveform.

Backplane
The printed circuit board that is mounted within a VXIbus mainframe to provide the interface between the VXIbus modules, and between the mainframe power supply and VXI modules.

BNF (Backus-Naur Form)
A standard notation system for command syntax diagrams. The syntax diagrams in this manual use BNF notation.

B-Size Card
A VXIbus instrument module that is 233.4 mm X 160 mm X 20.32 mm (9.2 in X 6.3 in X 0.8 in) for a single-wide unit.

Command
Any communication from a commander to a message-based servant.

Commander
A device that controls one or more devices (servants). A commander may be a servant of another commander, and is a message-based device which is also a bus master.

Controller
A computer or other device that sends commands to and accepts responses from the instrument.

C-Size Card
A VXIbus instrument module that is 340 mm X 233.4 mm X 30.48 mm (13.4 in X 9.2 in X 1.2 in) for a single-wide unit.

D-Size Card
A VXIbus instrument module that is 340 mm X 366.7 mm X 30.48 mm (13.4 in X 14.4 in X 1.2 in) for a single-wide unit.

Dynamic Configuration
A method of automatically assigning logical addresses to VXIbus devices at system power-on, or at other configuration times. Dynamic configuration allows for each slot to contain one or more devices, as well as different devices within a slot, to share address decoding hardware.

ECLTRG
Single-ended ECL trigger lines from VXI backplane that function as inter-module timing resources. Any module, including the Slot 0 module, may send and receive information on these lines.

Equation File
A data file that contains time region boundaries and algorithms that describe desired waveforms. The equation editor creates these files and compiles them into waveform files.

FAILED Indicator
A red LED indicator that lights when a device on the VXIbus has detected an internal fault.

Filter
An output parameter that restricts the output bandwidth.

GPIB
Acronym for General Purpose Interface Bus, the common name for the communications interface system defined by IEEE Std 488.

IACK Daisy Chain Driver
The circuit that drives the VXIbus Interrupt Acknowledge daisy chain line that runs continuously through all installed modules or through jumpers across the backplane.

IEEE
Acronym for the Institute for Electrical and Electronic Engineers.

Logical Address
An 8-bit number that uniquely identifies each VXIbus device in a system. The logical address defines a device's A16 register address, and indicates Commander/Servant relationships.

Mainframe
A rigid framework that provides mechanical support for modules inserted into a VXIbus backplane. The mainframe ensures that connectors mate properly, that adjacent modules do not contact each other, and that modules do not disengage from the backplane due to vibration or shock. The mainframe may also provide mechanical support and housing for power supplies and their distribution wiring to the backplane.

Marker
Front panel output signals that are synchronized to the output waveform.

Memory Device
A storage device (such as bubble memory, RAM, or ROM) that has configuration registers and memory attributes (such as type and access time).

Message
A series of data bytes that are treated as a single communication element, with a well-defined message body and terminator.

Message Based Device
A VXIbus device that supports VXI configuration and communication registers. Such devices support the word serial protocol, and and may support shared memory protocols.

Module
Typically consists of a board assembly and its associated mechanical parts; the VX4792 Arbitrary Waveform Generator is a module. A module may occupy one or more slots.

P1
The top-most backplane connector for a given module slot in a vertical VXIbus mainframe such as the Tektronix VX1410. The left-most backplane connector for a given slot in a horizontal mainframe, such as the Tektronix VX1405.

P2
The middle backplane connector for a given module slot in a VXIbus mainframe.

P3
The bottom backplane connector for a given module slot in a vertical VXIbus mainframe such as the Tektronix VX1410. The right-most backplane connector for a given slot in a horizontal mainframe, such as the Tektronix VX1405.

Query
A form of command that allows for inquiry to obtain status or data.

Register Based Device
A servant-only device which supports VXIbus configuration registers, but not high level VXIbus communication protocols. Register based devices are typically controlled by a commander via device-dependent register reads and writes.

Resource Manager
A message based commander that provides configuration management services such as configuring the address map, determining system hierarchy, allocating shared system resources, performing system self-test diagnostics, and initializing system commanders.

Self Calibration
A routine that verifies the basic calibration of the instrument module circuits, and adjusts the calibration to compensate for short- and long-term variables.

Self Test
A set of routines that test the operational functionality of the instrument module. These routines are performed on power-on, and on command.

Sequence File
An assembly of individual waveform files serially ordered to create a desired test pattern.

Servant
A device that is controlled by a commander. There are message-based and register-based servants.

Slot
A a position where a module can be inserted into a VXIbus backplane. Each slot provides the 96-pin J connectors to interface with one, two, or three P connectors on the VXI module.

Slot 0 Controller
See Slot 0 Module. See also Resource Manager.

Slot 0 Module
A VXIbus device that provides the minimum VXIbus Slot 0 services to slots 1–12 (CLK10 and the MODID module identify lines). The Slot 0 Module may also provide other VXIbus services such as CLK100, SYNC100, STARBUS, and trigger control.

System Hierarchy
The tree structure of the commander/servant relationships of all devices in the system at a given time. In the VXIbus structure, each servant has a commander. A commander may also have a commander.

TTLTRG
Open collector TTL lines used for inter-module timing and communications.

Word Serial Protocol (WSP)
The simplest required communication protocol supported by message-based devices in the VXIbus system. It uses the A16 communication registers to transfer data using a simple polling handshake method. WSP is a bidirectional word-oriented, serial protocol for VXIbus communications between message-based devices (that is, devices that include both communication registers and configuration registers).

Word Serial Communications
Inter-device communications using the Word Serial Protocol.

WSP
See Word Serial Protocol.

Index

Symbols

*CLS, 3–27
*ESE, 3–38
*ESR?, 3–39
*ESR? Query, 4–1
*IDN?, 3–43
*LRN?, 3–44
*OPC, 3–60
*OPT?, 3–62
*RST, 3–64
*SRE, 3–71
*SRE Command, 4–5
*STB?, 3–72
*STB? Query, 4–3
*TRG, 3–73
*TST?, 3–79
*WAI, 3–81

A

Accessories, Standard, 1–2
Address, Logical, 1–4
ALLEv?, 3–21
AM INPUT Connector, 2–2
Amplitude Modulated Signal, D–15
AMPLitude?, 3–22
Amplitude, Setting, 2–13
Applications Support, 1–2
Arbitrary Block, Description of, 3–6
Arguments
  Arbitrary block, 3–6
  Decimal numeric, 3–4
  Description of, 3–4
  SI prefix, 3–4
  String, 3–5
  Unit, 3–4
Arithmetic Operation, Procedure for checking, F–22
ASCII Code Chart, A–1
Autoconfigure, Logical Address, 1–4
Autostep Mode, Description of, 2–7
Autostep mode, Procedure for checking, F–21
AUTOStep:DEFine, 3–22

B

Backus-Naur-Form, 3–1
Bandwidth Filter, 2–14
Burst Mode
  Description of, 2–6
  Procedure for checking, F–15

C

CAL?, 3–23
Calibration
  Description of, 2–29
  Self–calibration, 2–30
Calibration and Diagnostic Commands
  *CAL?, 3–23
  *TST?, 3–79
  DIAG?, 3–32
  DIAG:RESUlt?, 3–33
  DIAG:SELect?, 3–33
  DIAG:STATe?, 3–34
  SELFcal?, 3–65
  SELFcal:RESULt?, 3–66
  SELFcal:SELect, 3–67
  SELFcal:STATe, 3–68
CH1?, 3–24
Character Encoding, 3–2
Clear SESR Command, 3–27
Clock
  Point rate, 2–19
  Setting source and frequency, 2–12
Clock Amplitude, Procedure for checking, F–27
Clock Frequency Accuracy, Procedure for checking, F–25
CLOCK INPUT Connector, 2–2
CLOCK OUTPUT Connector, 2–2
CLOCk?, 3–25
CLOCk:FREQuency, 3–25
CLOCk:SOURce, 3–26
Command Conventions
  Abbreviation, 3–10
  BNF notation, 3–1
  General, 3–10
  Structure of, 3–2
  Syntax, 3–1
  Upper and lower case, 3–10
Command Errors, B–2
Command Files, Performance Check Disk, F–4
Command Groups, List of, 3–13
Command Summaries
  Calibration and Diagnostic, 3–15
  Memory, 3–15
  Mode, 3–16
  Output, 3–17

Setup, 3–17
Status and Events, 3–18
Synchronization, 3–18
System, 3–19
Waveform, 3–19
Commands
Concatenating, 3–8
Descriptions of, 3–21
Quick Reference, 3–14
Common Commands
*CLS, 3–27
*ESE, 3–38
*ESR?, 3–39
*IDN?, 3–43
*LRN?, 3–44
*OPC, 3–60
*OPT?, 3–62
*RST, 3–64
*SRE, 3–71
*STB?, 3–72
*TRG, 3–73
*TST?, 3–79
*WAI, 3–81
Compiling Equations, 2–44
Components for Equations, 2–20
Concatenating Commands, 3–8
Configuration
Bus grant jumpers, 1–6
IACK jumper, 1–6
Jumper settings, 1–6
Preparing VXIbus mainframe, 1–6
Setting logical address, 1–4
Voltage and current requirements, 1–6
Connectors
AM INPUT, 2–2
CLOCK INPUT, 2–2
CLOCK OUTPUT, 2–2
Description of, 2–2
MARKER 1 OUTPUT, 2–2
MARKER 2 OUTPUT, 2–2
Rear, description of, 2–33
SYNC OUTPUT, 2–2
TRIGGER INPUT, 2–2
WAVEFORM OUTPUT, 2–2
Continuous Mode
Description of, 2–5
Procedure for checking, F–11
Conventions, Performance test procedure, F–2
Creating Waveforms, 2–3
CURVe, 3–27

D

Damped Sine Wave, D–17
Data Length
Explanation, 2–11
Not multiple of eight, 2–12
DATA?, 3–28
DATA:DESTination, 3–29
DATA:ENCDG, 3–29
DATA:SOURce, 3–30
DC Amplitude Accuracy, Procedure for checking, F–28
Decimal Numeric, Description of, 3–4
Default Settings, C–1
Defining Equations, 2–43
Delay, 2–16
Description, Product features, 1–1
DESE, 3–31
DESE Command, 4–4
DESER Register, 4–4
Device-Specific Messages, B–6
DIAG?, 3–32
DIAG:RESUlt?, 3–33
DIAG:SELect?, 3–33
DIAG:STATe?, 3–34
Diagnostics
Description of, 2–29
Front panel indicators, 2–30
Routines, 2–30
SYSFAIL~ signal, 2–29
Diagrams, Syntax, 3–10
diff() function, G–1

E

Electrical Characteristics, E–2
Electromagnetic Disk Signal 1, D–20
Electromagnetic Disk Signal 2, D–21
Enable Registers, 4–4
Environmental Characteristics, E–7
EQUAtion:COMPile, 3–35
EQUAtion:DEFine, 3–36
EQUAtion:WPOints, 3–37
Equations, 2–18
Compiling, 2–44
Components, 2–20
Defining, 2–43
Equipment Required, For performance tests, F–3
Error, No events, B–1
Error Messages, Listed, B–1

*ESE, 4–4
ESER Register, 4–4
Event Handling, 4–1
Event Queue, 4–6
Event Status Enable Command, 3–38
EVENT?, 3–39
EVMsg?, 3–40
EVQty?, 3–40
Example Equations, 2–18
Examples
  Compiling equations, 2–44
  Defining equations, 2–43
  Instrument control and basic waveform generation, 2–40
  Synchronous operation, 2–41
  Transferring waveforms, 2–45
Execution Errors, B–3
Execution Warning, B–6
External AM Operation, 2–17
  Procedure for checking, F–22
External CLOCK INPUT, Procedure for checking, F–37
External Trigger Level Accuracy, Procedure for checking, F–35

F

Factory Default Settings, C–1
FAILED Indicator, 2–2
Files, Performance Check Disk, F–4
FILTer, 3–41
Filter, Setting, 2–14
Floating Point Processing, 2–20
Frequency Modulated Signal, D–16
Frequency, Clock, 2–19
Functional Check
  Description of, 1–11
  Initial setup, 1–11
  Procedure, 1–11
Functions
  Differentiation, G–1
  Integration, G–3
  Random (rnd) function, G–4

G

Gated Mode
  Description of, 2–5
  Procedure for checking, F–16
Gaussian Pulse, D–7
Generating Waveforms, 2–3
GPIB Code Chart, A–1

H

HEADer, 3–42
Header, Description of, 3–6

I

ID?, 3–43
Identify Query, 3–43
Indicators
  Description of, 2–2
  FAILED, 2–2
  POWER, 2–2
Installation, Into mainframe, 1–7
Installation Record, 1–8
integ() function, G–3
Internal Errors, B–5
Internal Warnings, B–6

L

Learn Query, 3–44
Linear Frequency Sweep, D–13
Logarithmic Frequency Sweep, D–14
Logical Address, Setting, 1–4
Lorentz Pulse, D–8

M

Mainframe
  Current requirements, 1–6
  Jumper settings, 1–6
  Preparation of, 1–6
  Voltage requirements, 1–6
MARKER 1 OUTPUT Connector, 2–2
MARKER 2 OUTPUT Connector, 2–2
MARKer:DATA, 3–46
MARKER<x>:AOFF, 3–45
MARKER<x>:POINt, 3–45
Mechanical Characteristics, E–7
Memory, Waveform, 2–10
Memory Commands
  MEMory?, 3–47
  MEMory:CATalog?, 3–48
  MEMory:CATalog:ALL?, 3–49
  MEMory:CATalog:AST?, 3–49
  MEMory:CATalog:EQU?, 3–50
  MEMory:CATalog:SEQ?, 3–51
  MEMory:CATalog:WFM?, 3–51
  MEMory:COMMent, 3–52
  MEMory:COPY, 3–53

MEMory:DELete, 3–53
MEMory:FREE?, 3–54
MEMory:FREE:ALL?, 3–55
MEMory:LOCk, 3–56
MEMory:REName, 3–57
MEMory?, 3–47
MEMory:CATalog?, 3–48
MEMory:CATalog:ALL?, 3–49
MEMory:CATalog:ALL:AST?, 3–49
MEMory:CATalog:ALL:EQU?, 3–50
MEMory:CATalog:SEQ?, 3–51
MEMory:CATalog:WFM?, 3–51
MEMory:COMMent, 3–52
MEMory:COPY, 3–53
MEMory:DELete, 3–53
MEMory:FREE?, 3–54
MEMory:FREE:ALL?, 3–55
MEMory:LOCk, 3–56
MEMory:REName, 3–57
Message, Handling, 4–1
Messages
Error, B–1
Event, B–1
Program and response, 3–1
MODE, 3–57
Mode Commands
*TRG, 3–73
MODE, 3–57
RUNNing, 3–65
STARt, 3–72
STOP, 3–73
TRIGger?, 3–74
TRIGger:IMPedance, 3–74
TRIGger:INPut, 3–75
TRIGger:LEVel, 3–76
TRIGger:OUTPut, 3–77
TRIGger:POLarity, 3–77
TRIGger:SLOPe, 3–78

N

NTSC Directory, D–22
Nyquist Pulse, D–12

O

OFFSet, 3–59
Offset Accuracy, Procedure for checking, F–30
Offset, Setting, 2–13
Operating Modes
Autostep, 2–7
Burst, 2–6
Continuous, 2–5
Description of, 2–5
Gated, 2–5
Procedure for checking, F–11
Triggered, 2–5
Waveform Advance, 2–7
OPERation, 3–61
Operation, Synchronous, 2–41
Operation Complete Command, 3–60
Options Query, 3–62
Output Commands
OUTPut?, 3–62
OUTPut:STATe, 3–63
OUTPut:SYNC, 3–64
Output Queue, 4–5
OUTPut?, 3–62
OUTPut:STATe, 3–63
OUTPut:SYNC, 3–64

P

P1 Connectors, Description of, 2–33
P2 Connectors, Description of, 2–33
Performance Check Disk, Waveform Files, F–4
Performance Test Procedure, Interval, F–2
Performance Verification Procedure
Arithmetic operation checks, F–22
Check autostep mode, F–21
Check burst mode, F–15
Check clock amplitude, F–27
Check clock frequency accuracy, F–25
Check continuous mode, F–11
Check DC amplitude accuracy, F–28
Check external AM operation, F–22
Check gated mode, F–16
Check offset accuracy, F–30
Check triggered mode, F–12
Check waveform advance mode, F–19
Conventions, F–2
Description of, F–1
Equipment required, F–3
External CLOCK INPUT check, F–37
External trigger level accuracy check, F–35
List of, F–1
Operating mode checks, F–11
Prerequisites for, F–3
Pulse response check, F–30
SYNC OUTPUT and MARKER OUTPUT amplitude checks, F–33
Synchronous operation check, F–39
System setup, F–8
Point Count, 2–19

Point Rate Clock, 2–19
POWER Indicator, 2–2
Procedure
 Check arithmetic operation, F–22
 Check autostep mode, F–21
 Check burst mode, F–15
 Check clock amplitude, F–27
 Check clock frequency accuracy, F–25
 Check continuous mode, F–11
 Check DC amplitude accuracy, F–28
 Check external AM operation, F–22
 Check external CLOCK INPUT, F–37
 Check external trigger level accuracy, F–35
 Check gated mode, F–16
 Check offset accuracy, F–30
 Check operating modes, F–11
 Check pulse response, F–30
 Check SYNC OUTPUT and MARKER OUTPUT amplitude, F–33
 Check synchronous operation, F–39
 Check triggered mode, F–12
 Check waveform advance mode, F–19
Product Support, 1–2
Programming Examples
 Compiling equations, 2–44
 Defining equations, 2–43
 Instrument control and basic waveform generation, 2–40
 Synchronous operation, 2–41
 Transferring waveforms, 2–45
Pseudo–Random (PRBS) Signal, D–19
Pulse Response, Procedure for checking, F–30
Pulse Signal
 Gaussian, D–7
 Lorentz, D–8
 Nyquist, D–12
 Rising, Falling Exponential, D–11
 Sin(x)/x, D–9
 Squared Sine, D–10
Pulse Width Modulated Signal, D–18

Q

Query, Structure of, 3–2
Query Responses, Description of, 3–9
Queue
 Event, 4–6
 Output, 4–5
Quick Reference, Commands, 3–14

R

Ramp, Example of, D–5
Random (rnd) function, G–4
Reference, Commands, 3–14
Register
 Description of, 4–1
 DESER, 4–4
 Enable, 4–4
 ESER, 4–4
 SBR, 4–3
 SESR, 4–1
 SRER, 4–5
 Status, 4–1
Removal, From mainframe, 1–9
Reset Command, 3–64
Reset, Factory Default Settings, C–1
Resolution, 2–11
Rising, Falling Exponential Pulse, D–11
RUNNing, 3–65

S

Sample Waveforms, D–1
SBR Register, 4–3
Self–calibration, 2–30
SELFcal?, 3–65
SELFcal:RESULt?, 3–66
SELFcal:SELect, 3–67
SELFcal:STATe, 3–68
Sequence Control, Description of, 2–8
SEQUence:DEFine, 3–69
SEQUence:EXPand, 3–70
Serial Poll, 4–3
Service Request Enable Command, 3–71
SESR Register, 4–1
Setup Commands
 AMPLitude, 3–22
 CH1, 3–24
 CLOCk?, 3–25
 CLOCk:FREQuency, 3–25
 CLOCk:SOURce, 3–26
 FILTer, 3–41
 OFFSet, 3–59
 OPERation, 3–61
 WAVeform, 3–81
SI Units, description of, 3–4
Signal Examples
 Ramp wave, D–5

Sine wave, D–3
Square wave, D–6
Triangle wave, D–4
Sin(x)/x Pulse, D–9
Sine Wave, Example of, D–3
Specifications
Description of tables, E–1
Electrical, E–2
Environmental, E–7
Mechanical, E–7
Performance conditions, E–1
Square Wave, Example of, D–6
Squared Sine Pulse, D–10
SRER Register, 4–5
Standard Accessories, 1–2
STARt, 3–72
Status, 4–1
Status and Error Commands
DESE, 4–4
*ESE, 4–4
*ESR?, 4–1
*SRE, 4–5
*STB?, 4–3
Status and Event Commands
*CLS, 3–27
*ESE, 3–38
*ESR?, 3–39
*SRE, 3–71
*STB?, 3–72
ALLEv?, 3–21
DESE, 3–31
EVENT?, 3–39
EVMsg?, 3–40
EVQty?, 3–40
Status and Events, Processing of, 4–6
Status Byte Register Query, 3–72
Status Event Status Register Query, 3–39
Status Registers, 4–1
STOP, 3–73
String, Description of, 3–5
Support, Applications, 1–2
SYNC OUTPUT and MARKER OUTPUT Amplitude, Procedure for checking, F–33
SYNC OUTPUT Connector, 2–2
SYNC OUTPUT, Setting, 2–18
Synchronization Commands
*OPC, 3–60
*WAI, 3–81
Synchronous Operation
Procedure for checking, F–39
Programming example, 2–41
Syntax
Delimiters, description of, 3–3
Diagrams, 3–10
Special characters, 3–3
White space, 3–3
System Commands
*IDN?, 3–43
*LRN?, 3–44
*OPT?, 3–62
*RST, 3–64
HEADer, 3–42
ID?, 3–43
VERBose, 3–80
System Events, B–5
System Setup, For performance tests, F–8

T

Television Signals, D–22
Test Interval, F–2
Test Query, 3–79
Testing
Functional check procedure, F–1
Levels of, F–1
Performance verification procedure, F–1
Timing, 2–16
Timing Relationships, 2–15
Triangle Wave, Example of, D–4
Trigger Command, 3–73
TRIGGER INPUT Connector, 2–2
Trigger Parameters
Impedance, 2–16
Level, 2–16
Polarity, 2–16
Slope, 2–15
Trigger Parameters, Setting, 2–15
TRIGger?, 3–74
TRIGger:IMPedance, 3–74
TRIGger:INPut, 3–75
TRIGger:LEVel, 3–76
TRIGger:OUTPut, 3–77
TRIGger:POLarity, 3–77
TRIGger:SLOPe, 3–78
Triggered Mode
Description of, 2–5
Procedure for checking, F–12

V

VERBose, 3–80

W

Wait Command, 3–81
WAVeform, 3–81
Waveform Advance Mode
  Description of, 2–7
  Procedure for checking, F–19
Waveform Commands
  AUTOStep:DEFine, 3–22
  CURVe, 3–27
  DATA?, 3–28
  DATA:DESTination, 3–29
  DATA:ENCDG, 3–29
  DATA:SOURce, 3–30
  EQUAtion:COMPile, 3–35
  EQUAtion:DEFine, 3–36
  EQUAtion:WPOints, 3–37
  MARKer:DATA, 3–46
  MARKER<x>:AOFF, 3–45
  MARKER<x>:POINt, 3–45
  SEQUence:DEFine, 3–69
  SEQUence:EXPand, 3–70
  WAVFrm?, 3–82
  WFMPre?, 3–83
  WFMPre:BIT_NR, 3–83
  WFMPre:BN_FMT, 3–84
  WFMPre:BYT_NR, 3–85
  WFMPre:BYT_OR, 3–85
  WFMPre:CRVCHK, 3–86
  WFMPre:ENCDG, 3–87
  WFMPre:NR_PT, 3–87
  WFMPre:PT_FMT, 3–88
  WFMPre:PT_OFF, 3–89
  WFMPre:WFID, 3–89
  WFMPre:XINCR, 3–90
  WFMPre:XUNIT, 3–91
  WFMPre:XZERO, 3–91
  WFMPre:YMULT, 3–92
  WFMPre:YOFF, 3–93
  WFMPre:YUNIT, 3–93
  WFMPre:YZERO, 3–94
Waveform Files
  Description of, 2–3
  Performance Check Disk, F–4
Waveform Library, D–1
Waveform Memory
  Data length, 2–11
  Description of, 2–10
WAVEFORM OUTPUT Connector, 2–2
Waveforms, Generating, 2–3
WAVFrm?, 3–82
WFMPre?, 3–83
WFMPre:BIT_NR, 3–83
WFMPre:BN_FMT, 3–84
WFMPre:BYT_NR, 3–85
WFMPre:BYT_OR, 3–85
WFMPre:CRVCHK, 3–86
WFMPre:ENCDG, 3–87
WFMPre:NR_PT, 3–87
WFMPre:PT_FMT, 3–88
WFMPre:PT_OFF, 3–89
WFMPre:WFID, 3–89
WFMPre:XINCR, 3–90
WFMPre:XUNIT, 3–91
WFMPre:XZERO, 3–91
WFMPre:YMULT, 3–92
WFMPre:YOFF, 3–93
WFMPre:YUNIT, 3–93
WFMPre:YZERO, 3–94