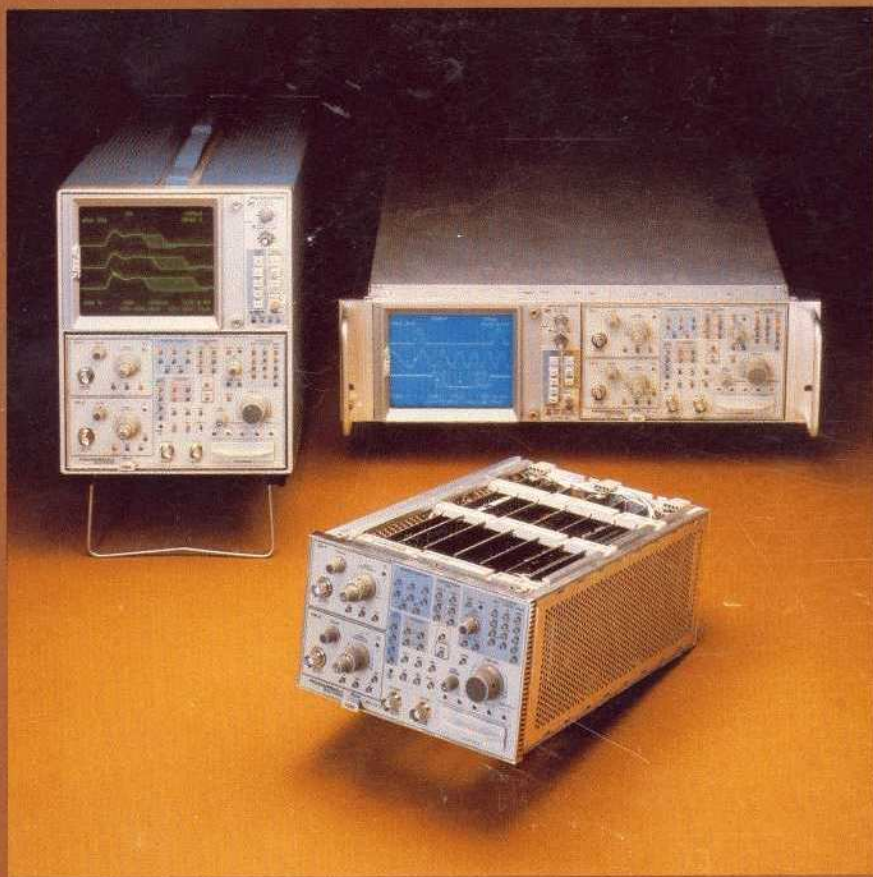


# 7D20

## Programmable Digitizer



# **7D20**

## **Programmable Digitizer**

### **With Options**

Serial Number \_\_\_\_\_

**PLEASE CHECK FOR CHANGE  
INFORMATION AT THE REAR  
OF THIS MANUAL**

## **INSTRUCTIONS FOR COMPLETING THE SOFTWARE/FIRMWARE PERFORMANCE REPORT**

- I. Please type or print clearly. Use a separate Software/Firmware Performance Report (SFPR) for each problem.
- II. **SECTION A**  
Fill in the serial number of the 7D20 PROGRAMMABLE DIGITIZER. Press the ID key on the 7D20 and copy the entire line of Version Information, beginning with TEK/7D20.
- III. **SECTION B**  
Use your complete company mailing address. Please include the name and phone number of the person reporting the error. Also, be sure to fill in the name of the person submitting the SFPR.
- IV. **SECTION C**  
Check the reason for report and whether the error is reproducible. We cannot properly evaluate the problem if we have difficulty in reproducing the error.
- V. **SECTION D**  
Give a complete description of the system configuration on which the problem occurred. Please include related peripherals, interfaces, options, and operating system.
- VI. **SECTION E**  
Describe the problem completely. Include any information which might help in evaluating the error with the SFPR. If you have determined a procedure to avoid the error condition, please include this procedure. If this problem prevents you from accomplishing any useful work with the product, please state this fact. Be sure to include with the SFPR any information (programs, listings, hard copies, etc.) which will help us duplicate your problem.
- VII. **SECTION F**  
This section is for use by Tektronix Lab Scopes Software Maintenance personnel. DO NOT WRITE IN THIS SPACE.
- VIII. Mail all copies of the Software/Firmware Performance Report to:

**TEKTRONIX, INC.  
LAB SCOPES SOFTWARE MAINTENANCE  
P.O. BOX 500 DEL. STA. 39-285  
BEAVERTON, OREGON 97077**

SEND TO: TEKTRONIX, INC.  
LAB SCOPES SOFTWARE MAINTENANCE  
P.O. BOX 800 DEL. STA. 38-288  
BEAVERTON, OREGON 97077

**7D20 PROGRAMMABLE DIGITIZER  
SOFTWARE/FIRMWARE  
PERFORMANCE REPORT**

7D20 PROGRAMMABLE DIGITIZER SERIAL # \_\_\_\_\_  
VERSION LINE OF 7D20 DISPLAY WHEN ID KEY IS PRESSED TEK/7D20

COMPANY NAME: \_\_\_\_\_  
USER: \_\_\_\_\_  
ADDRESS: \_\_\_\_\_  
CITY: \_\_\_\_\_ STATE: \_\_\_\_\_ ZIP: \_\_\_\_\_  
PHONE: \_\_\_\_\_ EXTENSION: \_\_\_\_\_  
FORM SUBMITTED BY: \_\_\_\_\_ DATE: \_\_\_\_\_

REASON FOR REPORT:  
 Software/Firmware Error  
 Documentation Error  
 Suggested Enhancement  
  
IS THE ERROR REPRODUCIBLE?  
 Yes  No  
 Intermittent

SYSTEM DESCRIPTION: (Hardware, software, firmware and host related to the problem)  
MAINFRAME: \_\_\_\_\_ MAINFRAME SERIAL # \_\_\_\_\_

DESCRIPTION OF PROBLEM:

LIST ENCLOSURES:

INTERNAL USE ONLY  
(DO NOT WRITE BELOW THIS LINE)

ACTION TAKEN:

RESPONDER:

DATE OF RESPONSE:

DATE RECEIVED \_\_\_\_\_

SFPR # \_\_\_\_\_

SFPR LOG

**Tektronix**



# TABLE OF CONTENTS

	Page
OPERATORS SAFETY SUMMARY .....	x
DESCRIPTION .....	xii

## 1 GENERAL INFORMATION

DESCRIPTION .....	1-1
ORANGE COLOR USED IN THIS MANUAL .....	1-1
TECHNICAL MANUALS .....	1-1
OPERATORS MANUAL .....	1-1
SERVICE MANUAL .....	1-2
INSTALLATION .....	1-3
RACKMOUNT MAINFRAMES .....	1-4
COMPATIBILITY .....	1-4
PACKAGING FOR SHIPMENT .....	1-4
SPECIFICATIONS .....	1-6
STANDARD ACCESSORIES .....	1-12
OPTIONAL ACCESSORIES (not included) .....	1-12

## 2 OPERATING INSTRUCTIONS

OPERATION .....	2-1
CONTROLS AND CONNECTORS .....	2-1
DISPLAY .....	2-1
FUNCTION KEYS .....	2-8
ERROR MESSAGES .....	2-8
GET-ACQUAINTED EXERCISES .....	2-9
PRELIMINARY SET-UP .....	2-9
DISPLAY CALIBRATION .....	2-10
DISPLAY ADJUSTMENT PROCEDURE .....	2-11
INITIALIZE .....	2-14
OBTAINING A WAVEFORM DISPLAY .....	2-16
EXERCISES .....	2-17
INDEX .....	2-17
EXERCISE 1—USING THE VERTICAL CONTROLS .....	2-17
SETTING THE INPUT COUPLING .....	2-19
HOW TO SET THE VZR (Vertical Zero Reference) .....	2-19
CHANGING THE VERTICAL SENSITIVITY (VOLTS/DIV) .....	2-21
USING THE VARIABLE VOLTS/DIV CONTROL .....	2-21
EXERCISE 2—USING THE AQR (ACQUIRE) MODE .....	2-22
EXERCISE 3—USING THE WAVEFORM MEMORY .....	2-24
COPYING AND DISPLAYING A WAVEFORM .....	2-25
A NOTE ABOUT THE DSW (DiSplayed Waveform) ORDER .....	2-27
EXERCISE 4—CHANGING THE DESIGNATED CURSOR	
WAVEFORM .....	2-28

# TABLE OF CONTENTS (CONT)

	Page
OPERATING INSTRUCTIONS (CONT)	
EXERCISE 5—HOLDING A WAVEFORM IN MEMORY .....	2-31
EXERCISE 6—USING THE CURSORS .....	2-32
CURSOR 1 .....	2-32
CURSOR 2 AND THE $f$ (function shift) KEY .....	2-34
USING THE CURSORS <b>INDE</b> pendently OR <b>ALIG</b> Ned .....	2-36
EXERCISE 7—USING THE CURSOR WAVEFORM DISPLAY	
MODIFIER KEYS .....	2-42
HOW TO VERTICALLY REPOSITION THE CURSOR	
WAVEFORM .....	2-43
HOW TO VERTICALLY EXPAND OR COMPRESS THE	
CURSOR WAVEFORM .....	2-46
USING HMAG (HORIZONTAL MAGNIFY) AND CURSOR	
MOVEMENT .....	2-49
HMAG AND THE <b>REF</b> erence WAVEFORM .....	2-52
HOW TO USE HMAG <b>ALL</b> .....	2-56
USING THE <b>VS</b> (versus) FUNCTION .....	2-63
USING THE <b>REF</b> erence WITH <b>VS</b> .....	2-65
<b>VECTORS</b> AND <b>DOTS</b> .....	2-66
EXERCISE 8—USING THE AVERAGE AND ENVELOPE	
FUNCTIONS AND SET N .....	2-67
EXERCISE 9—DIGITIZING MODES AND THE TIME/DIV	
CONTROL .....	2-72
REAL-TIME AND EXTENDED REAL TIME .....	2-72
ROLL MODE AND EX <b>TER</b> nal CLOCK .....	2-72
EQUIVALENT TIME MODE .....	2-73
EXERCISE 10—SETTING THE TRIGGER POSITION .....	2-74
EXERCISE 11—USING THE HOLD NEXT .....	2-77
REAL-TIME AND EXTENDED REAL TIME .....	2-78
EQUIVALENT TIME MODE .....	2-78
ROLL MODE AND EX <b>TER</b> nal CLOCK .....	2-78
EXERCISE 12—USING THE EXTERNAL CLOCK .....	2-79
DETAILED OPERATING INSTRUCTIONS .....	2-81
INTRODUCTION .....	2-81
POWER-UP .....	2-81
MENU FUNCTIONS .....	2-83
MASTER MENU SELECTIONS .....	2-83
1,# STORE PANEL # .....	2-83
2,# RECALL PANEL # .....	2-84
3 DISPLAY CAL PATTERN .....	2-84
4 UTILITIES .....	2-84
TEST MENU .....	2-86
PROMPTS AND WARNINGS .....	2-88
DISPLAY CAL <b>IB</b> ration .....	2-90

# TABLE OF CONTENTS (CONT)

	Page
OPERATING INSTRUCTIONS (CONT)	
READOUT DISPLAY.....	2-91
CURRENT CONTROL SETTINGS AND PROMPT FIELD.....	2-91
CURSOR WAVEFORM PARAMETERS.....	2-91
VERTICAL CONTROLS.....	2-92
SIGNAL CONNECTION.....	2-92
INPUT CONNECTORS.....	2-92
VOLTS/DIV (VARIABLE).....	2-92
POSITION.....	2-93
COUPLING (INPUT).....	2-93
AQR (ACQUIRE) GAIN.....	2-94
CH2 INV.....	2-94
ACQUIRE MODE.....	2-95
CH1 $\leftrightarrow$ 1.....	2-95
BOTH.....	2-95
ADD $\leftrightarrow$ 1.....	2-95
CH2 $\leftrightarrow$ 2.....	2-96
HOLD.....	2-96
<i>f</i> (shift function).....	2-96
TIME/DIV.....	2-96
DIGITIZING MODES.....	2-96
Extended Real-Time Digitizing (ERD).....	2-98
TRIGGERING.....	2-99
SOURCE.....	2-99
COUPLING.....	2-100
MODE.....	2-101
+SLOPE.....	2-103
TRIG'D.....	2-104
LEVEL.....	2-104
$\leftrightarrow$ TRIG POS $\leftrightarrow$ .....	2-104
HORIZ POSITION.....	2-105
VECTOR.....	2-105
MEMORY DISPLAY.....	2-105
COPY.....	2-106
CSW.....	2-107
<i>f</i> CSW REF.....	2-107
CURSOR WFM.....	2-108
VPUP $\leftrightarrow$ AND VPDN $\leftrightarrow$ (Vertical Position Up and Vertical Position Down).....	2-108
VCMP $\leftrightarrow$ AND VXP $\leftrightarrow$ (Vertical Compress and Expand).....	2-109
HMAG (Horizontal Magnify).....	2-110
<i>f</i> HMAG ALL.....	2-110
VS.....	2-110

# TABLE OF CONTENTS (CONT)

	Page
OPERATING INSTRUCTIONS (CONT)	
CURSORS .....	2-112
CURSOR COORDINATES .....	2-112
CURSORS AND MAGNIFIED WAVEFORMS .....	2-113
$\diamond 1 \diamond$ .....	2-114
$\diamond 2 \diamond$ .....	2-115
$\Delta$ ON $\Delta$ OFF .....	2-115
INDEPendent .....	2-115
ALIGN .....	2-115
SET N .....	2-115
AVE .....	2-116
<i>f</i> , AVE N .....	2-118
ENV .....	2-118
<i>f</i> , ENV N .....	2-120
<i>f</i> , TEST .....	2-120
RQS (REQUEST SERVICE) .....	2-121
<i>f</i> RQS # .....	2-121
REMOTE ONLY (indicator) .....	2-121
ID (ADDRed Indicator) .....	2-121
EXT CLOCK .....	2-122
EXT TRIG .....	2-123
GPIB (connector) .....	2-123
<b>3</b> OPERATIONAL THEORY	
BLOCK DIAGRAM DESCRIPTION .....	3-1
OVERALL BLOCK DIAGRAM DESCRIPTION .....	3-1
CHARGE COUPLED DEVICE (CCD) OPERATION .....	3-4
DIGITIZING MODES .....	3-4
WAVEFORM DISPLAY .....	3-11
<b>4</b> GPIB	
INTRODUCTION .....	4-1
KEY FEATURES .....	4-1
SECTION CONTENTS .....	4-2
GPIB DESCRIPTION .....	4-2
POWER UP .....	4-3
GPIB MODE, TERMINATOR, AND ADDRESS SELECTION .....	4-4
ADDRESS SELECTION .....	4-5
TERMINATOR SELECTION .....	4-6
MODE SELECTION .....	4-6
ALTERNATE GPIB STATUS INDICATORS .....	4-7
REMOTE AND LOCAL OPERATION .....	4-7
INDICATORS .....	4-8
GPIB INTERFACE FUNCTIONS .....	4-8

# TABLE OF CONTENTS (CONT)

	Page
GPIB (CONT)	
MESSAGES .....	4-9
GPIB INTERFACE MESSAGES .....	4-9
TALK AND LISTEN ADDRESS (MTA, MLA) .....	4-9
SERIAL POLL ENABLE COMMAND (SPE) .....	4-10
GROUP EXECUTE TRIGGER (GET) .....	4-10
DEVICE CLEAR (DCL) .....	4-10
SELECTED DEVICE CLEAR (SDC) .....	4-10
INTERFACE CLEAR (IFC) .....	4-10
DEVICE-DEPENDENT MESSAGES .....	4-11
INTRODUCTION .....	4-11
COMMAND USAGE .....	4-11
QUERY USAGE .....	4-13
DATA TRANSFERS .....	4-16
Readout .....	4-17
Text .....	4-17
Waveforms .....	4-19
Waveform Curve .....	4-23
Waveform Preamble and Curve .....	4-24
SERVICE REQUESTS .....	4-25
USING SRQ .....	4-25
Service Request Masks .....	4-31
EVENT QUERIES .....	4-32
Event Classification .....	4-35
Event Codes .....	4-36
USER-REQUEST FUNCTIONS .....	4-37
COMMAND LIST .....	4-38
SAMPLE PROGRAMS .....	4-77
INTRODUCTION .....	4-77
SAMPLE PROGRAM LIST .....	4-77
4050-Series Operating Programs .....	4-78
4050-Series Application Programs .....	4-89
4041 Operating Programs .....	4-97
4041 Application Programs .....	4-108

## 5 APPLICATIONS

APPLICATION 1—MONITORING SLOWLY CHANGING EVENTS WITH THE 7D20 AND 7D11 .....	5-1
INSTRUMENTATION SET-UP #1 .....	5-2
APPLICATION 2—ULTRASONIC, NON-DESTRUCTIVE TESTING USING 7D20 .....	5-3
INSTRUMENTATION SET-UP #2 .....	5-4
APPLICATION 3—MONITORING INTERCELLULAR NEURONAL DISCHARGE .....	5-6

# TABLE OF CONTENTS (CONT)

	Page
APPLICATIONS (CONT)	
APPLICATION 4—MEASURING PULSE JITTER, FREQUENCY SHIFT, AND AMPLITUDE VARIATIONS USING THE 7D20 .....	5-7
APPLICATION 5—AC LOADLINE ANALYSIS USING THE 7D20 AND 7A13 .....	5-9
INSTRUMENTATION SET-UP #5 .....	5-10
CALIBRATION .....	5-11
OPERATION .....	5-11
APPLICATION 6—USING THE 7D20 WITH THE 7854 FOR WAVEFORM PROCESSING .....	5-13
INSTRUMENTATION SET-UP #6 .....	5-14
MANUAL TRANSFER .....	5-15
COMPUTER CONTROLLED TRANSFER .....	5-15

## 6 INSTRUMENT OPTIONS

INDEX

CHANGE INFORMATION

### NOTE

*Service Personnel: Refer to the 7D20 Service Manual for information on installation, Theory of Operation, Maintenance, Checks and Adjustment, Parts List and Diagrams.*

### RELATED DOCUMENT

The tabbed "ACCESSORIES" page at the rear of the Service Manual lists the Tektronix part numbers for all Standard Accessories provided with this product. Also, Section 1, General Information, contains a brief basic-content description for the following publications:

#### MANUALS (Standard Accessories)

- 7D20 Operators
- 7D20 Service



# LIST OF ILLUSTRATIONS

Figure No.	Page
1-1	1-4
1-2	1-13
2-1	2-2
2-2	2-6
2-3	2-10
2-4	2-10
2-5	2-11
2-6	2-11
2-7	2-12
2-8	2-12
2-9	2-13
2-10	2-13
2-11	2-14
2-12	2-16
2-13	2-18
2-14	2-21
2-15	2-22
2-16	2-24
2-17	2-24
2-18	2-28
2-19	2-28
2-20	2-31
2-21	2-33
2-22	2-33
2-23	2-34
2-24	2-41
2-25	2-42
2-26	2-44
2-27	2-46
2-28	2-49
2-29	2-52
2-30	2-56
2-31	2-63
2-32	2-66
2-33	2-67
2-34	2-72
2-35	2-75
2-36	2-75
2-37	2-77
2-38	2-83
2-39	2-85
2-40	2-85
2-41	2-87
2-42	2-87
2-43	2-87
2-44	2-88
2-45	2-92

# LIST OF ILLUSTRATIONS (CONT)

Figure No.	Page
2-46	2-93
2-47	2-94
2-48	2-95
2-49	2-97
2-50	2-98
2-51	2-103
2-52	2-105
2-53	2-106
2-54	2-107
2-55	2-108
2-56	2-109
2-57	2-110
2-58	2-111
2-59	2-111
2-60	2-113
2-61	2-113
2-62	2-114
2-63	2-114
2-64	2-116
2-65	2-116
2-66	2-119
2-67	2-122
2-68	2-122
3-1	3-2
3-2	3-5
3-3	3-7
3-4	3-8
3-5	3-9
4-1	4-3
4-2	4-5
4-3	4-5
4-4	4-6
4-5	4-6
4-6	4-9
4-7	4-11
4-8	4-12
4-9	4-13
4-10	4-14
4-11	4-15
4-12	4-15
4-13	4-18
4-14	4-27
4-15	4-28
4-16	4-30
4-17	4-33
4-18	4-34

# LIST OF ILLUSTRATIONS (CONT)

Figure No.		Page
5-1	Viewing a reflected pulse signal showing the cursor position relative to the trigger pulse.....	5-5
5-2	Using the ENVeloping mode to show frequency shift.....	5-7
5-3	Using signal ENVeloping and multi-display feature to view pulse jitter.....	5-8
5-4	Using the 7D20 ENVeloping mode to show waveform variations..	5-8
5-5	7D20 display showing safe operating areas and other information.....	5-12
5-5	Example of manufacturer's spec's.....	5-12

# LIST OF TABLES

Table No.		Page
1-1	Electrical Characteristics.....	1-6
1-2	Environmental Characteristics.....	1-11
1-3	Physical Characteristics.....	1-11
2-1	Control Settings at Power Up.....	2-81
2-2	Front Panel Settings that can be Stored and Recalled.....	2-84
2-3	7D20 DIGITIZING MODES.....	2-97
2-4	Digitizing Mode Bandwidth.....	2-98
2-5	Equivalent Time Digitizing Mode Trigger Requirements.....	2-103
2-6	Number of Points Averaged for Each Displayed Point in ROLL Mode.....	2-117
2-7	Number of Points Averaged for Each Max and Min Pair of Displayed Points in ROLL Mode.....	2-119
3-1	Summary of Time Base Modes.....	3-5
3-2	ETD Mode.....	3-10
3-3	7D20 Waveform Display Position.....	3-11
4-1	MEMORY DISPLAY Keypad Code.....	4-7
4-2	7D20 GPIB Interface Functions.....	4-8
4-3	Supplemental Characters.....	4-17
4-4	7D20 Status Bytes.....	4-26
4-5	Service Request Masks.....	4-31
4-6	Event Codes.....	4-36
4-7	7D20 Command Set.....	4-39
5-1	Effective Number of Points When Using ENV While ROLLING.....	5-6

# OPERATORS SAFETY SUMMARY

The general safety information in this part of the summary is for both operating and servicing personnel. Specific warnings and cautions will be found throughout the manual where they apply, but may not appear in this summary.

## TERMS

### IN THIS MANUAL

**CAUTION** statements identify conditions or practices that could result in damage to the equipment or other property.

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

### AS MARKED ON EQUIPMENT


**CAUTION** indicates a personal injury hazard not immediately accessible as one reads the marking, or a hazard to property including the equipment itself.

**DANGER** indicates a personal injury hazard immediately accessible as one reads the marking.


## SYMBOLS

### IN THIS MANUAL


 Static-Sensitive Devices.

 This symbol indicates where applicable cautionary or other information is to be found.

### AS MARKED ON EQUIPMENT

 **DANGER**—High voltage.

 Protective ground (earth) terminal.

 **ATTENTION**—refer to manual.

## WARNINGS

### POWER SOURCE

This product is intended to operate in a mainframe connected to a power source that will not apply more than 250 volts rms between the supply conductors or between either supply conductor and ground. A protective ground connection by way of the grounding conductor in the mainframe power cord is essential for safe operation.

### GROUNDING THE PRODUCT

This product is grounded through the grounding conductor of the mainframe power cord. To avoid electrical shock, plug the mainframe power cord into a properly wired receptacle before connecting to the product input or output terminals. A protective ground connection by way of the grounding conductor in the mainframe power cord is essential for safe operation.

### DANGER ARISING FROM LOSS OF GROUND

Upon loss of the protective-ground connection, all accessible conductive parts (including knobs and controls that may appear to be insulating), can render an electric shock.

### DO NOT OPERATE IN EXPLOSIVE ATMOSPHERES

To avoid explosion, do not operate this product in an atmosphere of explosive gases.

### DO NOT REMOVE COVERS OR PANELS

To avoid personal injury, do not remove the product covers or panels. Do not operate the product without the covers and panels properly installed.

### DO NOT OPERATE WITHOUT COVERS

To avoid personal injury, do not operate this product without covers or panels installed. Do not apply power to the plug-in unit via a plug-in extender.

# **DESCRIPTION**

The 7D20 Programmable Digitizer is designed to enhance the capabilities of 7000-series oscilloscope mainframes. Multiple waveform storage, crt readout, two cursors for point-to-point measurements, pre- and post-trigger viewing, store and recall of up to six front-panel settings, and signal averaging to reduce noise are just a few of the features this three-wide plug-in unit provides.

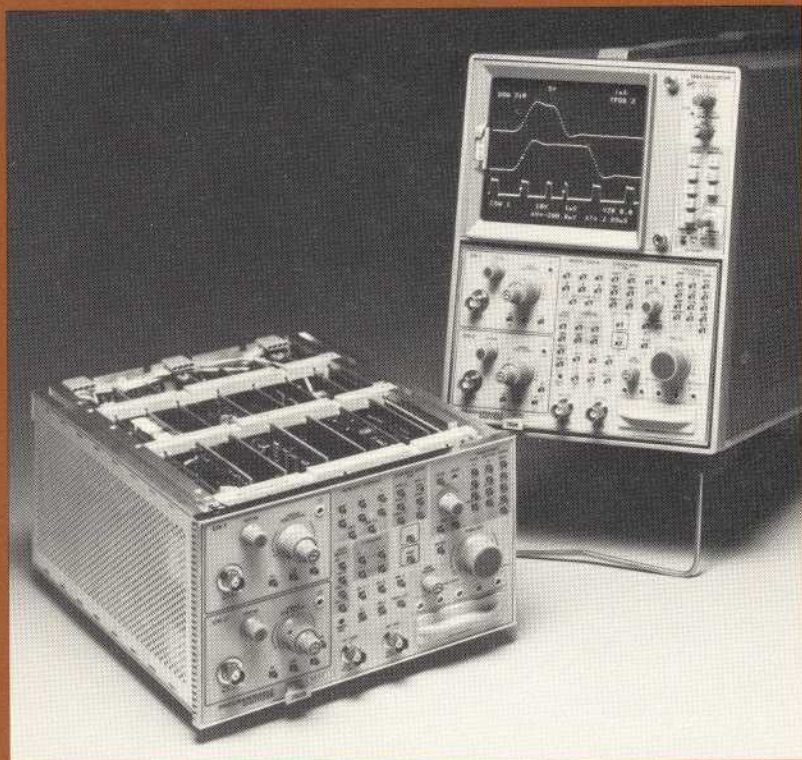
Waveform storage using digital memory eliminates the need for a storage crt, and allows viewing information that occurred prior to a triggering event. Six waveforms can be stored and displayed at the same time for easy comparison.

The 7D20 features a complete alphanumerical display of cursor waveform information and measurement values, time base and amplitude settings, trigger position, displayed waveform number, prompts and error messages, and a master menu that allows you to choose seldom used features easily and quickly.

The master menu offers a convenient way to enable special functions such as the STORE and RECALL of front-panel settings as well as allowing you to branch-out to other menus. The test menu is designed to assist in troubleshooting the 7D20 in the event of a failure. The SELF TEST may be implemented at any time to ensure user confidence. The other selections allow service personnel to diagnose faults to the component level.

Complete control of the 7D20's functions may be controlled via the IEEE-488 interface. Commands, waveforms, and alphanumeric text messages may be sent or received via the front-panel port.





This section is the first place to look for information concerning your 7D20 Programmable Digitizer. Here we describe the basic content of the operators and service manuals which are standard accessories to your new instrument. We also include a brief general description of the 7D20 and explain how to install it in a host mainframe for operation. In this section we give the electrical, environmental, and physical characteristics and specifications of the 7D20, list the standard and recommended accessories, and provide a dimensional drawing of the instrument. Should you have need to repackage the instrument for shipment, that information is also included.

# SECTION 1 CONTENTS

---

- DESCRIPTION..... 1-1
- ORANGE COLOR USED IN THIS MANUAL..... 1-1
- TECHNICAL MANUALS ..... 1-1
  - OPERATORS MANUAL..... 1-1
  - SERVICE MANUAL ..... 1-2
- INSTALLATION ..... 1-3
  - RACKMOUNT MAINFRAMES ..... 1-4
  - COMPATIBILITY..... 1-4
  - PACKAGING FOR SHIPMENT ..... 1-4
- SPECIFICATIONS..... 1-6
  - STANDARD ACCESSORIES..... 1-12
  - OPTIONAL ACCESSORIES (not included) ..... 1-12

# GENERAL INFORMATION

---

## DESCRIPTION

The 7D20 Programmable Digitizer is a three-compartment wide 7000-series plug-in. It provides simultaneous dual channel waveform digitizing and a variety of display features. Multiple digitizing modes include single shot, repetitive equivalent time, and roll modes. Programmability and GPIB talk-listen capability allows total or partial systemization and data transfer for more extensive signal processing. Display readout provides identification of waveforms, scale factors, prompting messages, and cursor measurements. Waveform storage is provided for six waveforms, which can be individually selected, positioned, expanded, compressed, magnified, or displayed in a Y versus X format. Digital storage allows multiple averaging and enveloping along with post and pretrigger capability.

## ORANGE COLOR USED IN THIS MANUAL

In using the operators manual, you will notice that some function names are printed in orange ink. The orange color indicates a shift function which requires that the *f* (shift function) key be pressed prior to selecting the functions.

## TECHNICAL MANUALS

Both an operators and a service manual are supplied with your 7D20 as standard accessories. The following information outlines the content of these manuals.

### Operators Manual

The operators manual for the 7D20 contains six sections of information to explain the operation of your instrument. The content of the operators manual is described below:

SECTION 1—GENERAL INFORMATION contains an instrument description along with information for installing the 7D20 into a host mainframe. We make a special note regarding rackmounting and another about compatibility considerations. This section also provides a complete list of the 7D20 electrical, environmental, and physical specifications and characteristics. It also lists standard and recommended accessories, and gives instructions for packaging the instrument for shipment.

SECTION 2—OPERATING INSTRUCTIONS provides a general overview of the instruments capabilities, controls and connector information, a series of get acquainted exercises, and detailed operating information.

**SECTION 3—OPERATIONAL THEORY** provides additional information about the operation of the 7D20 digitizer. This section discusses the more technical aspects of the digitizing techniques used by the 7D20.

**SECTION 4—GPIB** contains a general description of the GPIB interface, address selection information, description of the instrument status following power-up, and descriptions of the commands and messages which can be transferred over the GPIB. The section also provides a command language index.

**SECTION 5—APPLICATIONS** illustrates how your 7D20 can be used in a variety of situations. In this section we combine the 7D20 with other Tektronix products for some unique and interesting digital storage application possibilities.

**SECTION 6—INSTRUMENT OPTIONS** contains descriptions of available options.

## **Service Manual**

Your service manual contains eight sections of information pertaining to the servicing needs of your 7D20. The following is a brief overview of this manual's contents.

**WARNING**

*The following service instructions are for use by qualified personnel only. To avoid personal injury, do not perform any service other than that contained in operating instructions unless you are qualified to do so. Refer to Operators Safety Summary and Service Safety Summary prior to performing any service.*

**SECTION 1—GENERAL INFORMATION** contains an instrument description along with information for installing the 7D20 into a host mainframe. We make a special note regarding rackmounting and another about compatibility considerations. This section also provides a complete list of the 7D20 electrical, environmental, and physical specifications and characteristics. It also lists standard and recommended accessories, and gives instructions for packaging the instrument for shipment.

**SECTION 2—THEORY OF OPERATION** provides a basic block diagram description of the 7D20 along with general and specific circuit analysis that may be useful for servicing the instrument.

**SECTION 3—MAINTENANCE** describes routine and corrective maintenance procedures with detailed instructions for replacing assemblies, subassemblies, and individual parts. Included in this section are full instructions for troubleshooting the 7D20 using internal diagnostic routines and signature analysis.

**SECTION 4—CHECKS AND ADJUSTMENTS** is divided into three separate parts: Part I—Functional Check Procedure used to verify that the major functions of the

instrument perform properly. Part II—Performance Check Procedure used to verify that this instrument meets the applicable electrical specifications in Section 1. Part III—Adjustment Procedure provides an adjustment procedure to ensure that this instrument is performing at peak capabilities and meets or exceeds the listed electrical specifications at the time of adjustment under the specified conditions. These three parts provide for verification of the qualitative integrity of the product, its performance relating to specifications in Section 1, and the optimization of its performance respectively.

SECTION 5—INSTRUMENT OPTIONS contains descriptions of available options and provides a table for locating any option information found elsewhere within the manual.

SECTION 6—REPLACEABLE ELECTRICAL PARTS provides the information necessary to order replaceable parts and assemblies.

SECTION 7—DIAGRAMS AND CIRCUIT BOARD ILLUSTRATIONS includes detailed circuit schematics, locations of assembled boards within the instrument, voltage and waveform information and circuit board component locators. Signature analysis troubleshooting charts and tables also are included within this section.

SECTION 8—REPLACEABLE MECHANICAL PARTS provides information necessary for ordering mechanical parts and includes exploded-view drawings which identify individual parts and assemblies within the 7D20.

## INSTALLATION

The 7D20 is a three-compartment wide, Tektronix 7000-series plug-in unit. To install the 7D20 in a host mainframe, align the grooves in the top and bottom of the instrument with the guides at the top and bottom of the plug-in compartment of the mainframe. Then, push the 7D20 in until its front panel is flush with the front panel of the host mainframe. In four-compartment mainframes, the preferred position for installation is in the three right hand compartments. This leaves the left vertical compartment available for other 7000-series plug-in units.

Installation in any mainframe requires that the 7D20 display output be adjusted to compensate for the calibration tolerances of the host mainframe. The procedure for doing this is given in the Operating Instructions, Section 2.

### NOTE

*Switch off the mainframe power before installing or removing the 7D20.*

To remove the 7D20 from its host mainframe, pull the release latch (see Fig. 1-1) to disengage it from the mainframe. Pull straight out to remove the 7D20 from the plug-in compartment.

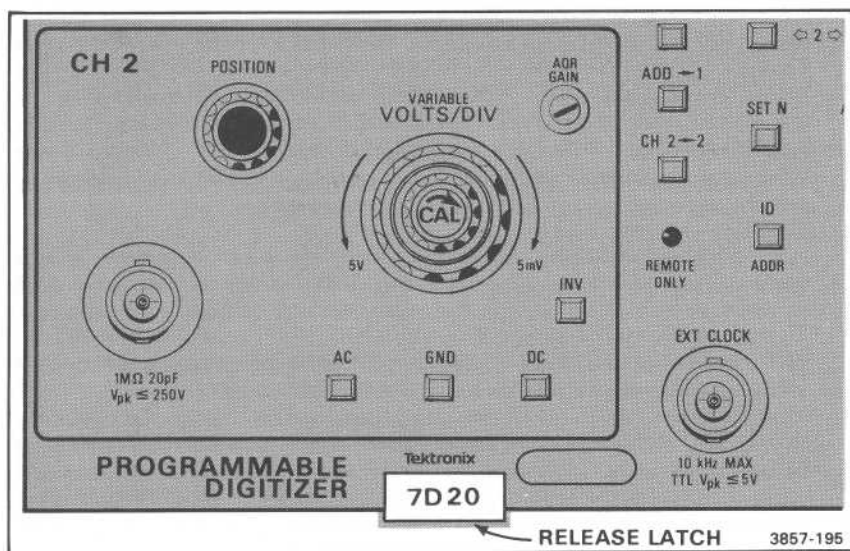


Figure 1-1. 7D20 Release Latch.

## RACKMOUNT MAINFRAMES

A special option (Option 20) is available for TEKTRONIX R7603 Oscilloscope (rackmount) mainframes that enable you to make GPIB connections at the rear of the mainframe. The option includes a special cable which must be installed in the 7D20 for this purpose. This cable is also available separately and can be ordered as an optional accessory.

## COMPATIBILITY

The 7D20 is compatible with all Tektronix 7000-series mainframes except the 7104. Because of the small area and fixed pattern of the 7D20 readout, the display from the 7D20 can cause permanent reduction in the crt microchannel plate gain, permanently reducing the writing rate of the 7104. Any crt damage caused by use of the 7D20 in the 7104 will not be covered under the instrument warranty.

## PACKAGING FOR SHIPMENT

If this instrument is to be shipped for long distances by commercial transportation, we recommend that it be packaged in the original manner. The carton and packaging material in which your instrument was shipped should be saved and used for this purpose.

Also, if this instrument is to be shipped to a Tektronix Service Center for service or repair, attach a tag to the instrument showing the following: Owner of the



instrument (with address), the name of the person at your firm who can be contacted, complete instrument type and serial number, and a description of the service required. Include any self test failure information you have available.

If the original packaging is unfit for use or not available, package the instrument as follows:

1. Obtain a corrugated cardboard shipping carton with a 200-pound test strength and having inside dimensions at least six inches greater than the instrument dimensions. This allows for cushioning.
2. Wrap the instrument with polyethylene sheeting or equivalent material to protect the finish of the instrument.
3. Cushion the instrument on all sides by tightly packing dunnage or urethane foam between the carton and the instrument, allowing three inches on each side.
4. Seal the carton with shipping tape or with an industrial stapler.
5. Mark the address of the Tektronix Service Center and your return address on the carton in one or more prominent locations.

# SPECIFICATIONS

The electrical characteristics listed in Table 1-1 apply when the following conditions are met: (1) Adjustment of the instrument must have taken place at an ambient temperature between +20° and +30° C, (2) the instrument and its host mainframe must be allowed a 20-minute warm-up period, (3) all specifications are valid at an ambient temperature of 0° to +45° C, unless otherwise stated, (4) the instrument must be in an environment that meets the limits described in Table 1-2.

Any applicable conditions not listed above are expressly stated as part of that characteristic. Environmental characteristics are listed in Table 1-2 and Physical characteristics are listed in Table 1-3.

**TABLE 1-1**  
**Electrical Characteristics**

Characteristic	Performance Requirement
<b>VERTICAL</b>	
Deflection Factor (Volts/Div)	
Calibration Range	5 mV/Div to 5 V/Div in 1,2,5 sequence.
Gain Ratio Accuracy	Within 2% with AQR GAIN adjusted at 10 mV/Div.
Uncalibrated (Variable)	Continuously variable; VARIABLE extends deflection factor to at least 12.5 VOLTS/DIV
Vertical Linearity (1 kHz square wave)	0.12 Div or less expansion or compression of a center screen 2 Div signal positioned anywhere within the ±4 division graticule area.
Invert Deflection Factor Ratio	1:1 within 1%.
Common Mode Rejection Ratio (using ADD, INVERT)	At least 10:1, dc to 50 MHz.
Bandwidth (1 μs to 50 ns/div)	Dc to 70 MHz.
AC Coupled Low-Frequency Bandwidth	10 Hz or less.

**TABLE 1-1 (CONT)**  
**Electrical Characteristics**

Characteristic	Performance Requirement
Risetime (ETD)	5 ns or less (6 div. signal centered on screen).
Overdrive Recovery	1 ms or less to recover within 1 div with the VOLTS/DIV set at 5 mV and an overdrive signal of $\pm 1.5$ V.
Maximum Input Voltage	
DC Coupled (DC+Peak AC)	250 V, 1 kHz or less.
AC Coupled (DC+Peak AC)	400 V, 1 kHz or less.
Input R and C	
Resistance	1 M $\Omega$ $\pm$ 1%.
Capacitance	Approximately 20 pF.
Signal Isolation between Channels (8 Div Ref)	100:1 up to 20 MHz.
Noise (Full scale = 10.24 Divisions)	Mean value of 50 measurements taken at 0.02 Division increments.
10 mV/Div to 5 V/Div	55 dB Full Scale/RMS Noise.
5 mV/Div	52 dB Full Scale/RMS Noise.
Phase Match Between Channels	2 Degrees at 10 MHz.

## TRIGGER

Sensitivity	Triggering Frequency Range	Minimum Vertical Signal Required		
		Internal	Ext.	Ext+10
AC Coupled	30 Hz—30 MHz	0.4 Div	60 mV	0.6 V
	30 MHz—70 MHz	1.0 Div	150 mV	1.5 V
AC LF REJ	50 kHz—30 MHz	0.4 Div	60 mV	0.6 V
	30 MHz—70 MHz	1.0 Div	150 mV	1.5 V
AC HF REJ	30 Hz—30 kHz	0.4 Div	60 mV	0.6 V

**TABLE 1-1 (CONT)**  
**Electrical Characteristics**

Characteristic	Performance Requirement			
	Triggering Frequency Range	Minimum Vertical Signal Required		
		Internal	Ext.	Ext÷10
DC HF REJ	DC—30 kHz	0.4 Div	60 mV	0.6 V
DC	DC—30 MHz 30 MHz—70 MHz	0.4 Div 1.0 Div	60 mV 150 mV	0.6 V 1.5 V
Max Signal		±6 Div	±1.0 V	±10 V
P-P AUTO	30 Hz—200 Hz 200 Hz—30 MHz 30 MHz—70 MHz	2.0 Div 0.6 Div 1.2 Div	300 mV 90 mV 200 mV	3.0 V 0.9 V 2.0 V
Programmed Trigger Levels				
Resolution		0.05 Div	7.8 mV	78 mV
Gain Accuracy		±5%	±8%	±12%
Offset		±0.25 Div	±50 mV	±500 mV
Range (Nominal)		±6.4 Div	±1.0 V	±10 V
Position				
Normal		-1500 to +10 Div (+0, -1 sample interval).		
ROLL ENV, ROLL AVE		0 to 10 Div (+0, -1 sample interval).		
Ext Trigger Input				
Resistance		1 MΩ, ±10%.		
Capacitance		Approximately 20 pF.		
Maximum Input Voltage (DC + Peak AC)		250 V (1 kHz or less).		

**TABLE 1-1 (CONT)**  
**Electrical Characteristics**

Characteristic	Performance Requirement		
<b>DIGITIZER</b>			
Acquisition Window	Nominally $\pm 5$ Div from center graticule line.		
Resolution Vertical	Nominally 0.04 Div.		
Horizontal	TIME/DIV	Points/Waveform	Resolution
	EXT, 20s—500 $\mu$ s	1024	0.01 Div.
	200 $\mu$ s—2 $\mu$ s	820	0.0125 Div.
	1 $\mu$ s—50 ns	1024	0.01 Div.
Stored Timing Accuracy 1 Cursor	$\pm 0.1\%$ of reading +0, -1 sample interval $\pm 300$ ps.		
2 Cursors	$\pm 0.1\%$ of reading $\pm 600$ ps.		
Digitizer Maximum Sample Rate	40 megasamples/second.		
Digitizing Modes			
Equivalent Time	1 $\mu$ s/div through 50 ns/div.		
Extended Real-Time	200 $\mu$ s/div to 2 $\mu$ s/div.		
Real-Time	50 ms/div to 500 $\mu$ s/div.		
Roll	20 s/div to 100 ms/div and EXTERNAL CLOCK.		
External Clock			
Input Frequency (Max)	$\leq 10$ kHz.		
TTL	$\leq 5$ Vpk.		

TABLE 1-1 (CONT)  
Electrical Characteristics

Characteristic	Performance Requirement
<b>GPIO INTERFACE</b>	
Functions Implemented (As Per IEEE 488-1978)	Description:
SH1	Complete Source Handshake.
AH1	Complete Acceptor Handshake.
T5	Talk Function Complete (no secondary addressing).
L3	Listen Function Complete (no secondary addressing).
SR1	Complete Service Request Capability.
RL1	Complete Remote/Local Capability.
PPO	No Parallel Poll Capability.
DC1	Complete Device Clear Capability.
DT1	Complete Device Trigger Capability.
C0	No Controller Capability.



**TABLE 1-2**  
**Environmental Characteristics**

<b>Characteristics</b>	<b>Information</b>
Temperature (External Ambient Mainframe Temperature)	
Operating	0 to +40°C in 7403N/7603 without fan. (fan kit is available) 0 to +45°C in other 7000-series mainframes.
Storage	-55° C to +75° C.
Altitude	
Operating	15,000 feet (4.6 Km).
Nonoperating	To 50,000 feet (15.2 Km).
Vibration	
Operating	Tested to MIL-T-28800C SECT. 4.5.5.3.1 Type 2, Class 5, Style E & F.
Shock	Tested to MIL-T-28800C SECT. 4.5.5.4.1 Type 2, Class 5, Style E & F.
Bench Handling	Tested to MIL-T-28800C SECT. 4.5.5.4.3 Type 2, Class 5, Style E & F.
Transportation	National Safe Transit Association, Preshipment Test Procedure.
Vibration and Bounce (packaged product)	NSTA, PROJECT 1 A-B-1.
Drop (packaged product)	NSTA, PROJECT 1 A-B-2.

**TABLE 1-3**  
**Physical Characteristics**

<b>Characteristics</b>	<b>Information</b>
Net Weight	Approximately 8.5 lb, 3.9 Kg.
Dimensions	See Figure 1-2.

**STANDARD ACCESSORIES**

- 1 ea ..... Operators Manual
- 1 ea ..... Service Manual

For more detailed information, refer to the tabbed Accessories page at the rear of the 7D20 Service Manual.

**OPTIONAL ACCESSORIES (not included)**

The following accessories are available for use with the 7D20. Refer to the tabbed Accessories page at the rear of the 7D20 Service Manual for more detailed information.

- 1 ea ..... Circuit-Board Extender
- 3 ea ..... 7000-series Plug-In Flexible Extenders
- 1 ea ..... Internal GPIB Cable for use with  
R7603 Option 20

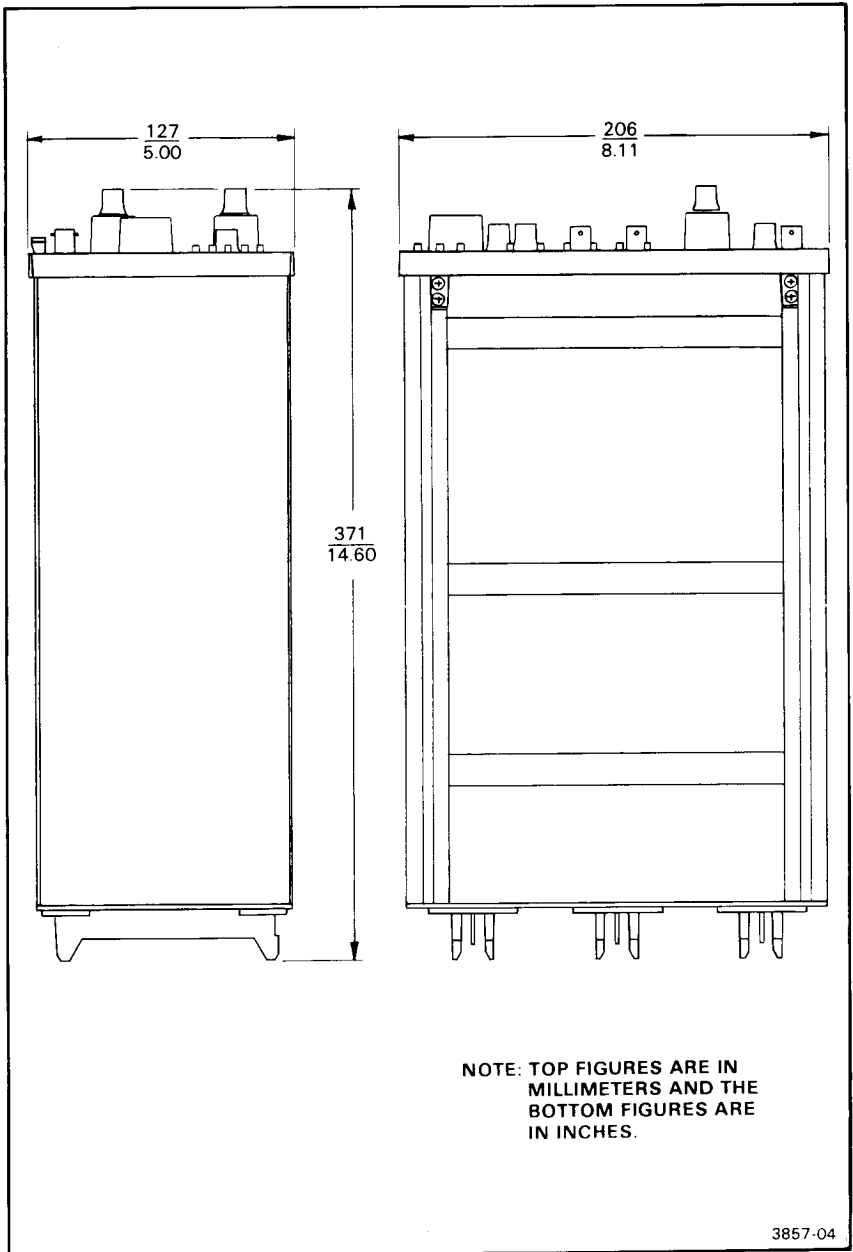
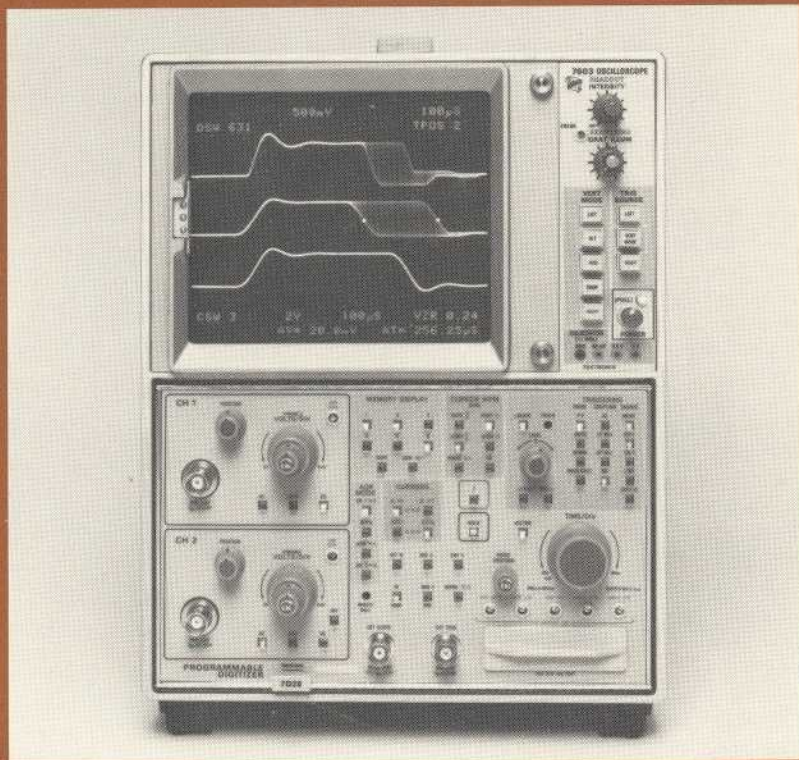


Figure 1-2. Dimensional Drawing.



This section will familiarize you with the capabilities and operation of the 7D20 Programmable Digitizer. A thorough understanding of this information will remove later uncertainty when operating your equipment.

Familiarization begins with illustrations and descriptions of all front-panel controls, connectors, and indicators followed by explanations of the crt readout fields generated by the 7D20.

A series of get-acquainted exercises for the first-time user is followed by detailed operating instructions which give you specific operational information for each function.

# SECTION 2 CONTENTS

---

CONTROLS AND CONNECTORS .....	2-1
DISPLAY .....	2-1
GET-ACQUAINTED EXERCISES .....	2-9
PRELIMINARY SET-UP .....	2-9
DISPLAY CALIBRATION .....	2-10
INITIALIZE .....	2-14
OBTAINING A WAVEFORM DISPLAY .....	2-16
EXERCISES .....	2-17
DETAILED OPERATING INSTRUCTIONS .....	2-81
POWER-UP .....	2-81
MENU FUNCTIONS .....	2-83
PROMPTS AND WARNINGS .....	2-88
READOUT DISPLAY .....	2-91
VERTICAL CONTROLS .....	2-92
ACQUIRE MODE .....	2-95
HOLD .....	2-96
<i>f</i> (shift function) .....	2-96
TIME/DIV .....	2-96
TRIGGERING .....	2-99
+SLOPE .....	2-103
TRIG'D .....	2-104
LEVEL .....	2-104
⇐TRIG POS ⇨ .....	2-104
HORIZ POSITION .....	2-105
VECTOR .....	2-105
MEMORY DISPLAY .....	2-105
CURSOR WFM .....	2-108
CURSORS .....	2-112
SET N .....	2-115
AVE .....	2-116
<i>f</i> , AVE N .....	2-118
ENV .....	2-118
<i>f</i> , ENV N .....	2-120
<i>f</i> , TEST .....	2-120
RQS (REQUEST SERVICE) .....	2-120
<i>f</i> RQS # .....	2-120
REMOTE ONLY (indicator) .....	2-120
ID (ADDRed Indicator) .....	2-121
EXT CLOCK .....	2-121
EXT TRIG .....	2-122
GPIB (connector) .....	2-122

# OPERATING INSTRUCTIONS

---

## OPERATION

For operation, your 7D20 Programmable Digitizer must be properly installed in a Tektronix 7000-series host mainframe. Installation is explained in the General Information section. Waveforms acquired by your 7D20 are viewed in the mainframe display, along with waveform identification, scale factors, cursor measurements, and prompting messages.

## CONTROLS AND CONNECTORS

All controls for operating your 7D20 Programmable Digitizer are located on the front panel of the instrument. Figure 2-1 shows the front panel and gives brief functional descriptions of the controls and connectors. You might find it useful to mark these pages for quick reference as you read the instructions in this section.

## DISPLAY

The 7D20 uses the host mainframe to display information. The readout associated with the 7D20 is generated within the 7D20, not in the readout circuitry of the mainframe. The 7D20 display, shown in Figure 2-2, gives the VOLTS/DIV and TIME/DIV settings along with information about displayed waveforms, the cursor waveform, and the cursor measurements. Also, PROMPTS and WARNINGS are generated and displayed. The top two lines of the readout gives current control settings, prompting, and warning messages. The two lower lines of the readout give cursor waveform information.

7D20 Controls & Connectors  
Foldout for  
Figure 2-1





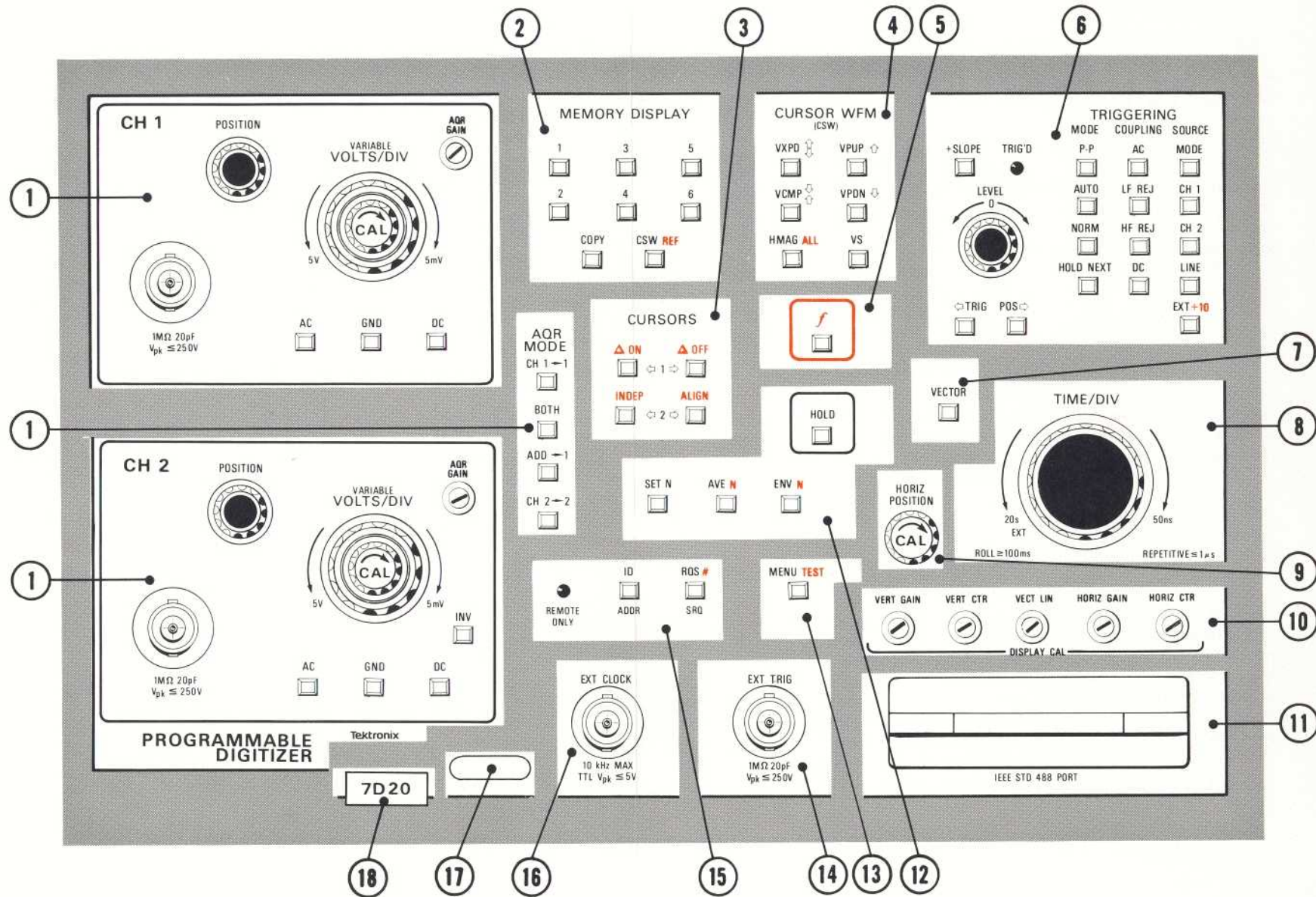


Figure 2-1. 7D20 Controls and Connectors.

**1 INPUT CONTROLS—CH 1 and CH 2**

**Input Connectors**—Bnc connectors for signal connection.

**POSITION**—Vertically positions the incoming signal during digitizing.

**VOLTS/DIV**—Selects calibrated vertical sensitivity factors. Volts/Div readout is displayed on the crt of the host mainframe.

**VARIABLE VOLTS/DIV**—Provides continuously variable uncalibrated settings between calibrated steps.

**AC, GND, DC**—These keys select the input coupling.

**AQR GAIN**—Screwdriver adjustment for adjusting input gain.

**INV (CH 2 only)**—Electrically inverts the signal connected to the CH 2 input.

**AQR MODE**—These keys select the method of input to the digitizer:

**CH 1→1**—Selects CH 1 for digitizing into waveform memory 1.

**BOTH**—Selects both channels for simultaneous digitizing into their respective waveform memories.

**ADD→1**—Causes input signals from both channels to be added together and digitized into waveform memory 1.

**CH 2→2**—Selects CH 2 for digitizing into waveform memory 2.

**2 MEMORY DISPLAY**

**KEYS 1-6**—These keys select their respective waveform memories for display and are also used for making selections from the various MENU's.

**COPY**—Allows you to copy waveforms from one memory to another.

**CSW**—Allows you to designate which waveform memory is the cursor waveform.

**CSW REF**—Shift function of the CSW key allows you to view the REFerence waveform when using HMAG and VS.

**3 CURSORS**

⇄1⇄—Cursor 1 movement keys.

⇄2⇄—Cursor 2 movement keys.

**△ON**—Shift function turns ON cursor 2.

**△OFF**—Shift function turns OFF cursor 2.

**ALIGN**—Shift function aligns cursors on all waveforms.

**INDEP**—Shift function allows the cursors on the cursor waveform to move independently of the other waveforms.

**4 CURSOR WFM (CSW)**

**VXPD**  $\begin{smallmatrix} \updownarrow \\ \updownarrow \end{smallmatrix}$ —Vertically expands the cursor waveform.

**VCMP**  $\begin{smallmatrix} \updownarrow \\ \updownarrow \end{smallmatrix}$ —Vertically compresses the cursor waveform.

**VPUP**  $\uparrow$ —Moves the cursor waveform up and is used as vertical separation when the reference waveform is present.

**VPDN**  $\downarrow$ —Moves the cursor waveform down and is used as vertical separation when the reference waveform is present.

**HMAG**—Horizontally magnifies the cursor waveform by a factor of 10.

**HMAG ALL**—Shift function horizontally magnifies all the waveforms by a factor of 10.

**VS**—Allows you to display the cursor waveform versus another waveform.

**NOTE**

*VXPD, VCMP, VPUP, VPDN keys operate on HMAG and VS display of the cursor waveform when the REFerence waveform is displayed although not in HOLD. These keys will also operate with waveforms 3 through 6 regardless of HOLD status. See Exercise 7 (Get-Acquainted Exercises) for more information.*

**5** *f*—Shift function key must be pressed prior to selecting any function printed in orange.



## 6 TRIGGERING

**+SLOPE**—Selects positive slope for triggering when illuminated. When not illuminated, -slope is used.

**LEVEL**—Adjusts the triggering level.

**TRIG'D**—Indicator illuminates when waveform being digitized is triggered.

**SOURCE**—These keys select the source of the trigger signal.

**COUPLING**—These keys select trigger coupling.

**MODE**—These keys are used to select the triggering mode, including HOLD NEXT.

◊ **TRIG POS** ◊—Adjusts the trigger position with respect to the display graticule for pretrigger and post trigger operation.

7 **VECTOR**—When illuminated connects adjacent dots with vectors.

8 **TIME/DIV**—Selects the digitizing rates and EXT mode. Control setting appears in the display.

9 **HORIZ POSITION**—Adjusts the horizontal position of the stored displayed waveforms.

10 **DISPLAY CAL**—These screwdriver adjustments calibrate the 7D20 display to match the host mainframe.

11 **IEEE STD 488 PORT**—Allows connection to GPIB.

## 12 SIGNAL PROCESSING

**HOLD**—Stops all digitizing and HOLDS the current waveforms in memory.

**AVE**—Causes the waveform(s) being digitized to be continuously averaged.

**ENV**—Causes the waveform(s) being digitized to be continuously enveloped.

**SET N**—Sets the value of N.

## SIGNAL PROCESSING (cont)

**AVE N**—Shift function causes the waveform(s) being digitized to be averaged N number of times then enters HOLD mode.

**ENV N**—Shift function causes the waveform(s) being digitized to be enveloped N number of times then enters HOLD mode.

13 **MENU**—Causes the 7D20 menu(s) to be displayed.

**TEST**—Shift function causes the 7D20 TEST menus to be displayed.

14 **EXT TRIG**—Bnc connector for connection of external trigger signal.

## 15 GPIB

**RQS**—Causes a service request to be asserted over the GPIB.

**RQS #**—Shift function permits selection of 6 unique SRQ event codes.

**ID**—Produces special GPIB menu for selection of system parameters.

**SRQ**—Key illuminates when service request is asserted over the GPIB.

**ADDR**—Key illuminates when 7D20 is addressed through GPIB.

**REMOTE ONLY**—When illuminated indicates that front panel is under GPIB control. Local operation locked out.

16 **EXT CLOCK**—Bnc connector for EXTERNAL CLOCK signal input.

17 **Instrument serial number.**

18 **Latching mechanism.** Pull to release from mainframe.

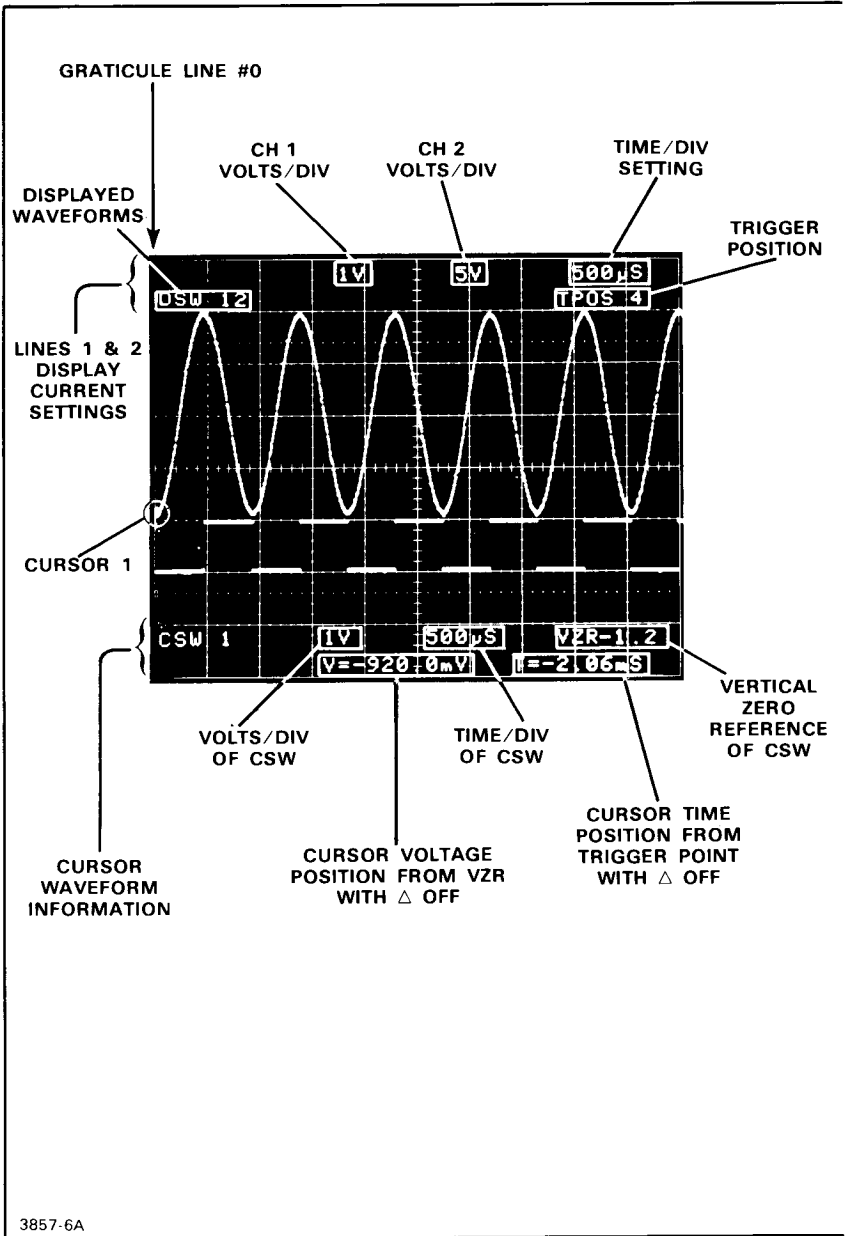
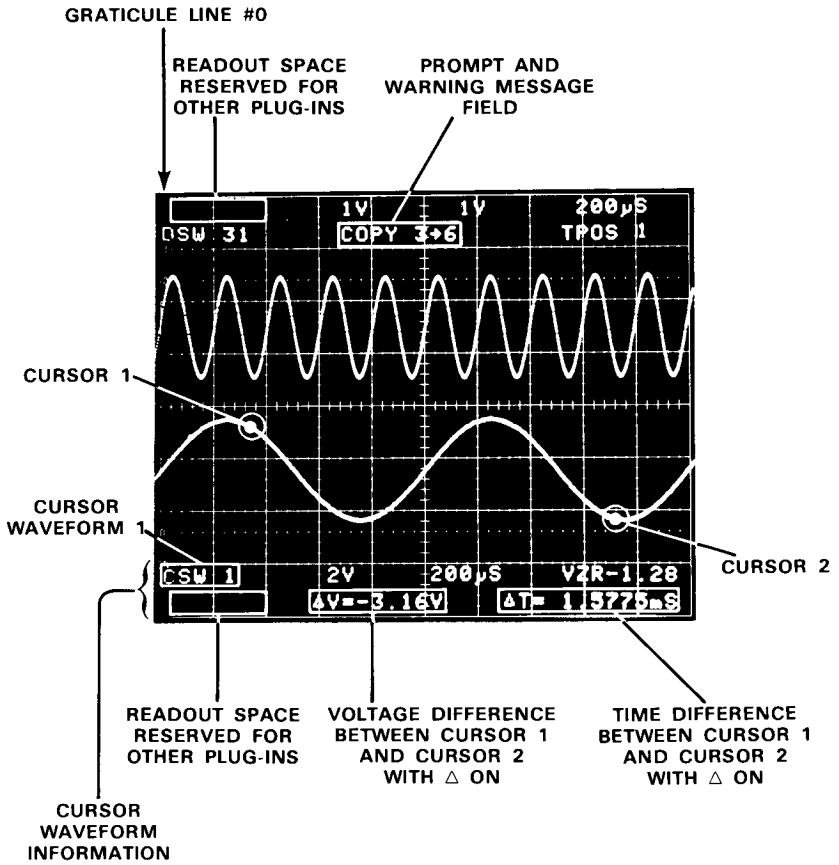


Figure 2-2. 7D20 Display.



3857-6B

Figure 2-2 (cont). 7D20 Display.

## FUNCTION KEYS

The 7D20 function keys allow you to perform a great number of different tasks. You will find that some of these keys perform simple on/off chores while others do some rather elaborate operations. Most keys illuminate when their respective function is active or selected; for example, if memory 3 is being displayed, MEMORY DISPLAY key 3 will stay illuminated. Some keys, like the COPY key will blink when it is pressed. This is an indication that the 7D20 is expecting you to COPY a waveform and is waiting for your selection. A blinking key function always is accompanied by a prompt message in the display readout except for the  $f$  key. Upon completion of such a process the key blinking will cease. The prompt message disappears when another key is used. Blinking key functions can be interrupted and thereby cancelled by selecting any function not called for by the prompt.

Slewing keys perform yet another kind of job. If you give a single press to one of these keys, for instance the  $1\Rightarrow$  (cursor 1, right movement key), cursor 1 will move to the next data point to the right. If you press and hold the same key, cursor 1 will begin continuous and increasingly more rapid movement to the right. Of course, cursor 1 will stop when it reaches the limit of its travel.

No doubt you've noticed that some keys are labeled with an orange color. This indicates that the function thus labeled is a shift function and requires that you first press the  $f$  (shift function) key before that function can be selected or turned off.

## ERROR MESSAGES

Some functions can be selected or used only when certain conditions are met. Attempts to do otherwise will result in a warning message being displayed in the readout. An example of such would be when you try to use the VPUP $\uparrow$  (Vertical Position UP) key when the cursor waveform is not in HOLD. A complete list of WARNINGS and their causes appears in the Detailed Operating Instructions later in this section.

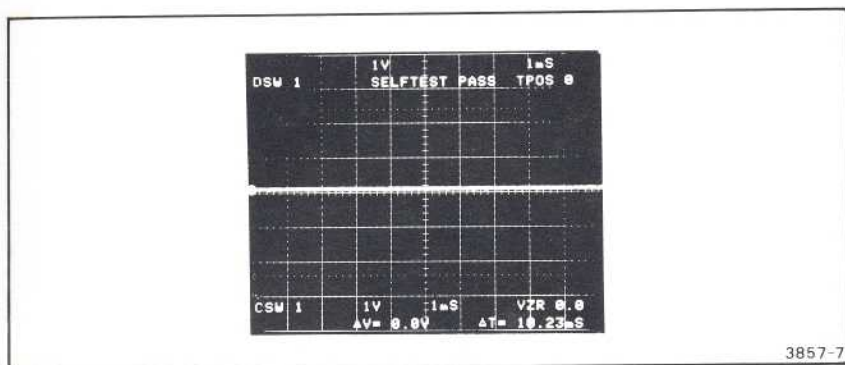
## GET-ACQUAINTED EXERCISES

The following pages contain a series of exercises that will acquaint you with the front panel (local) operation of your 7D20. Before you start you should carefully review the CONTROLS, CONNECTORS and DISPLAY information given in Figures 2-1 and 2-2.

In general, the 7D20 Programmable Digitizer allows you to digitize waveform signals through two input channels without compromising the record length and the sampling rate. These inputs can be digitized individually or simultaneously. The analog signal information is clocked through CCDs (charge coupled devices) then converted to digital information and stored in the appropriate memory of the 7D20. Waveform data is always read from the 7D20 memory for display. The 7D20 contains six memories for waveform storage and all six can be displayed individually, in any combination. Further, the cursor waveform (which can be any of the six) can be vertically expanded or compressed, horizontally magnified, positioned up or down in the display, and displayed versus another waveform. The following exercises illustrate how these functions operate.

### PRELIMINARY SET-UP

Before you begin the Get-Acquainted Exercises there are a few preliminary considerations. First, your 7D20 must be properly installed in a Tektronix 7000-series host mainframe as described in Section 1—General Information. When using the 7D20 in 4-compartment mainframes, the preferred installation position is in the three right compartments since this keeps the left vertical position available for using other 7000-series plug-ins such as the 7D11 Digital Delay Unit, the 7D15 Counter/Timer, or the 7D13A Digital Multimeter. Vertical plug-in units can also be used in this configuration by connecting to the VERT SIG OUT connector on the rear of the host mainframe. The Applications Section 5 gives an example of using the 7A13 Amplifier unit with the 7D20. Readout display space is reserved for use with other plug-ins as shown in Figure 2-2. Set the 4-compartment mainframe to select the right vertical channel and B time base. Set the 3-compartment mainframe to select the left vertical channel. Turn on the host mainframe power switch to power-up the 7D20. Adjust the mainframe controls for a good viewable display. During the first few seconds after you turn it on, the 7D20 automatically goes through a series of internal self-tests. The display will flash and the front panel keys will illuminate during the self test period. If all is well at the conclusion of its self-test "SELFTEST PASS" will appear in the mainframe display as shown in Figure 2-3. If all is not well the 7D20 tells you where to look for the problem. Refer to the Detailed Operating Instructions for more information about self test.

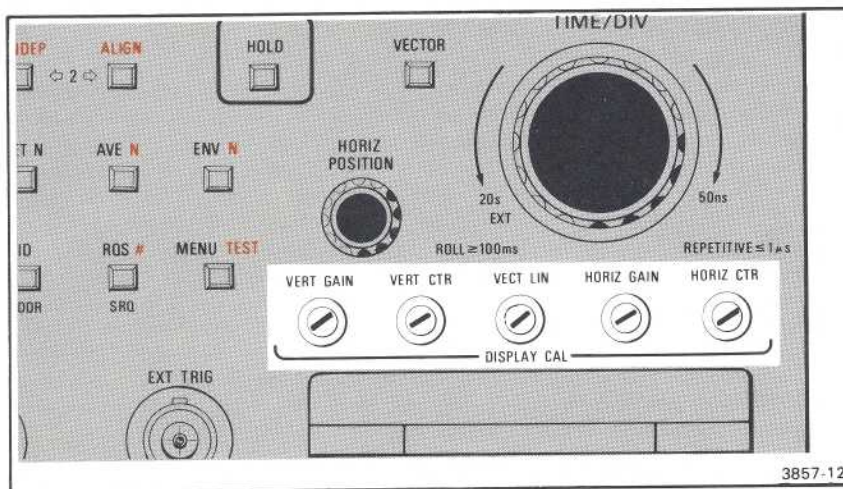


3857-7

Figure 2-3. Power-up SELFTEST PASS display.

## DISPLAY CALIBRATION

Assuming that this is the first time your 7D20 has been operated in the mainframe in which you now have it installed, it is necessary at this point to adjust the 7D20 to compensate for the calibration tolerance of the host mainframe using the adjustments shown in Figure 2-4. These operator adjustments affect only the 7D20 display and in no way affect its digitizing accuracy or abilities. Allow the 7D20 and the host mainframe to warm-up for about 20 minutes to ensure that the temperature has stabilized before performing the following procedure.



3857-12

Figure 2-4. DISPLAY CAL adjustments.

## DISPLAY ADJUSTMENT PROCEDURE

**Step 1** Press MENU, then press MEMORY DISPLAY key 3 to select the DISPLAY CAL PATTERN from the MASTER MENU (Fig. 2-5). Once the DISPLAY CAL PATTERN is selected the display will appear similar to that shown in Figure 2-6. You may find it necessary to adjust the TRACE ROTATION control on the host mainframe to make the vertical and horizontal lines of the DISPLAY CAL PATTERN parallel with those of the display graticule.

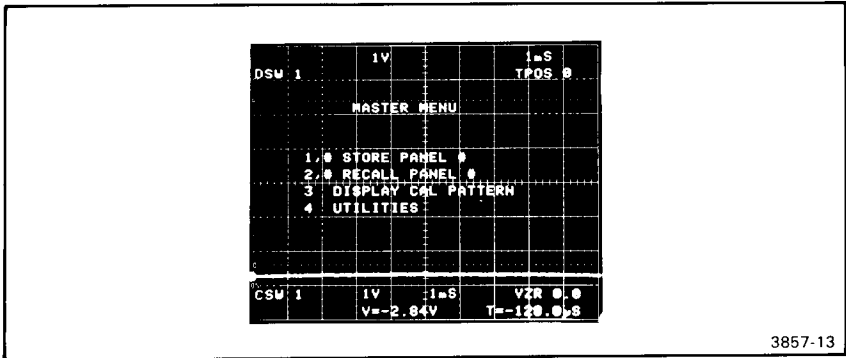


Figure 2-5. 7D20 MASTER MENU.

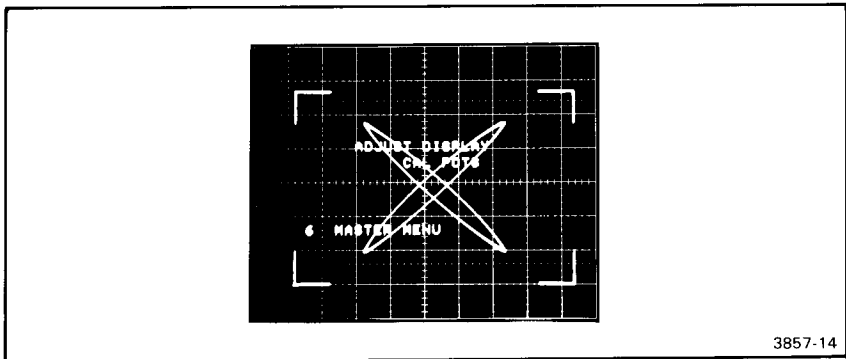


Figure 2-6. 7D20 DISPLAY CAL PATTERN.

**Step 2** Adjust the HORIZ CTR (horizontal center) control to bring the left edge of the DISPLAY CAL PATTERN in alignment with graticule line 1 of the display as shown in Figure 2-7 (left-most graticule line is 0).

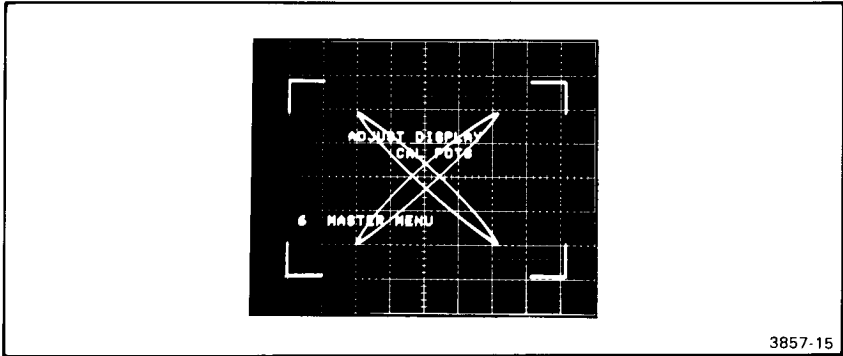


Figure 2-7. Adjust HORIZ CTR.

**Step 3** Adjust the HORIZ GAIN (horizontal gain) control to bring the right edge of the DISPLAY CAL PATTERN into alignment with vertical graticule line 9 of the display as shown in Figure 2-8. These adjustments are interactive and you may need to work back and forth between them to optimize pattern and graticule alignment.

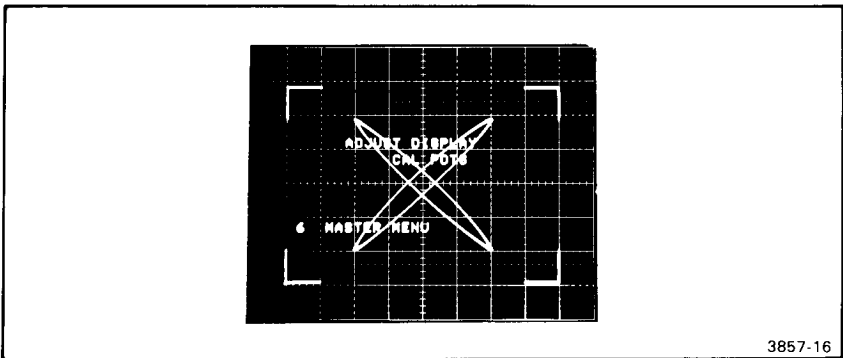


Figure 2-8. Adjust HORIZ GAIN.



**Step 4** Adjust the VERT CTR (vertical center) and VERT GAIN (vertical gain) controls to align the upper and lower horizontal edges with horizontal lines 1 and 7 of the display graticule as shown in Figure 2-9 (bottom graticule line is 0). As with the HORIZ CTR and HORIZ GAIN controls, these vertical adjustments are interactive and may require that you work back and forth between them to optimize this adjustment.

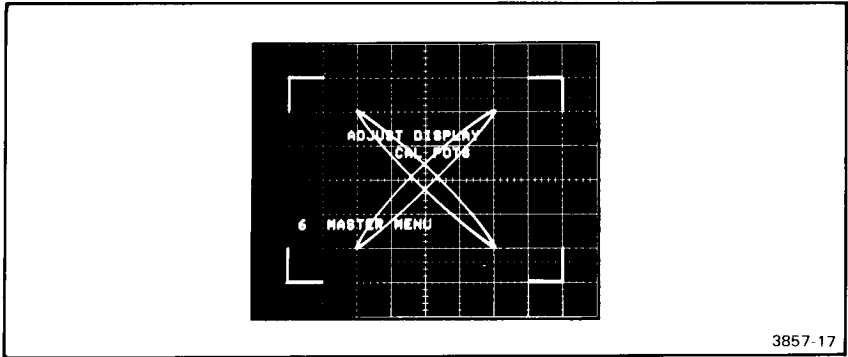


Figure 2-9. Adjust VERT CTR and VERT GAIN.

**Step 5** Adjust the VECT LIN (Vector Linearity) control to close the crossing pattern in the DISPLAY CAL test pattern as shown in Figure 2-10. This will optimize the vector display by compensating for the delay line length of the host mainframe.

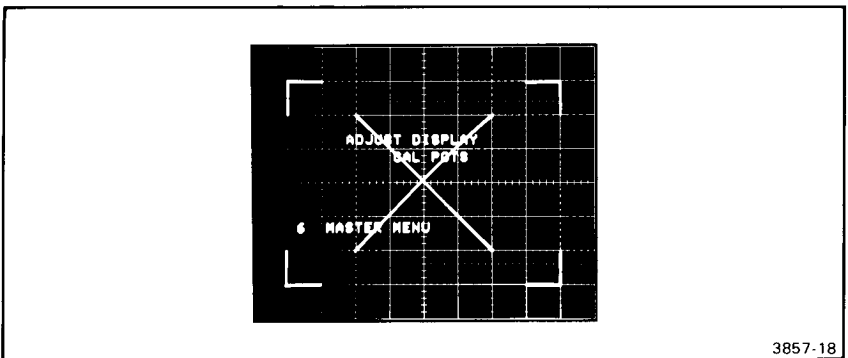


Figure 2-10. Adjust VECT LIN.

**Step 6** When you are satisfied the display is properly aligned with the graticule and the vector display is adequately compensated as shown in Figure 2-11, press MEMORY DISPLAY key 6 to return the MASTER MENU to the display. Press the MENU key to extinguish the MENU display and proceed with the following instructions.

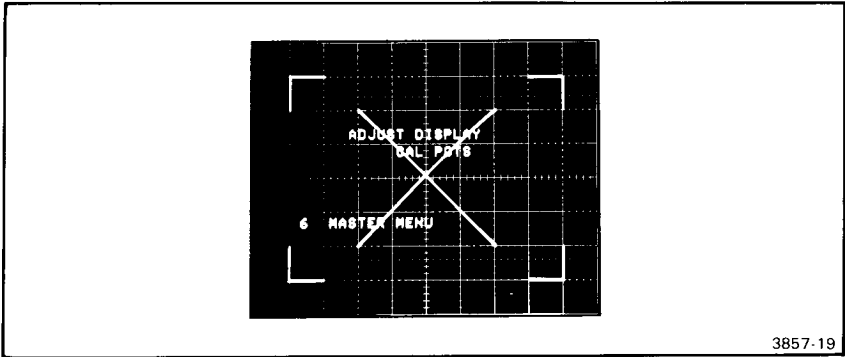
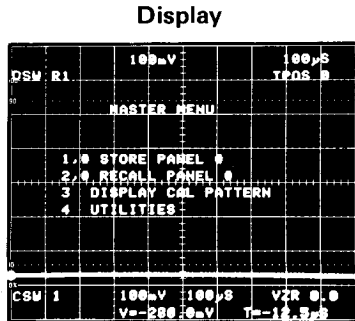


Figure 2-11. Correctly adjusted DISPIAY CAL PATTERN.

## INITIALIZE

When the 7D20 is powered up, it will then re-establish the control settings it remembers from when it was last turned off. Ignore these settings and use the INITIALIZE routine included in the 7D20 MENU especially for these "Get-Acquainted Exercises". A full list of the INITIALIZE settings is given in the Detailed Operating portion of this section. To INITIALIZE, proceed with the following:

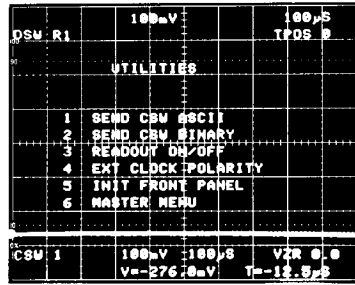
Press	Remarks
MENU	The key illuminates and the MASTER MENU appears in the display.



3857-8

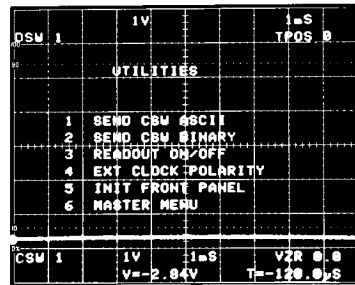
- Press**                      **Remarks**
- 4                      Selects the UTILITIES menu which appears in the display.

**Display**



3857-9

- 5                      Selects the INIT FRONT PANEL routine from the UTILITIES menu and initializes the 7D20. On power up, the 7D20 remembers its control settings from when it was last turned off. Initialize sets the 7D20 to a known state for the purpose of our "Get-Acquainted Exercises."



3857-10

- MENU**                      Turns off the displayed UTILITIES menu. The MENU key will extinguish.

**NOTE**

See "Menu Functions" in the Detailed Operating Instructions for a comprehensive list of the INIT FRONT PANEL predetermined control settings.

## OBTAINING A WAVEFORM DISPLAY

After initialization your 7D20 is set to acquire signals at the channel 1 input connector and to display the contents of memory 1. Initialization also selects the waveform in memory 1 as the cursor waveform, which simply means the waveform that has the cursors on it. In the 7D20, the cursor waveform is always displayed.

A sine wave will be used for most of the Get-Acquainted Exercises so adjust your sine-wave generator to output a 1-kHz, 4-volt signal and connect this signal to the CH 1 input connector of the 7D20. The signal should be visible in the display as soon as you have it connected. Use the CH 1 POSITION control to vertically center the displayed sine wave about the center graticule line and make sure the CAL (variable volts/div) control is in its calibrated detent position. The HORIZ POSITION control also has a calibrated detent position which horizontally centers the waveform. Make sure this control is also in its detent position. If necessary, adjust the signal generator output amplitude for 4 divisions of signal display. After proper adjustment the display should appear as shown in Figure 2-12.

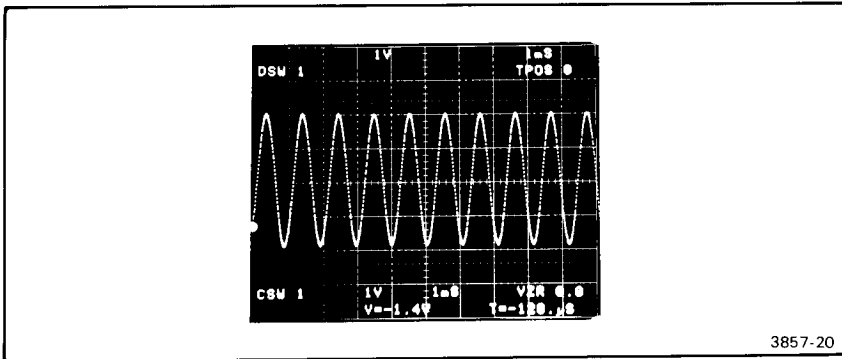


Figure 2-12. Sine-wave display.

## EXERCISES

The exercises on the following pages will acquaint you with the basic, front panel (local) operation of your 7D20 Programmable Digitizer. Each exercise begins with the sine-wave signal displayed exactly as at the end of the preliminary set-up instructions which you have just completed. You can go from this initialized point to the beginning of any of the following Get-Acquainted Exercises.

## INDEX

1. USING THE VERTICAL CONTROLS.....	2-17
2. USING THE AQR (ACQUIRE) MODE .....	2-22
3. USING THE WAVEFORM MEMORY .....	2-24
4. CHANGING THE DESIGNATED CURSOR WAVEFORM .....	2-28
5. HOLDING A WAVEFORM IN MEMORY .....	2-31
6. USING THE CURSORS .....	2-32
7. USING THE CURSOR WAVEFORM DISPLAY MODIFIER KEYS .....	2-42
8. USING THE AVERAGE AND ENVELOPE FUNCTIONS AND SET N .....	2-67
9. DIGITIZING MODES AND THE TIME/DIV CONTROL .....	2-72
10. SETTING THE TRIGGER POSITION.....	2-74
11. USING THE HOLD NEXT .....	2-77
12. USING THE EXTERNAL CLOCK.....	2-79

## EXERCISE 1 USING THE VERTICAL CONTROLS

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

This exercise demonstrates the controls associated with the 7D20 vertical input functions shown in Figure 2-13. Since channel 1 and channel 2 vertical controls function identically (except for the channel 2 INV function), only the controls for channel 1 will be discussed. Before you begin, make sure the 7D20 is "INITIALIZED" and the 1 kHz sine wave is connected to the CH 1 input connector as described in the PRELIMINARY SET-UP INSTRUCTIONS, then proceed with the following exercises.

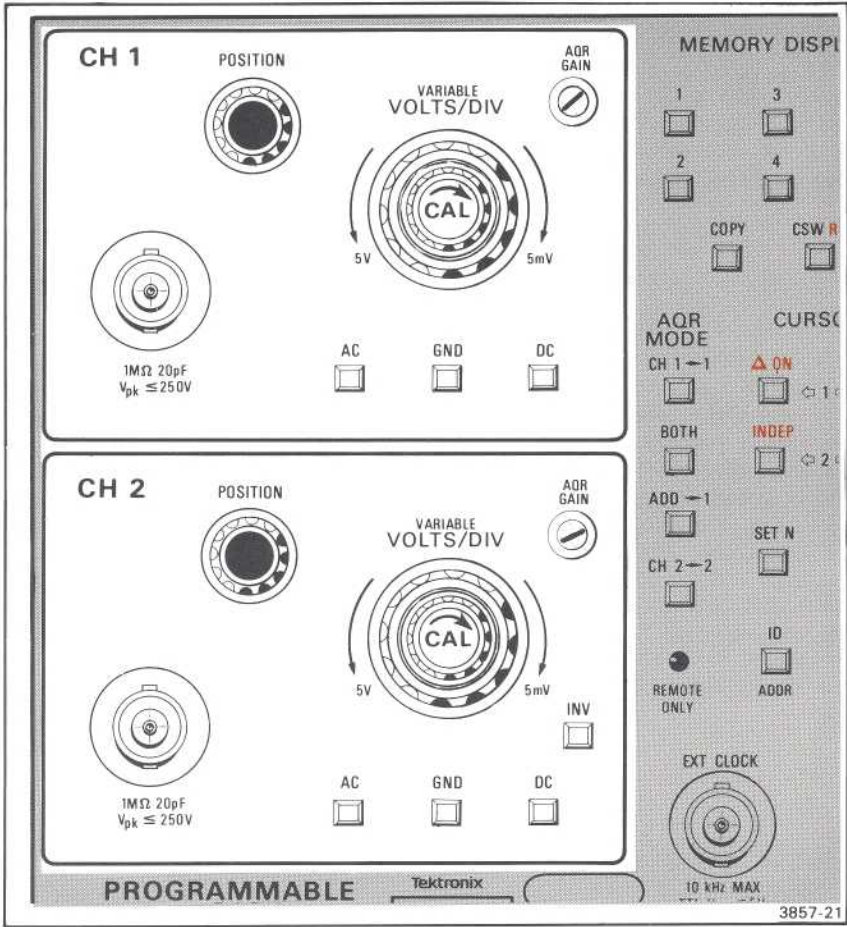


Figure 2-13. Vertical controls.

## SETTING THE INPUT COUPLING

The 7D20 allows three choices for setting the input coupling.

Press	Remarks
AC DC GND	These keys select the input coupling for CH 1. Initialization sets the 7D20 for AC coupling so the AC key is illuminated when you begin. Alternately press these keys and notice that the pressed key illuminates and the previously illuminated key extinguishes. These keys are mutually exclusive so only one can be selected at a time. Notice too as you alternately press these keys, the effect that each has on the displayed waveform. The next exercise shows how to use GND input coupling to set the VZR (vertical zero reference).

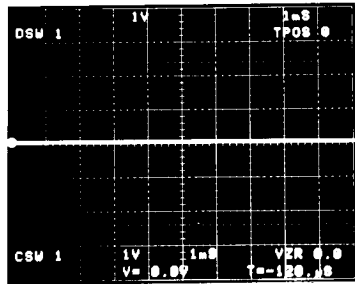
## HOW TO SET THE VZR (Vertical Zero Reference)

The 7D20 uses cursors to make measurements on the waveforms. Cursor 1 is always on but cursor 2 can be turned off or on as needed. In order for the cursors to be accurate they must be referenced to ground or zero volts. This is called the vertical (zero) reference level, VZR. The VZR readout indicates where the zero level is in divisions from center screen.

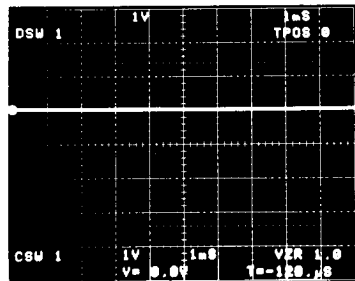
Upon initialization, the VZR is established to be the center of the 7D20 display. You can change this approximately 5 divisions (plus or minus) from the center position as shown below.

Press	Remarks
GND	This references the input signal to ground and causes a horizontal line to be displayed across the display. Adjust the CH 1 POSITION control and notice that the displayed line can be positioned up or down in the display area. Also notice that the VZR readout changes as the line is repositioned. Although the VZR range is $\approx \pm 5$ divisions, the POSITION control can provide $\pm 9$ divisions of signal offset when in AC or DC. Position the line one full division above the center line and observe that the VZR readout reads 1 division.

Display

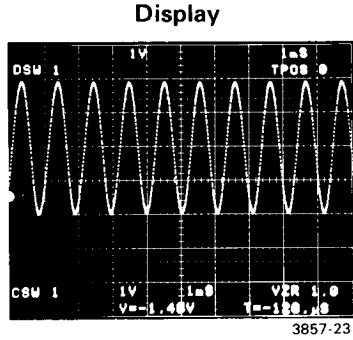


3857-22



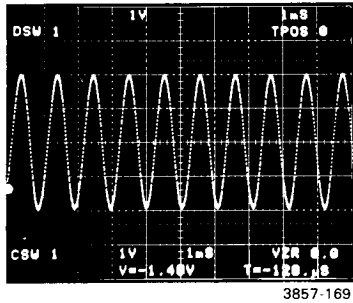
3857-34

Press	Remarks
AC	Again selects AC input coupling and causes the incoming sine wave to be digitized and displayed. Now, however, the displayed sine wave is centered one full graticule line above the center of the display graticule as shown. This is the new VZR level as indicated in the VZR readout. Notice that turning the CH 1 POSITION control no longer changes the VZR level.



GND	When you press GND this time you'll notice the VZR readout changes again. Because you moved the CH 1 POSITION control in the preceding step, you also changed the zero reference.
-----	---

AC	This establishes the VZR at the last position of the CH 1 POSITION control. Any time you want to change it, use the POSITION control to position the incoming waveform, press GND, then press AC (or DC) again. Return the VZR level to the center graticule line.
----	--



To set the VZR reference at any time simply press GND, then return to either AC or DC. The POSITION control should not be changed without repeating this process.

**NOTE**

*The VZR value is only used by the single vertical cursor readout and to establish a reference for data transferred over the GPIB.*



## EXERCISE 2 USING THE AQR (ACQUIRE) MODE

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

These four keys (see Fig. 2-15) control the way your 7D20 acquires signals for digitizing. These are mutually exclusive functions so only one can be selected at any one time. Upon initialization, the 7D20 is set to acquire signals present at the CH 1 input connector as indicated by the illumination of the CH 1 AQR MODE key. Connect the 1 kHz 4 V P-P sine wave to channel 1 and connect the trigger output (square wave) from your signal generator, if so equipped, to channel 2. If a trigger signal is not available, choose some other time related signal. Initialize the 7D20 using the UTILITIES menu. Select 2 volts/div on Channels 1 and 2.

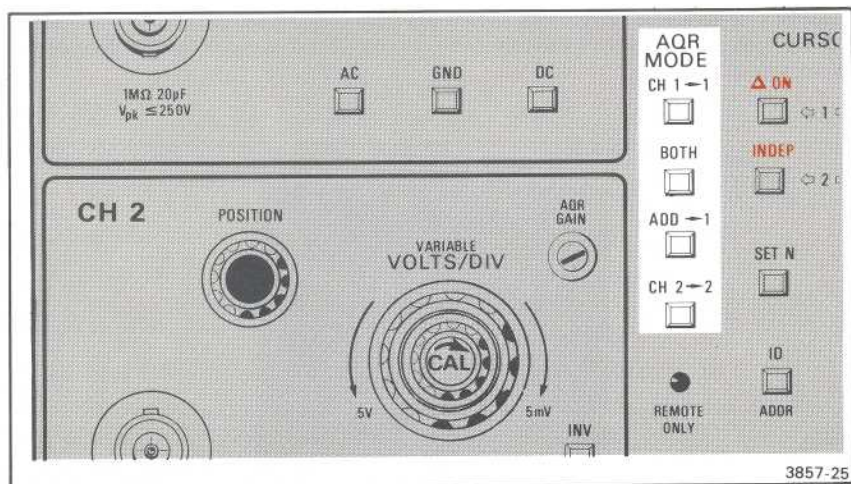


Figure 2-15. AQR MODE keys.

## CHANGING THE VERTICAL SENSITIVITY (VOLTS/DIV)

Initialization sets the vertical sensitivity to 1 volt/div as indicated in the display readout. The VOLTS/DIV control allows you to select the vertical sensitivity of the 7D20 in volts per division. Slowly rotate the VOLTS/DIV control and observe the effect this has on the displayed waveform. Notice too that each change is indicated in the top line of the display readout. As the VOLTS/DIV control is rotated in the counterclockwise direction the Vertical sensitivity can be decreased to a minimum of 5 V/div. Rotation of the control in a clockwise direction increases the vertical sensitivity, in a 1, 2, 5 sequence to a maximum of 5 mV/div. Rotating the VOLTS/DIV control beyond either the minimum or maximum setting will have no further effect on the vertical sensitivity or the displayed sensitivity setting. Return the vertical sensitivity to 1 volt/division.

## USING THE VARIABLE VOLTS/DIV CONTROL

This control allows you to decrease the vertical sensitivity from its current setting. Rotate the knob counterclockwise (out of its calibrated detent position) and notice that a > sign appears in the display readout next to the VOLTS/DIV setting as shown in Figure 2-14. This indicates that the vertical attenuation is now uncalibrated and is greater than the displayed VOLTS/DIV setting. Observe that the vertical sensitivity continues to decrease as you rotate the VARIABLE VOLTS/DIV control and notice the effect this has on the displayed waveform. Return the control (by rotating clockwise) to its detent position and note that the > symbol disappears from the display.

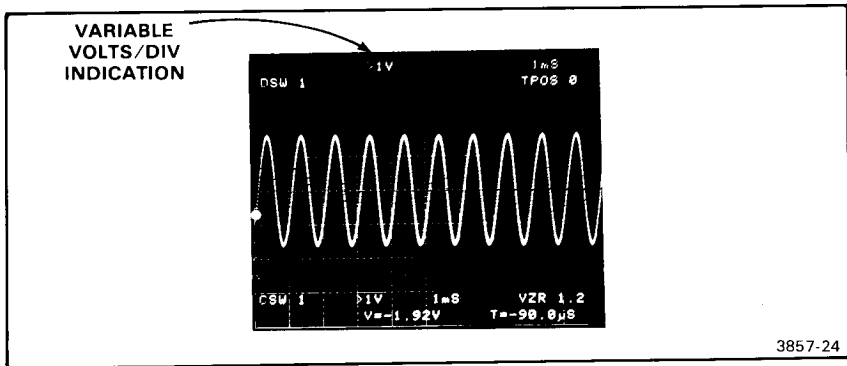


Figure 2-14. VARIABLE VOLTS/DIV readout.

## EXERCISE 2 USING THE AQR (ACQUIRE) MODE

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

These four keys (see Fig. 2-15) control the way your 7D20 acquires signals for digitizing. These are mutually exclusive functions so only one can be selected at any one time. Upon initialization, the 7D20 is set to acquire signals present at the CH 1 input connector as indicated by the illumination of the CH 1 AQR MODE key. Connect the 1 kHz 4 V P-P sine wave to channel 1 and connect the trigger output (square wave) from your signal generator, if so equipped, to channel 2. If a trigger signal is not available, choose some other time related signal. Initialize the 7D20 using the UTILITIES menu. Select 2 volts/div on Channels 1 and 2.

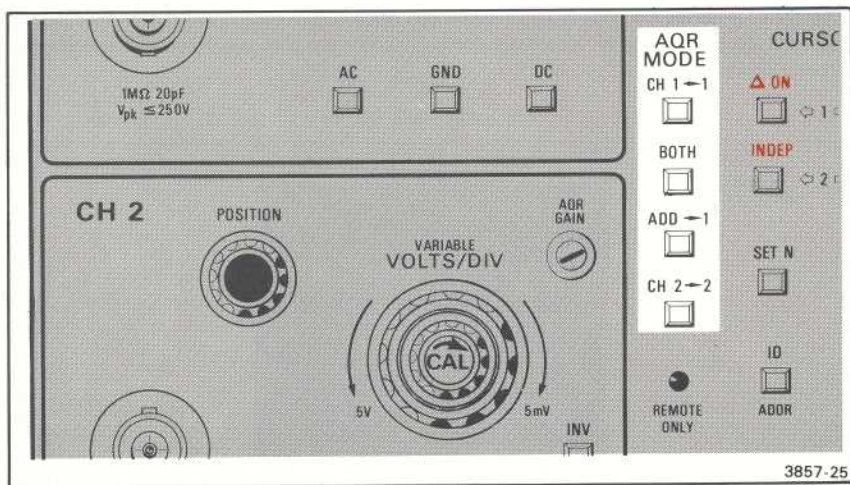


Figure 2-15. AQR MODE keys.

Press	Remarks	Display
CH 1→1	Selects channel 1 for digitizing and stores the digitized waveform in waveform memory 1. Position the sine wave into the top half of the display using the CH 1 POSITION control.	<p style="text-align: right;">3857-48</p>
BOTH	Selects both channels (CH 1 and CH 2) for digitizing and stores the digitized waveforms in waveform memories 1 and 2 respectively. Position the square wave to the lower half of the display using the CH 2 POSITION control.	<p style="text-align: right;">3857-141</p>
ADD	Selects the ADD AQR MODE illuminating the ADD key and causes both inputs (channels 1 and 2) to be added together and the result stored in waveform memory 1. Notice the result of adding the square wave with the sine wave.	<p style="text-align: right;">3857-120</p>
CH 2→2	Selects only the channel 2 input for digitizing. The key illuminates upon selection and the waveform present at the CH 2 input connector is digitized into waveform memory 2. Notice that the trigger source of the time base has been automatically changed to channel 2. This will occur any time MODE has been selected as the trigger source.	<p style="text-align: right;">3857-121</p>

## EXERCISE 3 USING THE WAVEFORM MEMORY

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

Your 7D20 contains six memory registers devoted to the storage of waveform data. MEMORY DISPLAY keys are shown in Figure 2-16. Waveforms are digitized into memories 1 and 2 and can be copied from these memories to any of the others. The memories can be displayed or not as you wish, **except that one of the six memories must be designated the cursor waveform memory and is always**

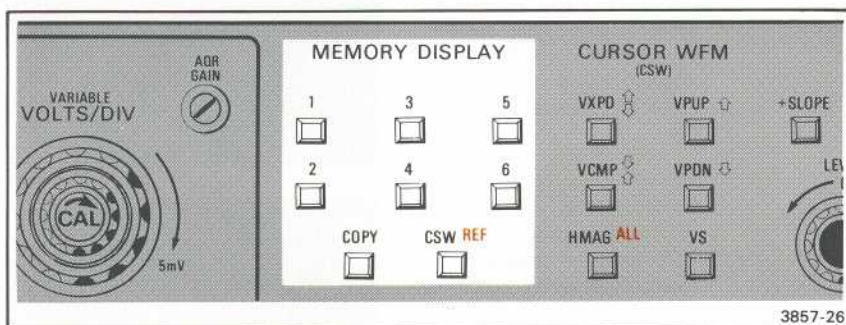


Figure 2-16. MEMORY DISPLAY keys.

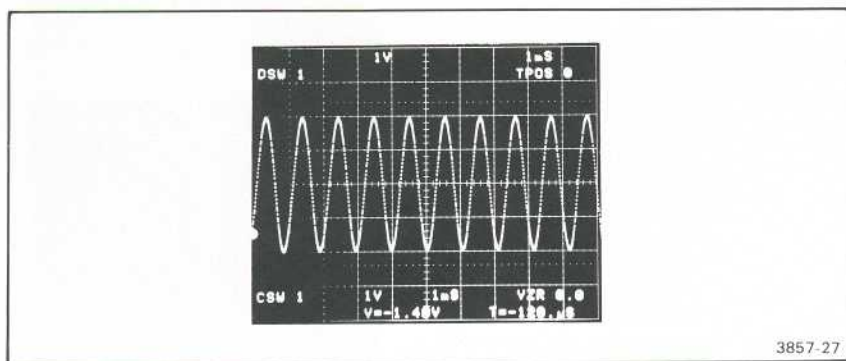


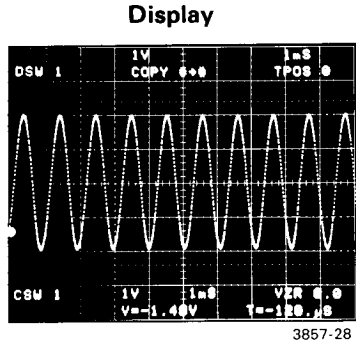
Figure 2-17. Sine-wave display.

**displayed.** Initialization sets the 7D20 to digitize signal information into waveform memory 1 and also establishes waveform memory 1 as the cursor waveform. The sine-wave signal is currently visible in the display (see Fig. 2-17) and MEMORY DISPLAY key 1 illuminated. Because of initialization, waveform memory 1 is the only memory that contains waveform data at this time. The other memories (2 through 6) have no stored data and, if selected for display, will show a center horizontal line.

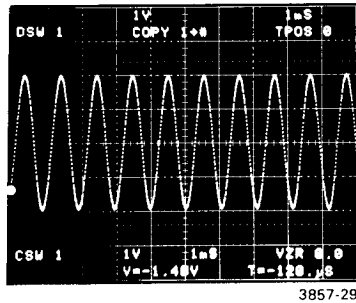
### COPYING AND DISPLAYING A WAVEFORM

Waveforms can be copied from one memory to any other so long as you are not digitizing into the receiving memory. After initializing, because data is acquiring waveform memory 1, you cannot copy into waveform memory 1 at this time (the HOLD function terminates the digitizing process). For now, however, the sine-wave signal being digitized into channel 1 can be copied from waveform memory 1 into any of the other memories. When a waveform is copied, all the acquire and display conditions are copied too, the memory being copied into is overwritten and cleared of any previous data. The following shows how to copy a waveform into another memory and then display the copied waveform. Initialize the 7D20 then:

Press	Remarks
COPY	The COPY key will begin to blink and a prompt appears in the display. The left most # sign in the prompt also will blink. A blinking key or prompt indicates that a command requiring additional keypushes is in process. The 7D20 is currently waiting for you to enter the number of the waveform memory you wish to copy.



1	Tells the 7D20 that you are going to copy the contents of memory 1 into another memory. The left # sign in the prompt will change to the number 1 and the right # will begin to blink. This indicates that you should now enter the number of the memory into which you want to copy waveform memory 1.
---	---



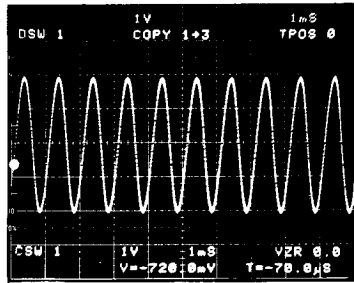
**Press**

**Remarks**

**Display**

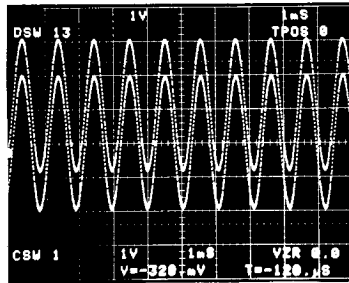
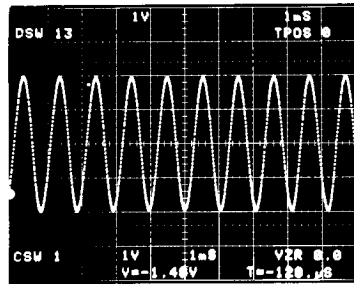
3

Tells the 7D20 to copy waveform memory 1 into waveform memory 3 and completes the copy process. (You could have copied the waveform into any of the other five memories of the 7D20.) At this point memory 3 contains a copy of the sine-wave signal being acquired into channel 1. The prompt now reads as shown and the COPY key light goes out. However, the content of waveform memory 3 is not yet displayed.



3

The 3 key illuminates, the copy prompt disappears, and the display changes to indicate that waveform 3 is being displayed. This will be difficult to see, however, because the waveform in memory 3 was copied in the same position as the original waveform in memory 1. Consequently, the two waveforms now overlap in the display. Separate the displayed waveforms by turning the CH 1 POSITION control to vertically reposition the signal being digitized into waveform memory 1.



You can display the contents of any of the six waveform memories just by pressing the appropriate MEMORY DISPLAY keys. Likewise, you can remove any waveform (except the cursor waveform) from the display by pressing the appropriate MEMORY DISPLAY key a second time.

## **A NOTE ABOUT THE DSW (DiSplayed Waveforms) ORDER**

*The DSW readout order is determined by the vertical position of the first data points of the displayed waveforms. Notice as you vertically reposition waveform 1 so its first data point is above the first data point of waveform 3, the DSW readout lists waveform 1 first. If you vertically position the first data point of waveform 1 below the first data point on waveform 3, the DSW readout changes to list waveform 3 first.*



## EXERCISE 4 CHANGING THE DESIGNATED CURSOR WAVEFORM

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

The CSW (Cursor Waveform) key shown in Figure 2-18 allows you to choose any of the six waveform memories as the cursor waveform. In the 7D20 the cursor waveform is always displayed. It cannot be turned off. The CSW (cursor waveform) readout in the lower left corner of the display tells which memory is currently designated as the cursor waveform (see Fig. 2-19). The two bottom lines of the

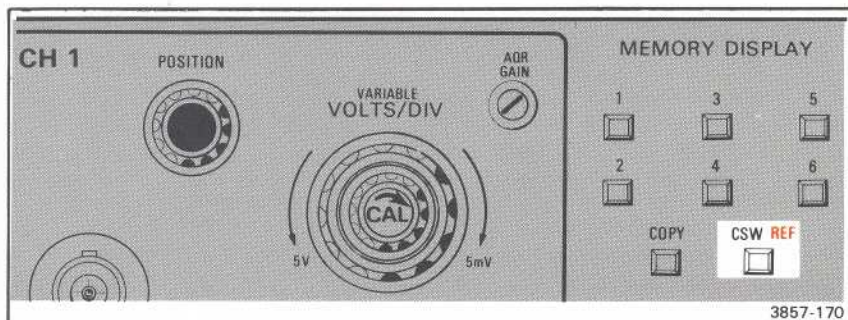


Figure 2-18. CSW REF key.

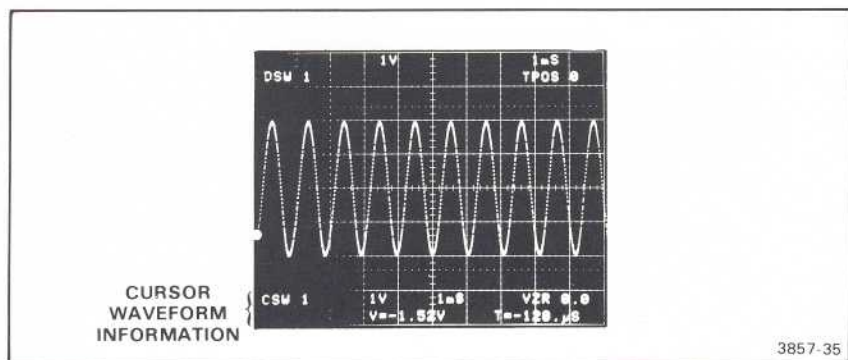
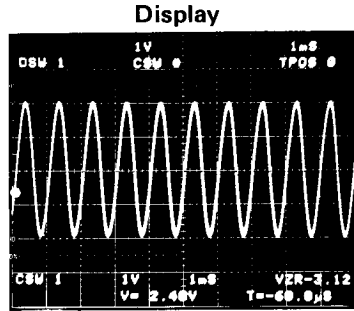


Figure 2-19. Cursor Waveform readout.

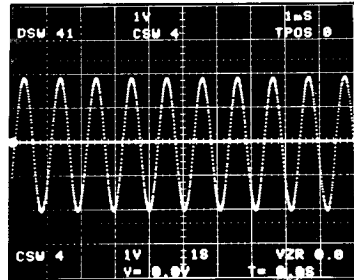
readout pertain only to the CSW. Initialization designates that memory 1 be the cursor waveform. Any waveform memory can be selected as the cursor waveform. The following illustrates how:

Press	Remarks
CSW	The CSW key will begin to blink and the prompt: CSW # appears in the display readout. The # in prompt also blinks to indicate that the 7D20 is waiting for the number of the new cursor waveform.



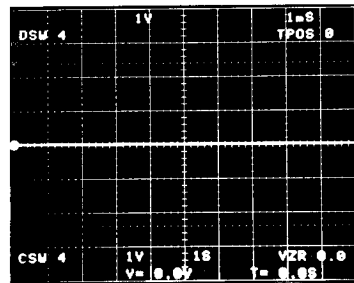
3857-36

4	The # in the prompt changes to a 4 and a horizontal line appears across the center of the display. The 4 key also illuminates and the CSW readout now indicates that waveform 4 is the cursor waveform.
---	---



3857-37

1	Waveform 1 can now be turned off since it is no longer the cursor waveform. The key extinguishes and the DSW readout changes to indicate that only waveform 4 is being displayed.
---	---



3857-38

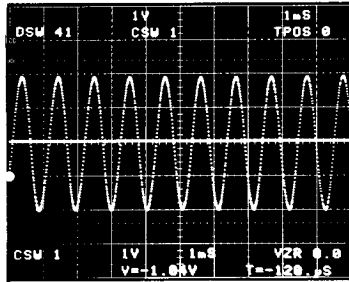
**Press**

**Remarks**

**Display**

CSW 1

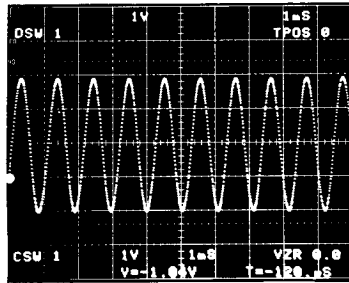
Reselects waveform 1 as the cursor waveform, returns waveform memory 1 to the display and illuminates the key.



3857-171

4

Turns off waveform memory 4.



3857-172

## EXERCISE 5 HOLDING A WAVEFORM IN MEMORY

Any waveform being digitized, such as the sine-wave signal currently being digitized and displayed, can be locked or held in the 7D20 memory by pressing the HOLD key shown in Figure 2-20. When you press the HOLD key, signal digitizing is instantly terminated. Waveform data contained in waveform memories 1 and 2 at that instant is retained unchanged in memory, complete with the associated acquire and display conditions. This is a manual or arbitrarily initiated function. For a triggered HOLD refer to EXERCISE 11 for HOLD NEXT.

The HOLD condition is visually indicated in two ways; the HOLD key illuminates and the displayed waveform appears to freeze (waveform motion is no longer visible) in the display. The HOLD key performs an on/off function, so pressing the key a second time will turn the HOLD function off and extinguish the key illumination. The 7D20 will then resume digitizing new waveform data into memory.

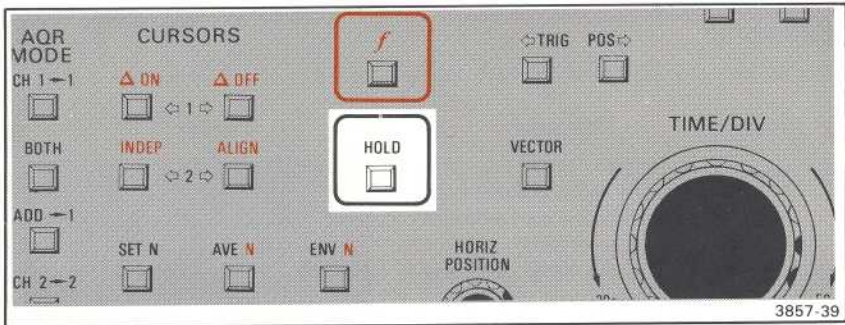


Figure 2-20. The HOLD key.



Digitally signed by  
<http://www.aa4df.com>

## EXERCISE 6 USING THE CURSORS

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

Two cursors are available for use with each waveform memory when the memory is selected as the cursor waveform. The cursors appear as intensified dots on the cursor waveform. Cursor 1 is always turned on but cursor 2 can be turned on or off as you choose. Initialization turns cursor 2 off and sets cursor 1 to the left most data point on the cursor waveform. The intensified dot at the beginning of the cursor waveform is cursor 1 (see Fig. 2-21). Using the cursor movement keys shown in Figure 2-22, cursor 1 can be positioned to any data point on the cursor waveform. Use of the cursors is illustrated in the following procedures.

### CURSOR 1

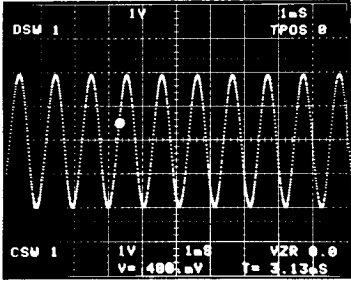
The following shows how to move cursor 1 on the cursor waveform.

Press	Remarks
-------	---------

1⇨

A single press of this slewing key causes cursor 1 to move to the next adjacent data point. Notice that when you press and hold the movement key, the cursor moves continuously across the full length of the cursor waveform. Cursor 1 stops when it reaches the last waveform data point. Also notice that as cursor 1 is moved, the cursor waveform readout changes to give both the vertical position and the horizontal time value of the waveform data point where cursor 1 is located. (The horizontal time value is referenced to the trigger position and the cursor 1 vertical position is referenced to Vertical Zero Reference.)

Display



3857-42

Press	Remarks	Display
⇐1	Moves cursor 1 back toward the beginning of the cursor waveform. Return cursor 1 to the first data point.	

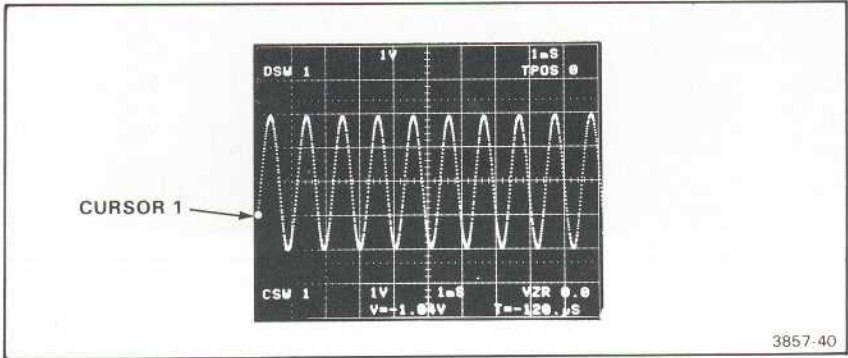


Figure 2-21. Cursor 1.

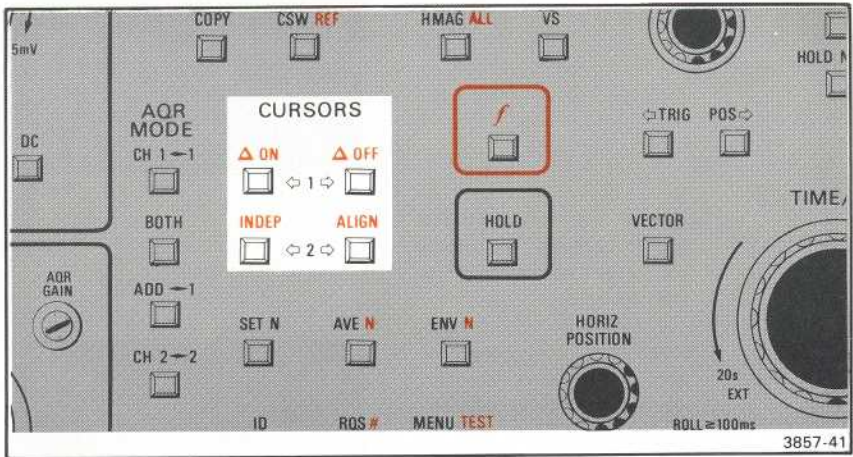


Figure 2-22. Cursor keys.



## CURSOR 2 AND THE $f$ (function shift) KEY

The following shows how to turn on cursor 2 using the  $f$  (function shift) key (see Fig. 2-23) and also illustrates how cursor 2 is moved about on the waveform.

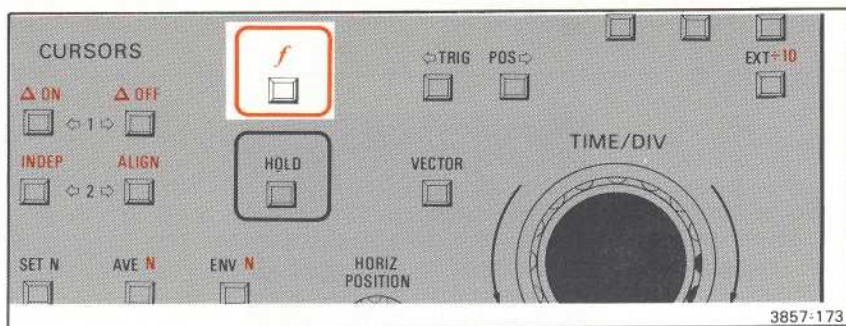
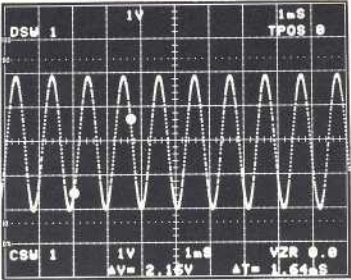
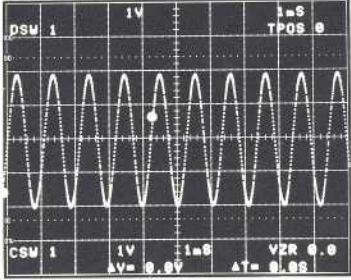
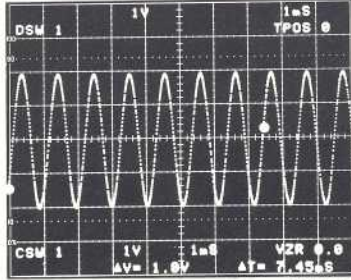


Figure 2-23. The shift function key.

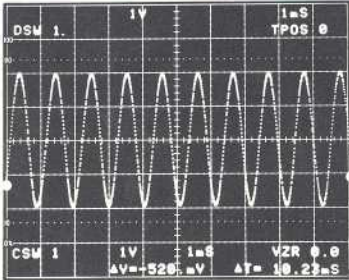
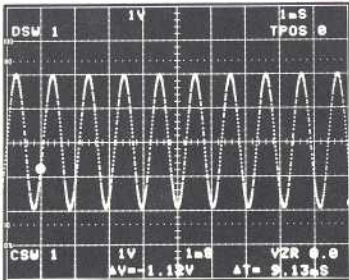
Press	Remarks	Display
1 $\rightarrow$	Move cursor 1 to the right about two divisions as shown.	
$f$	The $f$ (function shift) key begins to blink. This allows you to now select any of the shift functions.	
$\Delta$ ON	This turns on cursor 2 (which appears as an intensified dot), extinguishes the $\Delta$ OFF key and $f$ key, and illuminates $\Delta$ ON key. Note that when cursor 2 is turned on the cursor readout changed; it now gives the voltage and time difference of cursor 2 with respect to cursor 1.	

Press	Remarks	Display
<p>⇐ 2 2 ⇐</p>	<p>These two keys move cursor 2 the same way the ⇐1 and 1⇐ keys move cursor 1. However, cursor 2, when moving toward the beginning of the cursor waveform, cannot be moved past cursor 1. Cursor 2 comes to a stop when it reaches cursor 1. Position cursor 2 about two divisions to the right of cursor 1.</p>	 <p>3857-45</p>
1⇐	<p>When cursor 1 reaches the same data point as cursor 2 it pushes cursor 2 toward the end of the waveform. Cursor 1 cannot be positioned to the right of cursor 2.</p>	 <p>3857-46</p>
⇐1	<p>Notice as cursor 1 moves away, cursor 2 remains where it was previously positioned. Return cursor 1 to the first data point.</p>	 <p>3857-47</p>
<i>f</i>	<p>Again permits the selection of a shift function. The key will begin to blink.</p>	
△OFF	<p>Turns off cursor 2 illuminates the △OFF key. The △ON key and the <i>f</i> key are extinguished.</p>	



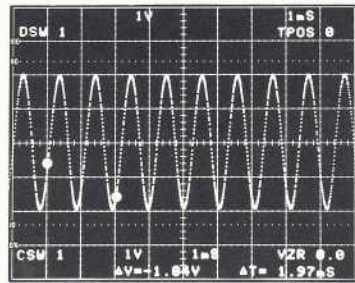
## USING THE CURSORS INDEPendently OR ALIGNed

The cursors associated with each of the six waveform memories can be operated in one of two ways, **INDEP**endently or **ALIGN**ed. If **INDEP**endent is used you can move the cursors on the cursor waveform without affecting the positions of the undisplayed cursors on the other waveforms. This can be useful when examining different portions of two different waveforms. When you want to examine vertical values at the same horizontal position on different stored waveforms, using **ALIGN** forces the cursor positions on all the waveforms to align themselves with the cursor positions on the cursor waveform. Initialize the 7D20 and continue as instructed below.

Press	Remarks	Display
<i>f</i>	Allows the selection of a shift function. The key begins to blink.	
<b>ΔON</b>	Turns on cursor 2. The key illuminates when pressed, the <i>f</i> key stops blinking, and the <b>ΔOFF</b> key extinguishes. Notice that the cursor readout at the bottom of the display has changed to now show the difference in voltage and time between cursor 1 and cursor 2.	3857-49
1⇨	Position cursor 1 to vertical graticule line 1.	
		3857-60

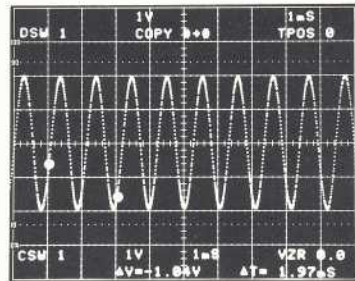
Press	Remarks
◊2	Position cursor 2 to vertical graticule line 3.

Display



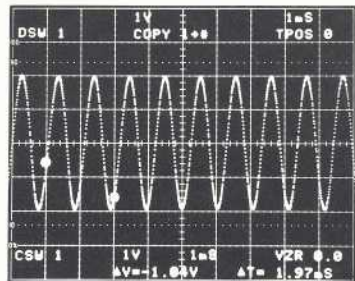
3857.61

COPY	The COPY key begins to blink and the COPY prompt appears in the display.
------	--



3857.62

1	Tells the 7D20 that you want to COPY waveform memory 1. Your selection is noted in the first # position of the COPY prompt.
---	---



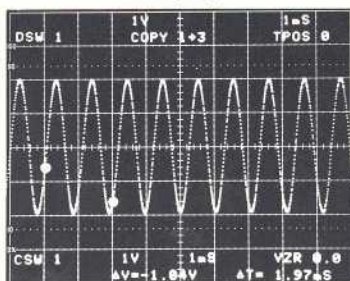
3857.63

Press

Remarks

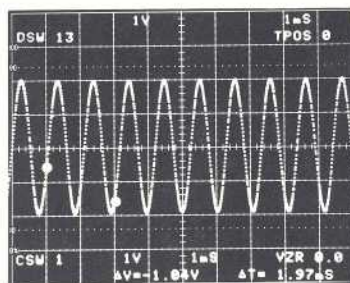
Display

- 3 Tells the 7D20 that you wish to COPY waveform memory 1 into waveform memory 3. The COPY prompt 3. The COPY prompt reflects your selection. The COPY prompt will disappear when another key is pressed.

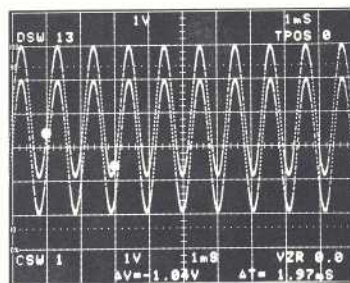


3857-64

- 3 Causes waveform memory 3 to be displayed, illuminates the 3 key, and removes the COPY prompt from the display. The DSW readout indicates that both waveforms 1 and 3 are being displayed. The displayed waveforms overlap at this time so adjust the CH 1 POSITION control to vertically position waveform 1, up one full division above its current position. This will allow you to view both waveforms. Notice that the cursors move up with waveform 1 because it is the the cursor waveform. Also, note the position of the cursors in preparation for the next step.

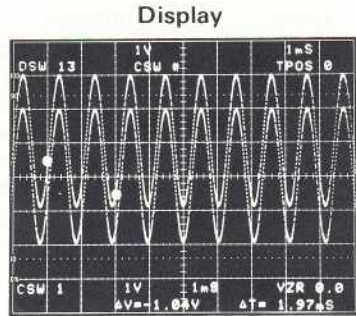


3857-65



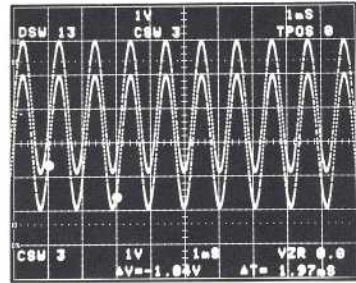
3857-66

Press	Remarks
CSW	The CSW key begins to blink and the CSW prompt appears in the display. The 7D20 is waiting for you to select the number of the cursor waveform.



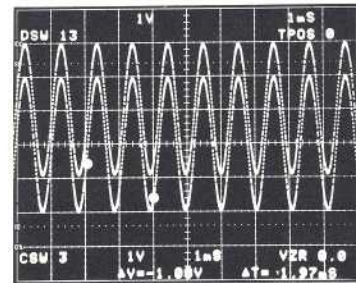
3857-67

3	This designates waveform memory 3 as the cursor waveform. Notice that the cursors are now on waveform 3 in the same horizontal location they previously occupied on waveform 1. In other words, the cursors are still <b>ALIGNED</b> with their previous horizontal position.
---	---



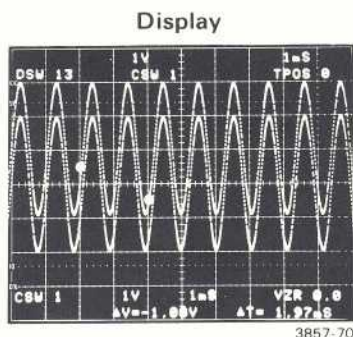
3857-68

1⇨ 2⇨	Position both cursors about one division to the right of their present position.
-------	--



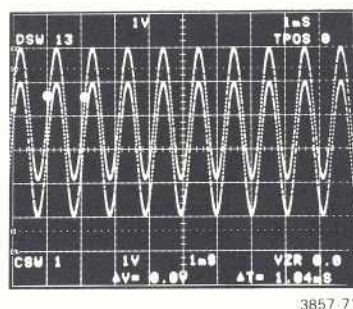
3857-69

Press	Remarks
CSW 1	Re-select waveform memory 1 as the cursor waveform and notice that the cursors are now on waveform 1 but are <b>ALIGNED</b> with the position you moved them to when waveform 3 was the cursor waveform.

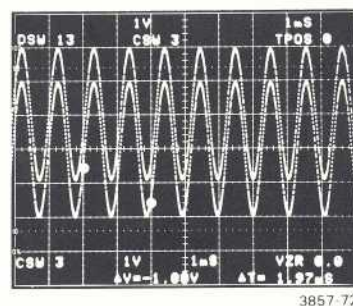


<b>f</b> INDEP	Select the <b>INDEP</b> endent cursor mode. The <b>ALIGN</b> key extinguishes and the <b>INDEP</b> key illuminates. As you will see, movement of the cursors on the cursor waveform (waveform 1) will have no effect on the position of the cursors (non displayed) on waveform 3.
----------------	--

⇐1 ⇐2	Position both cursors toward the beginning of waveform.
-------	---



CSW 3	Re-select waveform memory 3 as the cursor waveform. Notice that the cursors return to their previous position on waveform 3 and are no longer <b>ALIGNED</b> with the cursors on waveform 1.
-------	--





Press  
**ALIGN**  
 CSW 1

### Remarks

Notice the cursor positions have changed again. The **ALIGN** command forces immediate movement of the cursors without the need to use the cursor movement keys. The **ALIGN** mode allows you to measure vertical values on waveforms in exactly the same horizontal location. The **INDEPENDENT** mode allows you to measure different parts of the same waveform at the same time (see Fig. 2-24). Additional uses of the **ALIGN** and **INDEPENDENT** modes are illustrated in conjunction with the HMAG (Horizontal MAGnification) later in these exercises.

### Display

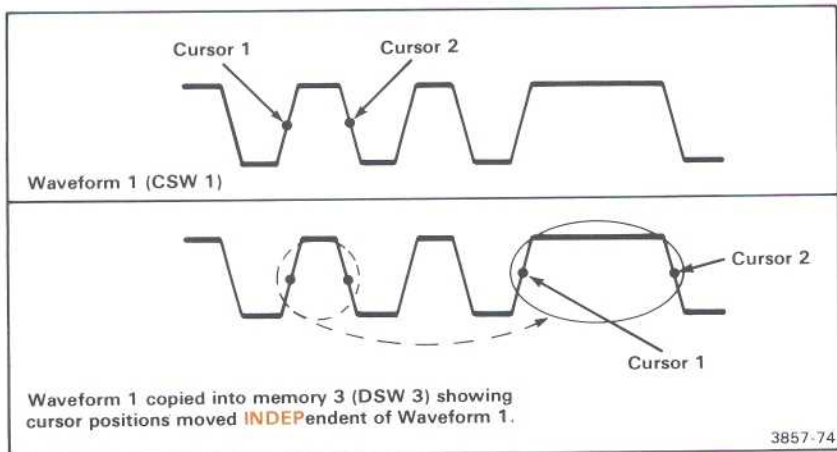
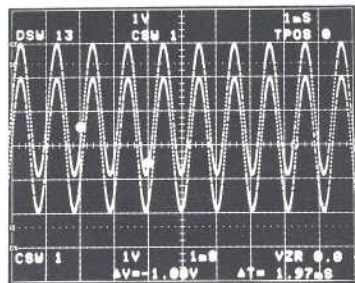


Figure 2-24. Using the cursors to measure different parts of the same waveform.

## EXERCISE 7 USING THE CURSOR WAVEFORM DISPLAY MODIFIER KEYS

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

The CURSOR WFM keys (shown in Fig. 2-25) permit you to manipulate the cursor waveform in a variety of ways. Generally, the VPUP $\uparrow$ , VPDN $\downarrow$ , VXPDP $\updownarrow$ , and VCMP $\updownarrow$  keys perform the same functions on the stored waveform displays that the CH 1 and CH 2 vertical POSITION and VOLTS/DIV controls perform on signals being actively acquired. These keys are operational on the cursor waveform only. When CH 1 and CH 2 are selected as the cursor waveform, CH 1 or CH 2 must be placed in HOLD, or REF must be on. Waveforms 3 through 6 are always in a hold condition. HMAG and VS functions are operational on either the waveform(s) being acquired or on waveforms in HOLD.

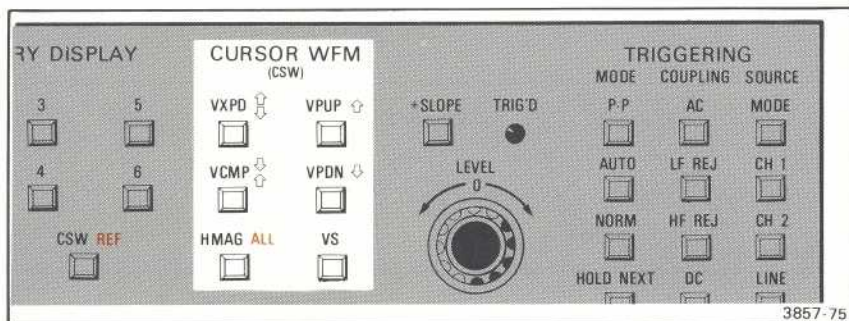
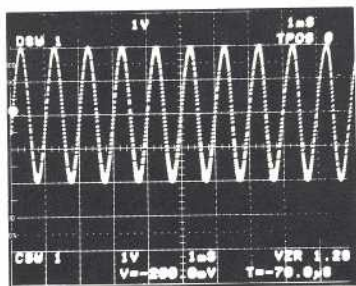


Figure 2-25. Cursor Waveform keys.

## HOW TO VERTICALLY REPOSITION THE CURSOR WAVEFORM

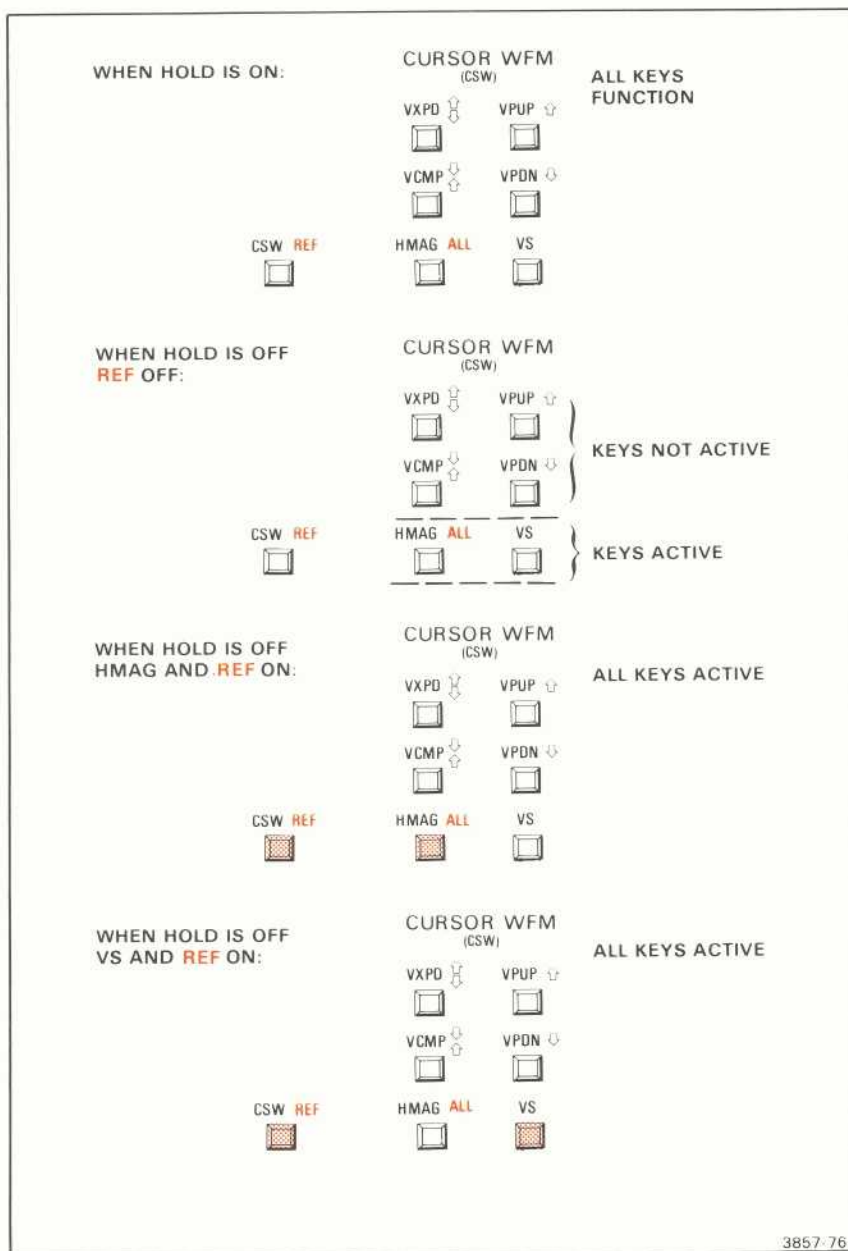
The VPUP $\uparrow$  (vertical position up) and the VPDN $\downarrow$  (vertical position down) keys permit you to vertically move the cursor waveform up or down in the display. The following exercise shows you how.

Press	Remarks	Display
HOLD	Terminates the digitizing process and holds the current digitized waveform in waveform memory 1. This is necessary because CURSOR WFM keys VPUP $\uparrow$ , and VPDN $\downarrow$ , have no effect on a cursor waveform that is being actively digitized. They work as vertical separation controls with the REFERENCE waveform present. Except under the conditions shown in Figure 2-26, a cursor waveform that is being actively digitized is vertically repositioned using the respective CH 1 or CH 2 POSITION control.	
VPUP $\uparrow$	(VERTICAL POSITION UP). A single press of this key moves the cursor waveform up incrementally. The key illuminates and the VZR (Vertical Zero reference) readout indicates the new GND position of the waveform in divisions with respect to the center horizontal graticule line. Continuous movement of the cursor waveform results when you press and hold this key. Maximum vertical positioning is approximately 5 divisions (plus or minus) from the zero reference.	



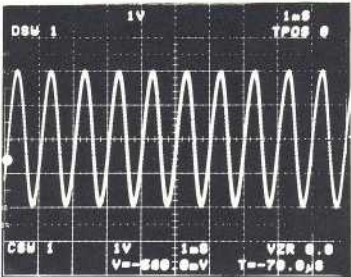
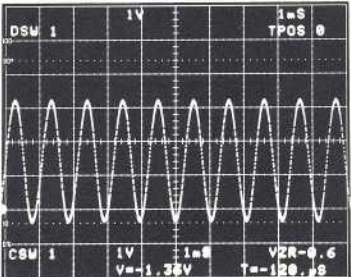
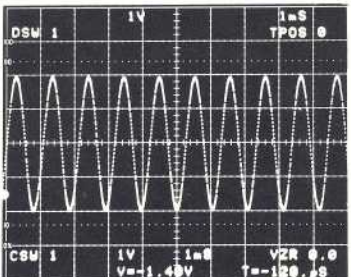
3857-78





3857-76

Figure 2-26. Conditions under which the CURSOR WFM keys are active for waveforms 1 or 2.

Press	Remarks	Display
VPDN ⇄	(VERTICAL POSITION DOWN). Press and hold this key to return the cursor waveform to its original position. Movement of the waveform stops when the original zero reference position is reached. Release the key and notice that the original VZR readout is restored and that neither the VPUP ⇄ key nor the VPDN ⇄ key is illuminated.	 <p style="text-align: right;">3857-79</p>
VPDN ⇄	A single press of this key will now cause the cursor waveform to move down incrementally and illuminate the VPDN ⇄ key. As with the VPUP ⇄ key, continuous movement of the waveform will result if you press and hold this key. Position the displayed waveform somewhat below center.	 <p style="text-align: right;">3857-80</p>
HOLD	Turns off the HOLD function and allows the 7D20 to resume digitizing into waveform memory 1. The HOLD key will extinguish. Notice the trace returns to its original position because the vertical position keys don't affect a cursor waveform that is not in HOLD.	 <p style="text-align: right;">3857-81</p>

## HOW TO VERTICALLY EXPAND OR COMPRESS THE CURSOR WAVEFORM

The VXPD  $\updownarrow$  (vertical expand) and VCMP  $\updownarrow$  (vertical compress) keys, shown in Figure 2-27, allow you to expand or compress the cursor waveform two full increments (in a 1,2,5 sequence) from its acquired vertical scale factor. To best demonstrate this feature, adjust the signal generator for a sine-wave amplitude of two divisions in the display. Then proceed with the following:

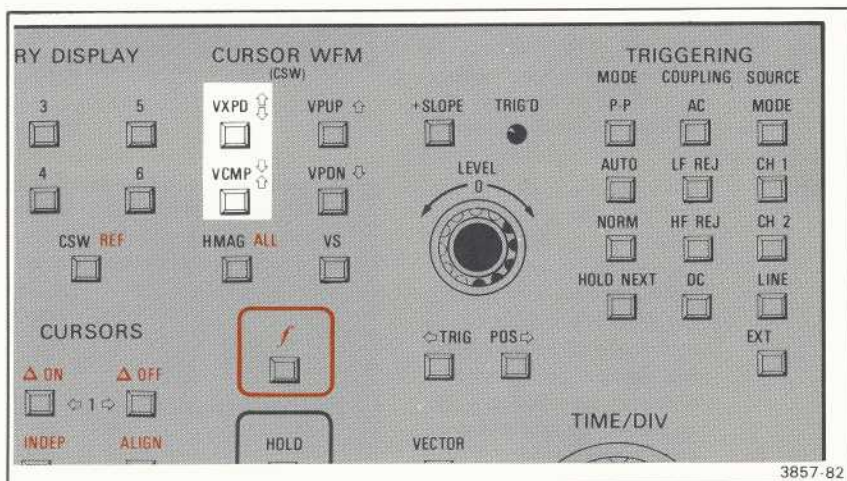
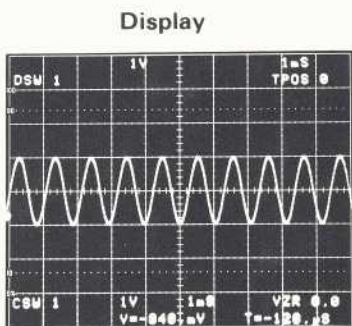


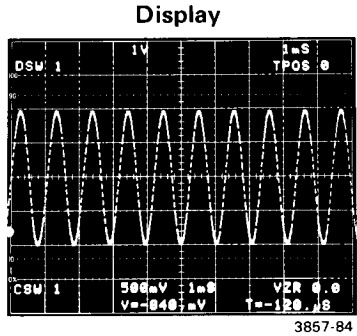
Figure 2-27. Vertical compress and expand keys.

Press	Remarks
HOLD	Terminates the digitizing process and holds the current digitized waveform in waveform memory 1. The VXPD $\updownarrow$ and VCMP $\updownarrow$ keys have no effect on a cursor waveform that is being actively digitized except with the <b>REFERENCE</b> waveform present.



3857 83

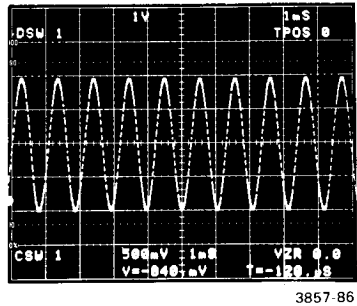
Press	Remarks
VXPD $\uparrow\downarrow$	Press this key one time and release. The cursor waveform expands and the VXPD $\uparrow\downarrow$ key illuminates. All expansions and compressions increment or decrement the stored VOLTS/DIV in calibrated 1,2,5 steps. The new vertical scale factor is displayed in the CSW readout.



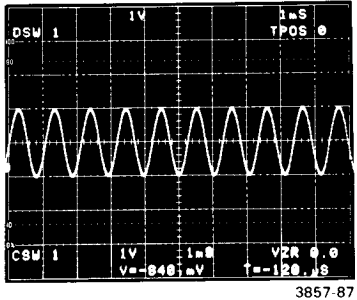
VXPD $\uparrow\downarrow$	Press this key a second time and the cursor waveform increments once more to the next higher vertical scale factor. Again, the CSW readout changes to reflect the new vertical display factor.
---------------------------	--



VCMP $\uparrow\downarrow$	Pressing this key causes the cursor waveform to compress. The CSW readout will reflect the change. The VXPD $\uparrow\downarrow$ key is still illuminated because the cursor waveform is still expanded.
---------------------------	--



VCMP $\uparrow\downarrow$	Press this key a second time and the cursor waveform returns to its original digitized vertical scale factor. The VXPD $\uparrow\downarrow$ key extinguishes when this occurs.
---------------------------	--



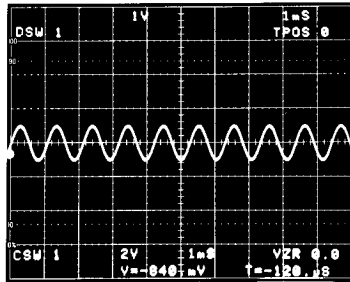
**Press**

**Remarks**

**Display**

VCMP  $\updownarrow$

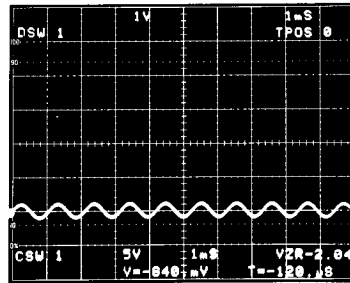
The cursor waveform compresses again. The VCMP  $\updownarrow$  key illuminates indicating the waveform has been compressed relative to its original digitized amplitude. The CSW readout displays the new vertical scale factor.



3857-88

VCMP  $\updownarrow$   
VPDN  $\updownarrow$

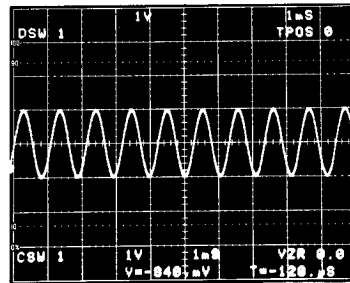
Compresses the cursor waveform again and positions it downward in the display.



3857-89

HOLD

Terminates the HOLD function and permits the 7D20 to resume digitizing into waveform memory 1. Notice the displayed cursor waveform returns to its original position and is no longer affected by the vertical position, vertical compression or expansion keys when HOLD is cancelled.



3857-90

## USING HMAG (HORIZONTAL MAGNIFY) AND CURSOR MOVEMENT

The HMAG key (see Fig. 2-28) permits you to horizontally magnify the cursor waveform by 10 times. Movement of cursor 1 appears different when the cursor waveform is magnified. This is because only a portion of the magnified waveform appears in the display and cursor 1 movement is designed to allow any portion of the magnified waveform to be positioned within the display area. Movement of cursor 2 is not affected by horizontal magnification. The following exercise illustrates the HMAG function and the movement of cursor 1 on the magnified waveform.

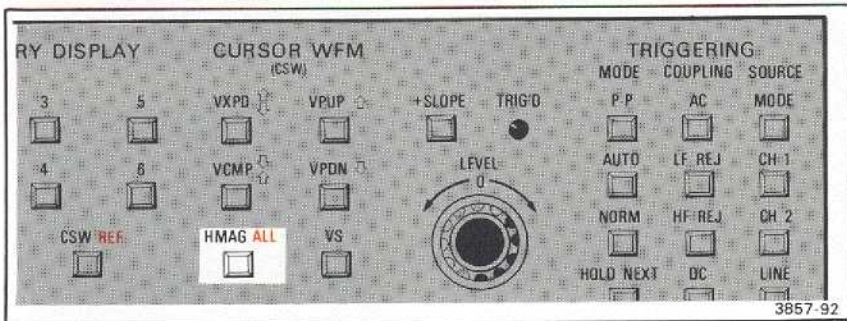
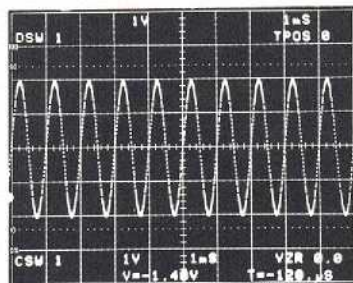


Figure 2-28. Horizontal Magnify key.

### Remarks

Adjust the signal generator amplitude for a 4-division display of the sine wave.

### Display





## USING HMAG (HORIZONTAL MAGNIFY) AND CURSOR MOVEMENT

The HMAG key (see Fig. 2-28) permits you to horizontally magnify the cursor waveform by 10 times. Movement of cursor 1 appears different when the cursor waveform is magnified. This is because only a portion of the magnified waveform appears in the display and cursor 1 movement is designed to allow any portion of the magnified waveform to be positioned within the display area. Movement of cursor 2 is not affected by horizontal magnification. The following exercise illustrates the HMAG function and the movement of cursor 1 on the magnified waveform.

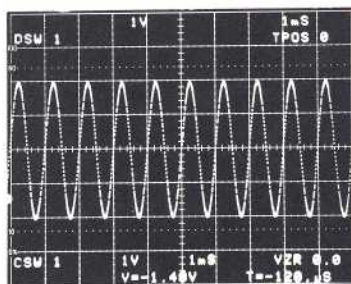


Figure 2-28. Horizontal Magnify key.

### Remarks

Adjust the signal generator amplitude for a 4-division display of the sine wave.

### Display



3857-91

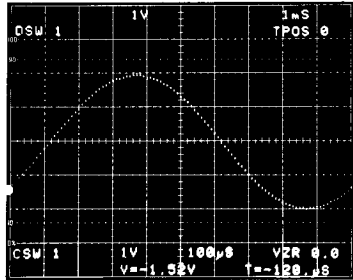
**Press**

**Remarks**

**Display**

HMAG

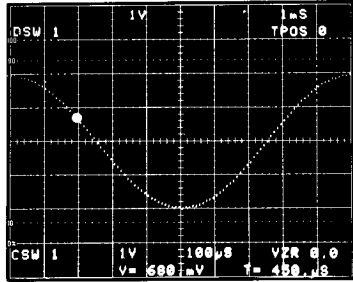
The key illuminates when pressed and the cursor waveform being digitized into waveform memory 1 is immediately magnified in the display. HMAGnification is by a factor of 10 times so the magnified waveform is displayed with one tenth the number of horizontal points. Notice that cursor 1 is on the first data point on the magnified waveform.



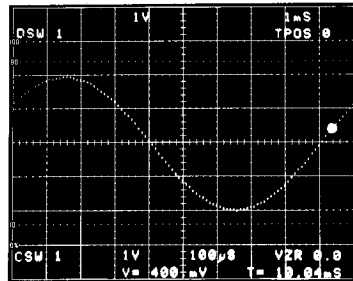
3857-93

1⇨

This slewing key moves cursor 1 toward the end of the magnified cursor waveform. If you press and hold the cursor movement key continuous movement of the cursor will result. However, notice that the cursor stops at vertical graticule line 2 and the waveform moves to the left with respect to the cursor. As the end of the magnified waveform comes into the display area, cursor 1 resumes movement to the end of the waveform while the magnified waveform remains stationary in the display. In this manner cursor 1 can be positioned to any data point on the horizontally magnified cursor waveform and you can scan the entire magnified waveform for viewing. This also ensures that cursor 1 stays always within the display area. This sequence is reversed when cursor 1 is moved back toward the beginning of the magnified cursor waveform. Return cursor 1 to the



3857-94



3857-95



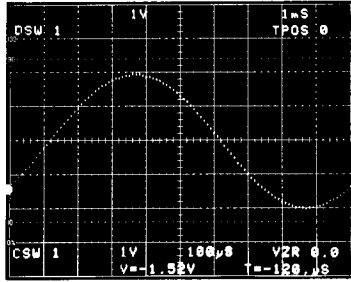
**Press**

**Remarks**

**Display**

HMAG

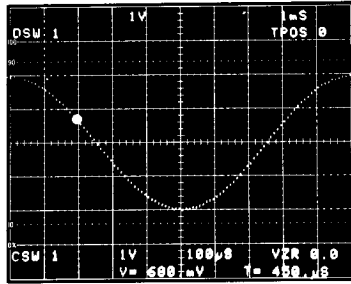
The key illuminates when pressed and the cursor waveform being digitized into waveform memory 1 is immediately magnified in the display. HMAGnification is by a factor of 10 times so the magnified waveform is displayed with one tenth the number of horizontal points. Notice that cursor 1 is on the first data point on the magnified waveform.



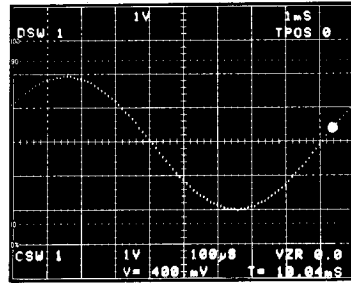
3857-93

1⇨

This slewing key moves cursor 1 toward the end of the magnified cursor waveform. If you press and hold the cursor movement key continuous movement of the cursor will result. However, notice that the cursor stops at vertical graticule line 2 and the waveform moves to the left with respect to the cursor. As the end of the magnified waveform comes into the display area, cursor 1 resumes movement to the end of the waveform while the magnified waveform remains stationary in the display. In this manner cursor 1 can be positioned to any data point on the horizontally magnified cursor waveform and you can scan the entire magnified waveform for viewing. This also ensures that cursor 1 stays always within the display area. This sequence is reversed when cursor 1 is moved back toward the beginning of the magnified cursor waveform. Return cursor 1 to the



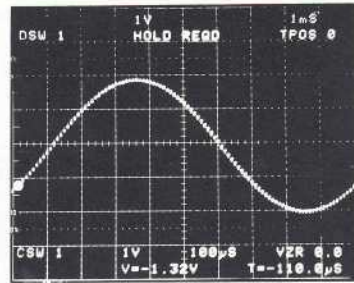
3857-94



3857-95

Press	Remarks
1⇨ (cont)	beginning of the cursor waveform.
VXPD⇨ VCMP⇨ VPUP⇨ VPDN⇨	Notice that these cursor waveform keys don't affect an HMAG'd cursor waveform that is not in HOLD. Also, notice that the message HOLD REQD (HOLD required) is displayed when you attempt to use these keys when not in HOLD except with the <b>REF</b> erence waveform present.

## Display



## HMAG AND THE REFERENCE WAVEFORM

The **REF** key (shown in Fig. 2-29) permits you to view the original, unmagnified cursor waveform at the same time you are viewing the horizontally magnified version. The cursors appear on the same data points on both versions of the waveform. The following exercise illustrates the use of this function.

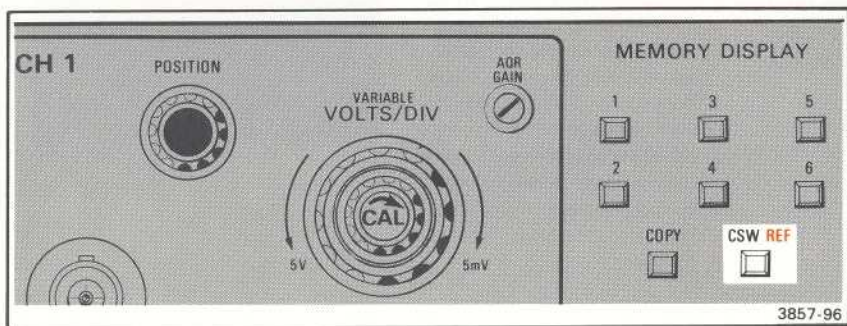


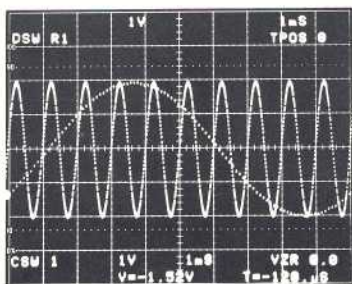
Figure 2-29. REFERENCE Waveform key.

Press	Remarks	Display
-------	---------	---------

	Make sure HOLD is not selected.	
--	---------------------------------	--

<i>f</i>	The <i>f</i> key illuminates to permit the selection of a shift function.	
----------	---	--

CSW REF	Selects the REFERENCE waveform for display. The REF waveform appears in the display as shown. Notice that the DSW readout indicates the REFERENCE waveform is being displayed. The first data point of each version of the waveform overlap. Cursor 1 is positioned on that point. Adjust the CH 1 POSITION control and notice that both versions of the cursor waveform are vertically re-positioned as the control is turned. Return the waveforms to their centered position.	
---------	--	--



3857-97

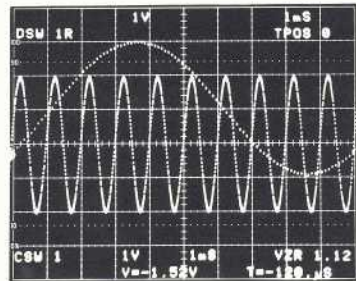
## Press

## Remarks

## Display

VPUP⇧

Notice that this key vertically re-positions only the magnified version of the waveform. Position the magnified waveform about one division above the REFERENCE waveform. Notice that cursor 1 is positioned on data point 1 on both versions of the waveform. Both cursors appear on the same data point on the REFERENCE waveform as on the HMAG'd waveform. VPUP⇧ and VPDN⇩ act as separation controls. Change CH 1 POSITION control and notice the waveforms move together but maintain separation.



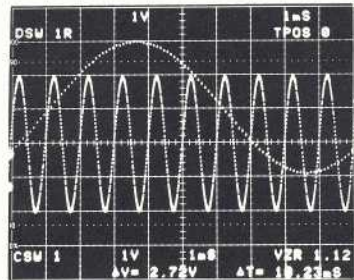
3857 98

*f*

The *f* key begins to blink indicating that you can select a shift function.

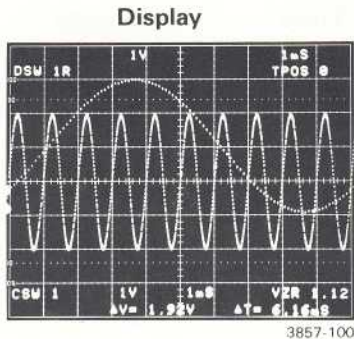
△ON

Turns on cursor 2 which is located on the last data point of both the HMAG'd and REFERENCE waveform. However, cursor 2 is not visible on the HMAG'd waveform at this time because the data point that it is positioned on is outside the display area.

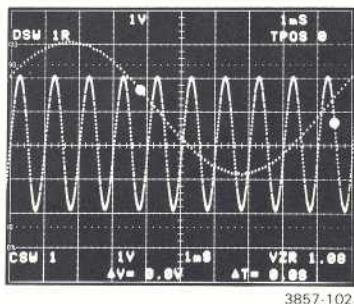
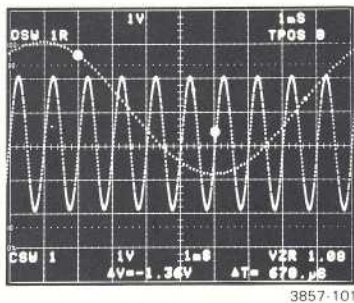


3857 99

Press	Remarks
⇨ 2	Position cursor 2 on or about vertical graticule line 6 on the REFERENCE waveform. Cursor 2 should still not be visible on the magnified cursor waveform.



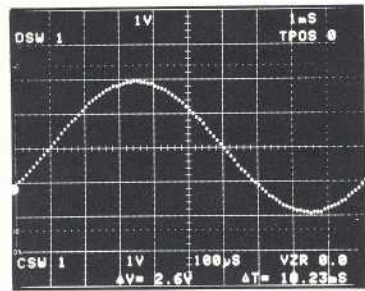
⇨ 1	Advance cursor 1 to the right toward the end of the cursor waveform. Notice how the magnified waveform moves to the left while cursor 1 (on the magnified waveform) remains fixed at vertical graticule line 2. On the REFERENCE waveform cursor 1 moves to the right in the usual manner. Observe cursor action on both the REFERENCE waveform and the magnified waveform as you continue moving cursor 1 toward the end of the waveform; cursor 2 comes into the display and travels with the magnified version of the waveform till it reaches cursor 1, at which time cursor 1 will push cursor 2 toward the end of the cursor waveform. When the end of magnified waveform comes into the display, the cursors resume travel till they reach the last data point on the waveform.
-----	--



Press	Remarks	Display
1 ⇄	Return cursor 1 to the beginning of the waveform while observing the movement of the cursors on both versions of the waveform.	
VCMP ⇄ VPUP ⇄ VPDN ⇄ VXPD ⇄	Notice that because the HMAG key is illuminated, these functions affect the magnified portion of the cursor waveform display. Note: To modify the REFERENCE waveform you must be in a HOLD condition and HMAG and VS must be off.	

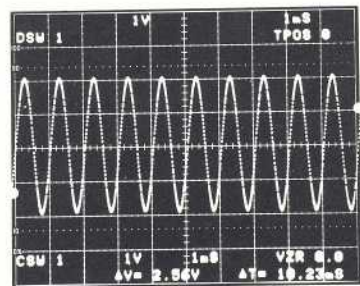
*f* The *f* key illuminates to permit the selection of a shift function.

**REF** Turns off the REFERENCE waveform.



3857-188

HMAG Terminates HMAG and the HMAG key extinguishes.



3857-174



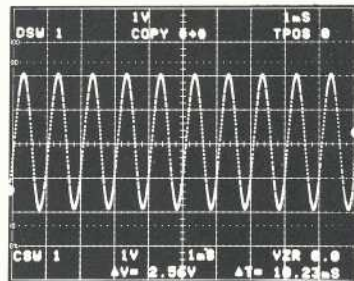
## HOW TO USE HMAG ALL

This function key, shown in Figure 2-30 allows you to horizontally magnify **ALL** the displayed waveforms. To illustrate this function we will first copy the cursor waveform into all the other waveform memories. Initialize the 7D20 and proceed as instructed below:



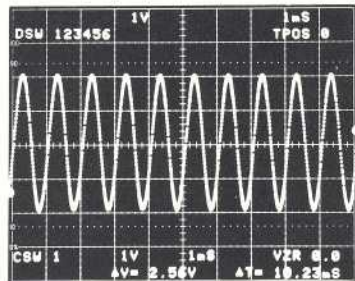
Figure 2-30. Horizontal Magnify **ALL** waveforms key.

Press	Remarks	Display
HOLD	Stops digitizing process and holds the waveform in memory.	
COPY	The COPY key begins to blink and the COPY prompt appears in the display.	



3857-175

Press	Remarks	Display
1	Selects waveform memory 1 as the waveform to be copied. The COPY prompt will confirm your selection.	
2	Selects waveform memory 2 as the destination of the COPY. Again, the COPY prompt confirms what you've selected.	
COPY	The COPY key begins to blink and the COPY prompt again requests that you select a waveform to copy.	
1	Selects waveform 1 as the waveform to be copied. The COPY prompt confirms your source selection.	
3	Selects waveform memory 3 as the destination of the COPY. Continue the above process until waveform 1 is copied into all the rest of the waveform memories (4, 5, and 6).	
2, 3, 4, 5, & 6	Displays the copied waveforms. This is indicated by the DSW readout. However, since the waveforms were all copied in the same position as waveform 1, it will appear as though only one waveform is being displayed.	



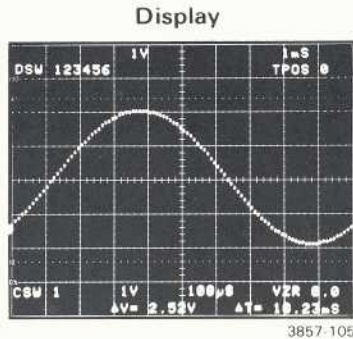
3857-104



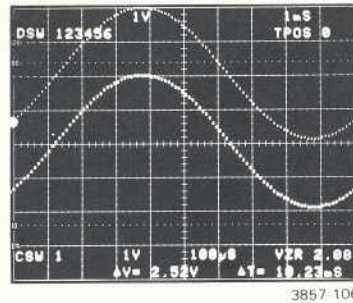
The key blinks indicating that you can now select a shift function.



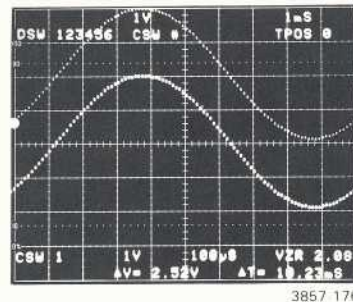
Press	Remarks
HMAG ALL	The key illuminates and all six displayed waveforms are horizontally magnified.



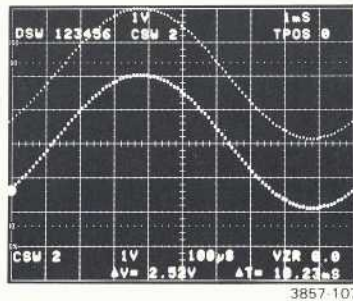
VPUP $\uparrow$	Vertically position the CSW to the top of the display.
-----------------	--



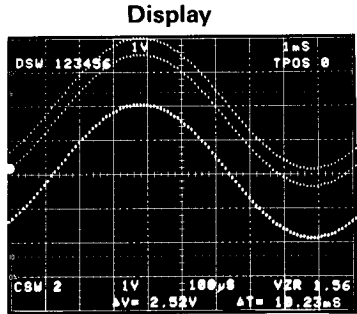
CSW	The key illuminates and the CSW prompt appears in the display to indicate that you can now select any waveform memory for the cursor waveform.
-----	--



2	Selects waveform memory 2 as the cursor waveform. Notice that cursor 1 moves to waveform 2 and that the readout prompt and the CSW display now indicate that waveform 2 is the cursor waveform.
---	---



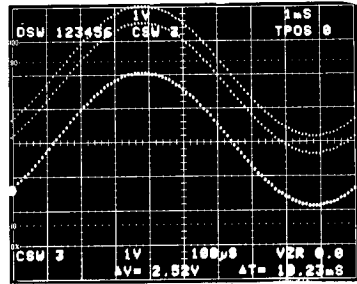
Press	Remarks
VPUP $\uparrow$	The key illuminates and causes waveform 2 to move vertically up in the display, separating it from the other magnified waveforms and closer to waveform 1. Position waveform 2 just below waveform 1 as shown.



3857-108

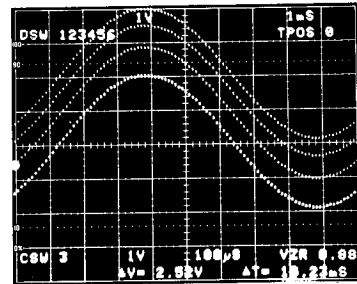
CSW	Again allows selection of a new cursor waveform.
-----	--

3	Selects waveform 3 as the cursor waveform. The cursor moves to waveform 3.
---	--



3857-109

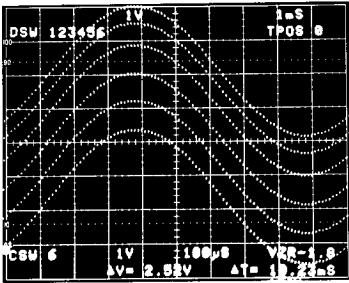
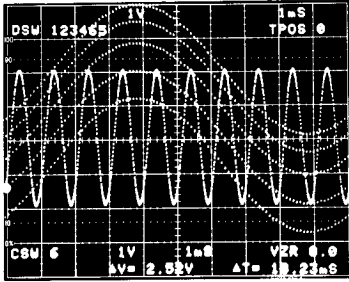
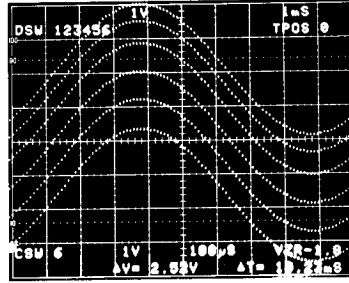
VPUP $\uparrow$	Now moves waveform 3 vertically up in the display. Position waveform 3 just below waveform 2.
-----------------	---



3857-110

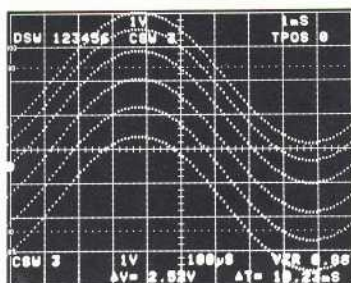
CSW	Again allows selection of a new cursor waveform.
-----	--

5	Selects waveform 5 as the cursor waveform.
---	--

Press	Remarks	Display
VPDN $\Downarrow$	Moves waveform 5 down in the display. Position waveform 5 just below its original position.	
CSW, 6 VPDN $\Downarrow$	Repeat the above process for waveform 6. When finished, the six waveforms should be separated and positioned as shown. Notice that the vertical position of the first data point of each waveform is reflected in the order of the DSW readout.	3857-111
HMAG	Notice that when you press this key, only the cursor waveform (waveform 6) is unmagnified and that the HMAG key extinguishes.	
HMAG	A second pressing of this key causes the cursor waveform to again be horizontally magnified and the key again illuminates.	3857-112
HMAG	A second pressing of this key causes the cursor waveform to again be horizontally magnified and the key again illuminates.	
		3857-113

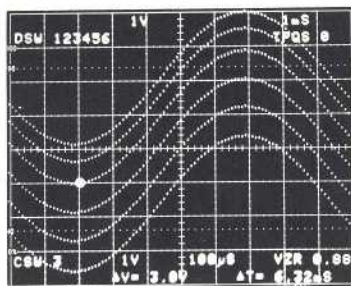
Press	Remarks
CSW 3	Select waveform memory 3 as the cursor waveform. Waveform 3 is selected as the cursor waveform for convenience only; any of the waveforms could be selected for the next two steps.

## Display



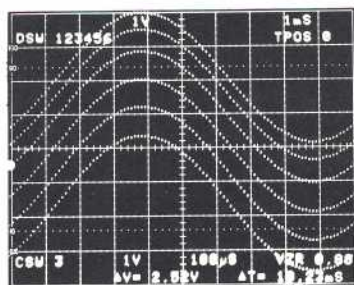
3857-114

1⇨	Press and hold the cursor 1 movement key. Notice that the cursor moves on waveform 3 till it reaches the second vertical graticule line. Then, the cursor movement stops and all the magnified waveforms travel across the display (simultaneously) with respect to cursor 1. The reason all the waveforms move together is because their horizontal cursor positions are <b>ALIGN</b> ed. When the last data point of the waveform comes into the display, waveform movement stops and cursor 1 resumes travel to the end of waveform 3.
----	---



3857-115

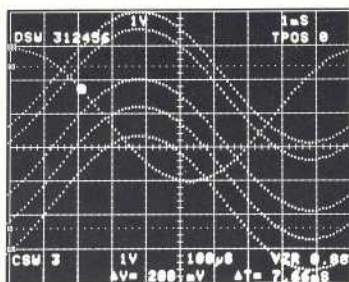
⇨1	Pressing this key reverses the previous process and returns cursor 1 to the beginning of waveform 3.
----	--



3857-116

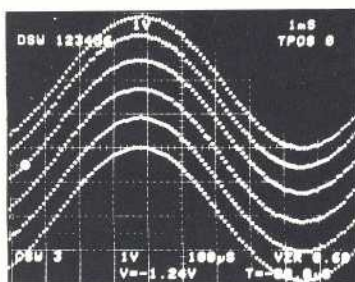
Press	Remarks	Display
<b>f INDEP</b>	Turns off the <b>ALIGN</b> mode and turns on the <b>INDEP</b> mode.	

1↔	Notice now that only the cursor waveform moves across the display when cursor 1 reaches the second graticule line.	
----	--	--



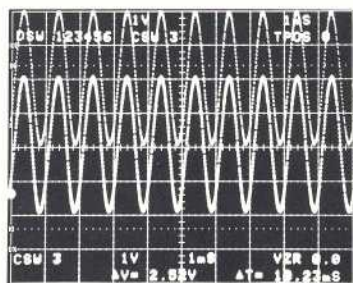
3857-117

<b>f ALIGN</b>	Notice that all waveforms are horizontally aligned with the cursor waveform.	
----------------	--	--



3857-118

<b>f HMAG ALL</b>	Turns off <b>HMAG ALL</b> . Notice that all displayed waveforms return to their original position in the display.	
-------------------	---	--



3857-119

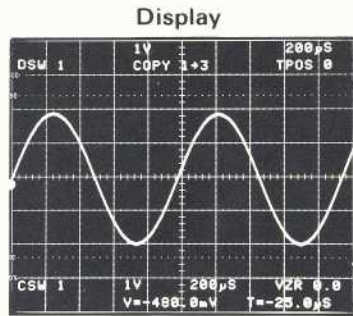
### USING THE VS (versus) FUNCTION

The versus function key, shown in Figure 2-31 permits you to display the cursor waveform with respect to another waveform rather than time. Initialize the 7D20 as instructed in the preliminary set-up instructions at the beginning of this section. Set the TIME/DIV to 200  $\mu$ s/div and set the trigger LEVEL to center position.



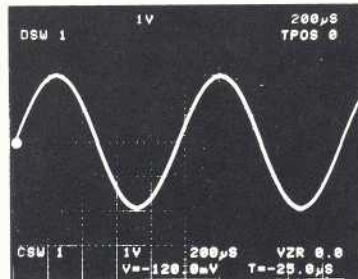
Figure 2-31. The VS (Versus) key.

Press	Remarks
COPY,1,3	Saves a duplicate of waveform 1 into waveform memory 3.



3857-122

VS	The VS key will begin to blink and the VS prompt appears in the display. The # sign in the prompt also blinks, indicating that the 7D20 is waiting for you to designate a waveform number to substitute for the time axis.
----	--



3857-124

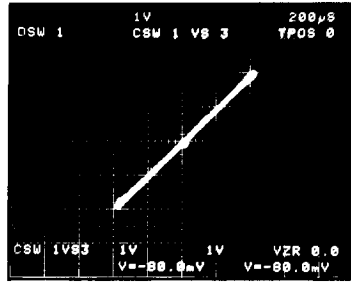
**Press**

**Remarks**

**Display**

3

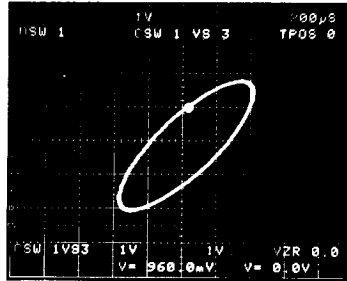
This causes the cursor waveform (waveform 1) to be versus waveform 3. The VS key remains illuminated but is no longer blinking. The resultant display will appear as shown because waveform 1 is used for the X and Y axis of the display. Notice that the cursor readout uses volts for the horizontal value rather than seconds.



3857-125

**TRIGGERING LEVEL**

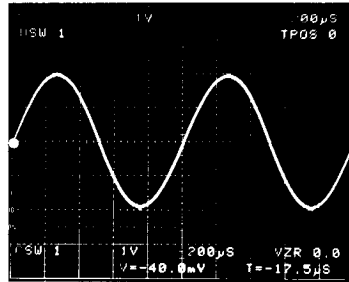
Turn this control to its fully clockwise position. This will change the phase relationship of waveforms 1 and 3 as shown. In this way, the cursor waveform can be displayed VS the data in any of the waveform memories, including itself.



3857-191

**VS**

Turns off the VS function and extinguishes the VS key. The display now shows only the cursor waveform displayed with respect to time.



3857-126

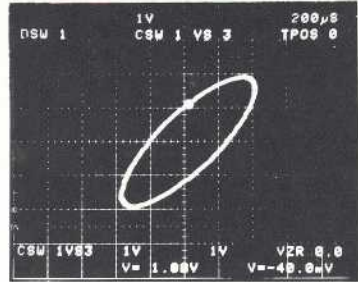


### USING THE REFERENCE WITH VS

The following illustrates how to use the REFERENCE feature when viewing a VS waveform display.

Press	Remarks
VS, 3	Displays the cursor waveform versus waveform 3 as before.

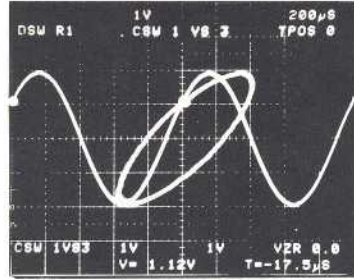
#### Display



3857-127

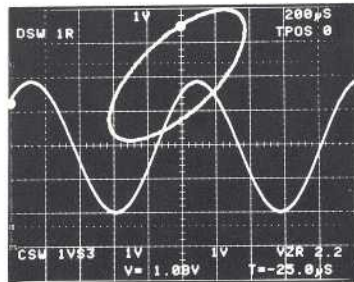
**f REF**

Turns on the REFERENCE Waveform for viewing at the same time.



3857-186

VPUP ⇅	Vertically position the versus display about 2 divisions higher.
--------	--

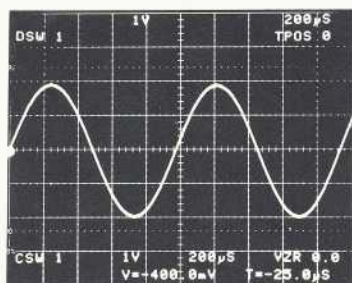


3857-128



Press	Remarks	Display
1⇨ ⇨1	While positioning cursor 1 left and right, observe its position on both the REFERENCE and VS waveforms. Here the REFERENCE serves as a time reference as indicated by the horizontal coordinate reading in seconds.	

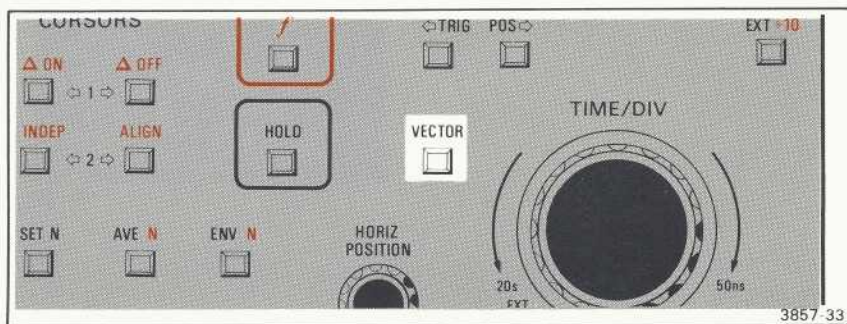
VS	Terminates both the VS function and turns off the REFERENCE waveform. Return the TRIGGERING LEVEL control to midrange.	
----	--	--



3857-187

## VECTORS AND DOTS

Up to this point the waveform displays you have viewed consist of a series of dots that represent the digitized waveform data points. As these dots become separated vertically it sometimes becomes difficult to determine which dot follows which. Your 7D20 has a selectable VECTOR function (see Fig. 2-32) that connects time adjacent dots. This allows you to better determine their time relationships. The VECTORS are designed such that you can still easily discern the waveform data points because they appear intensified with the VECTORS on.



3857-33

Figure 2-32. VECTOR key.

## EXERCISE 8 USING THE AVERAGE AND ENVELOPE FUNCTIONS AND SET N

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

These function keys, shown in Figure 2-33 permit you to average or envelope waveforms that are being digitized. AVEraging is useful for reducing uncorrelated noise in signals and improves its signal to noise ratio. ENVELOping is useful for revealing subtle variations in signals and allows the 7D20 to be left unattended for a long period of time while monitoring signals for time and amplitude variations such as frequency drift. Refer to the Detailed Operating Instructions later in this section for the algorithms used for waveform processing. Waveforms can be AVEraged or ENVELOped continuously or you can use the SET N feature to select the number of waveforms to process before automatically entering HOLD. The following shows how, but first, make sure the 7D20 is initialized and that the 1 kHz sine wave is connected as described in the PRELIMINARY SET-UP INSTRUCTIONS.

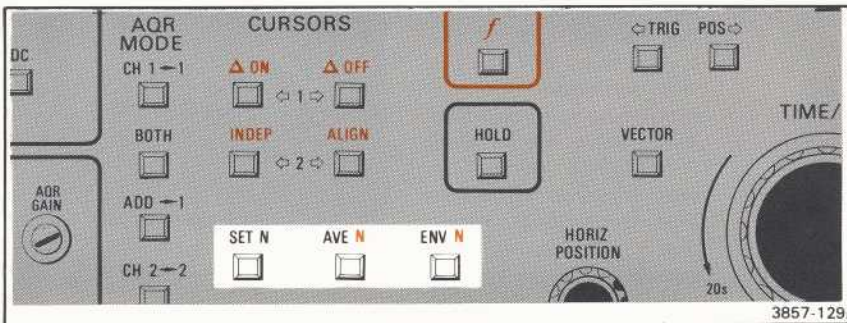
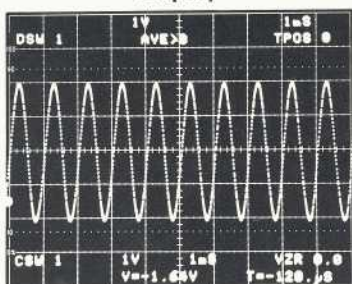


Figure 2-33. Waveform processing keys.

Display

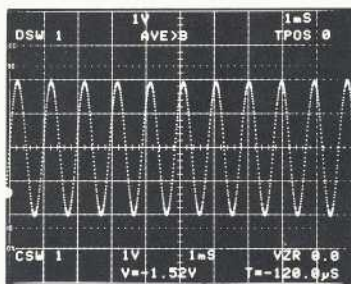
**Press**                      **Remarks**

**AVE**                      The AVE key illuminates and the 7D20 starts to continuously average the signal as it is being digitized. Slightly adjust the CH 1 POSITION control and observe the effect. The display readout displays the number of averages as they accumulate up to the current value of N, at which point the display simply indicates that the number of averages exceeds the current value of N and a ">" sign appears in the prompt. Selection of the AVE function will cancel the HOLD, ENV, ENV N, SET N, and AVE N functions if any are active.



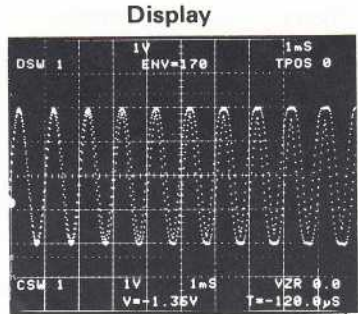
3857 130

**HOLD**                      This terminates the AVE process and displays the total number of averages in the readout up to the current value of N. You can also terminate the AVE function by selecting HOLD NEXT, which would allow the 7D20 to average the next triggered waveform, then enter HOLD. Again, the total number of waveforms averaged up to the current value of N would be displayed. Termination of the AVE function can also be effected by selecting AVE N, ENV, and ENV N with the selected function being initiated. Or, the AVE function can be terminated by pressing the AVE key a second time in which case the 7D20 would resume normal digitizing operation.

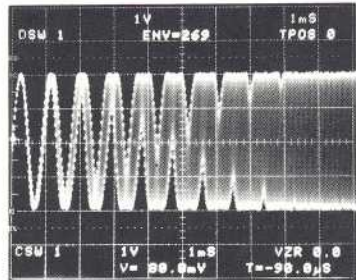


3857 131

Press	Remarks
ENV	<p>The HOLD function is cancelled, the ENV key illuminates and the ENVELOPE function is initiated. The sine wave will be continuously enveloped and the readout displays the accumulated total of times the waveform is ENVELOPED. While ENVELOPING, slowly adjust the frequency of your signal generator and observe the results. Selection of the ENV function automatically cancels HOLD, AVE, AVE N, ENV N, or SET N.</p>



VECTOR	<p>By connecting adjacent samples, the VECTORS paint in the envelope to aid in visual interpretation.</p>
--------	---



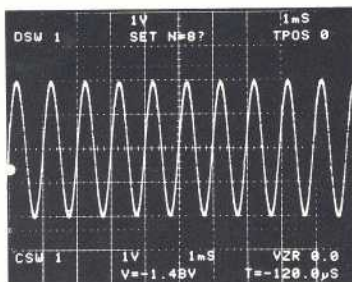
HOLD	<p>Terminates the ENV process and displays the total number of times the waveform was enveloped. As was the case with the AVE function, you can also terminate the ENV mode with the HOLD NEXT function which would allow one additional triggered waveform to be enveloped before automatically entering HOLD and displaying the accumulated total in the readout. You can also terminate the ENVELOPE mode by selecting AVE, AVE N, or ENV N with the</p>
------	---



Press	Remarks
HOLD (cont)	initiating of the function selected. The ENV mode can also be cancelled by pressing the ENV key a second time. The key will extinguish and the 7D20 will resume normal digitizing operation.

Display

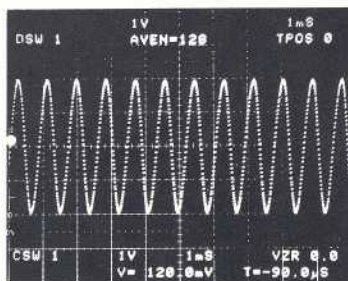
SET N	The SET N key will blink when pressed and the SET N? prompt appears in the display readout. The prompt will indicate the current value of N as shown. You can change the value of N by pressing and holding the SET N key. This will cause N to cycle repeatedly through this sequence of values: 8, 16, 32, 64, 128, and 256. Release the key to stop at the desired value of N. For the purpose of this exercise set the value of N to 128. The SET N prompt will disappear from the readout when you press the next key (this could be any key).
-------	---



3857-134

*f* The *f* key begins to blink indicating that you can now select a shift function.

AVE N	Selects the AVE N function. The key illuminates and the sine wave is averaged 128 times. Notice that the readout begins the AVE N count at the value of N and counts down to 0; at which time the HOLD function is initiated. The readout then changes to indicate that the waveform being displayed has been averaged 128 times. The AVE N function can be cancelled by
-------	--



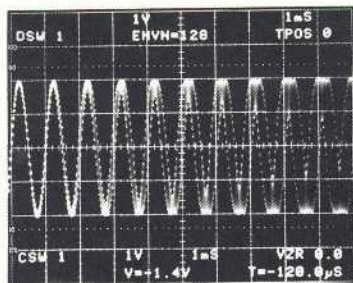
3857-135

Press	Remarks
AVE <b>N</b> (cont)	selecting HOLD, HOLD NEXT, AVE, ENV, or ENV <b>N</b> with the selected function being initiated.

**f**  
ENV **N**

Selects the ENV **N** function. The ENV **N** key illuminates and the sine wave is enveloped 128 times. As soon as you press ENV **N**, change the frequency of your signal generator slightly and observe the resulting display. Notice that just like the AVE **N** function, the readout begins the envelope count at 128 (the value of N) and counts down to 0. At the zero count the 7D20 automatically enters HOLD and the readout indicates an ENVELOPE count of 128. The ENV **N** mode will cancel with the selection of HOLD, HOLD NEXT, AVE, AVE **N**, or ENV with the selected function initiated. Note: The AVERaging and ENVELOping modes will function at all TIME/DIV settings. However, when using ROLL mode, some special features are provided. Refer to AVERaging, ENVELOping, and SET N functions in the Detailed Operating Instructions.

## Display



3857-136

## EXERCISE 9 DIGITIZING MODES AND THE TIME/DIV CONTROL

The 7D20 has four digitizing modes automatically selected by the TIME/DIV selector. These different digitizing modes ensure optimum digitizer operation for the full range of TIME/DIV settings. These modes are described here but for more information, refer to the Detailed Operating Instructions in this section and to the Operational Theory, section 3.

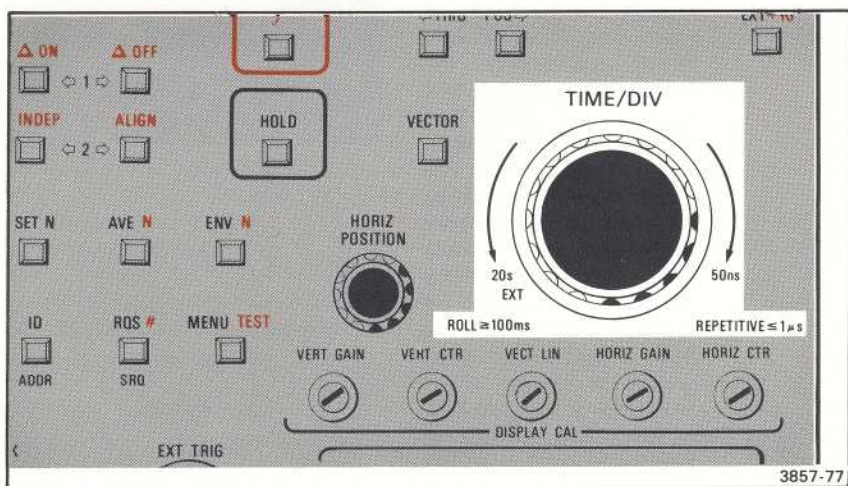


Figure 2-34. The TIME/DIV—digitizing modes selector.

### REAL-TIME AND EXTENDED REAL TIME

So far we have been using the real-time digitizing mode which extends from 50 ms/div to 500  $\mu$ s/div. The extended real-time mode extends from 200  $\mu$ s/div to 2  $\mu$ s/div. The primary difference between these two modes is that the record length (points per waveform) decreases from 100 points/div to 80 points/div in the extended real-time mode. See Detailed Operating Instructions.

### ROLL MODE AND EXTERNAL CLOCK

To demonstrate the roll mode, adjust your sine-wave signal to 1-Hz, DC coupled, and set the TIME/DIV selector for 100 ms/div. You will observe that the signal display rolls across the crt. This roll motion can be stopped at any time by pressing the HOLD key. The last digitized point always appears at the extreme right of the display. As you change from 100 ms/div to 20 s/div with the display not in HOLD, note that the digitizing process continues in the roll mode. To use the EXTERNAL CLOCK

function, the TIME/DIV control must be set to EXT, and a signal applied to the EXT CLOCK input connector. Refer to Exercise 12, Using the External Clock for further information.

### **EQUIVALENT TIME MODE**

In the equivalent time digitizing mode several trigger events are needed to digitize a full record (1024 waveform data points) length. This can be demonstrated by setting the TIME/DIV to 1  $\mu$ s/div and changing the sine-wave signal to 100 kHz. Make abrupt adjustments of the CH 1 POSITION control to vertically reposition the waveform by several divisions (either up or down). When you do this you will notice a short time lag occurs and the momentary presence of multiple traces as the digitizer completes the record at the changing POSITION control settings. Decreasing the TIME/DIV makes this more obvious as the digitizer requires more triggers to digitize the waveform at a new position. Refer to the Detailed Operating instructions in this section and to the Operational Theory, section 3, for more information.



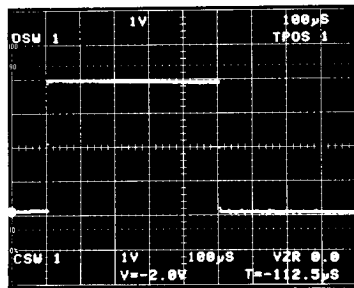
## EXERCISE 10 SETTING THE TRIGGER POSITION

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

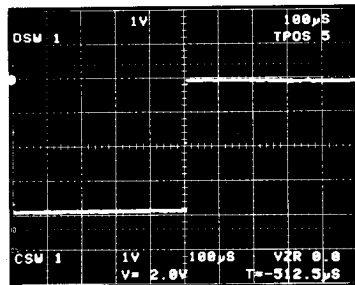
With the TRIG POS keys (shown in Fig. 2-35), the 7D20 permits you to view up to 10 divisions of pretrigger and up to 1500 divisions of post trigger. Pretrigger allows you to see events that occur before the trigger event while post triggering acts like a digital delay which allows you to view events that occur after the trigger event. The trigger position setting appears in the upper right corner of the display readout just below the TIME/DIV setting as shown in Figure 2-36. For this exercise, instead of the sine wave you've been using, connect the calibrator signal from the host mainframe to the CH 1 input connector of the 7D20. Set the calibrator for a 1-kHz, 4 volt signal output and set the 7D20 TIME/DIV for 100  $\mu$ s. Initialization set the 7D20 to trigger on the positive slope of the incoming signal so the display should appear as shown in Figure 2-36. Proceed with the exercise.

Press	Remarks
TRIG POS $\Rightarrow$	One press of this key causes the TRIGger POSition to move one full division to the right of its zero position. Four more presses will step the TRIG POS to the center of the display. Therefore, half of the display shows pretrigger and half shows post triggered waveform.

Display



3857-180



3857-181

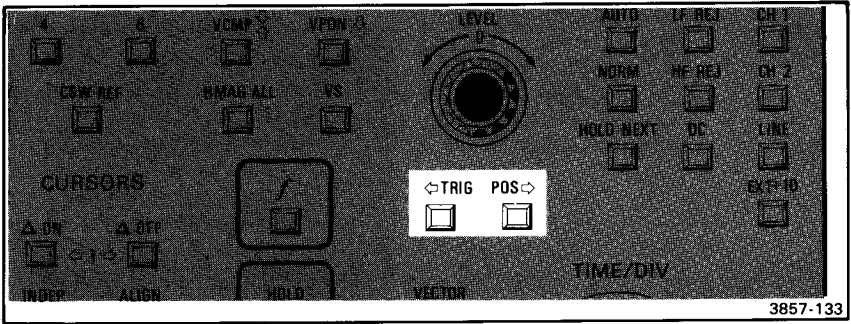


Figure 2-35. TRIGger Position keys.

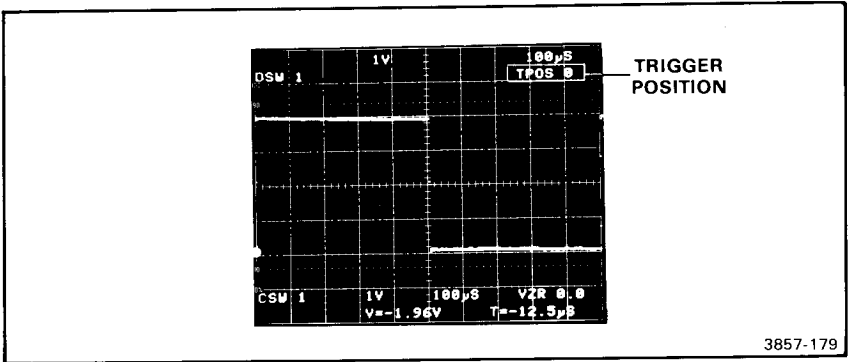
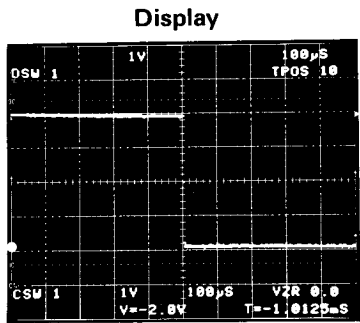


Figure 2-36. Trigger POSition readout.

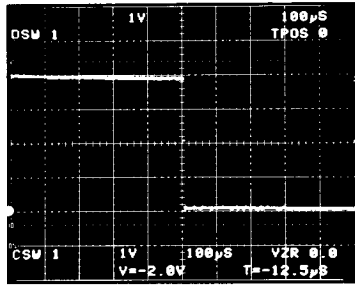
Press	Remarks
TRIG POS⇄	Press and hold this key and the TRIG POS will increment to 10. This results in a full 10-division display of information which occurred prior to the trigger event.



**Press                      Remarks**

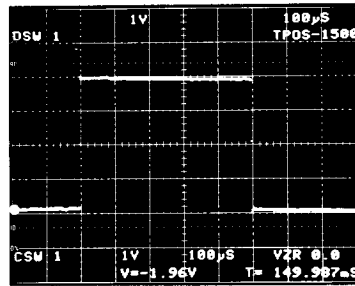
⇐TRIG POS    Press and hold this key and the TRIG POS increments back to zero where it automatically stops. Release the key.

**Display**



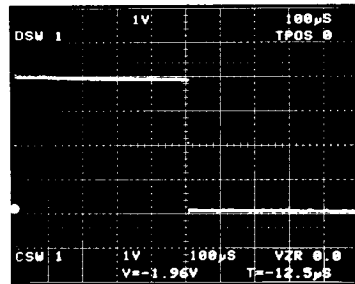
3857-183

⇐TRIG POS    Press and hold this key and again the TRIG POS will decrement to -1500 divisions of post trigger. The trigger event actually occurs 1500 divisions (150 ms) before the beginning of this waveform.



3857-184

TRIG POS⇒    Press and hold this key to automatically return the TRIG POS to zero.



3857-185

## EXERCISE 11 USING THE HOLD NEXT

So that you may start at any of the 12 Get-Acquainted Exercises, the Initialization feature has been used as necessary to set front-panel control settings to a known state by a single selection of INIT FRONT PANEL from the UTILITIES menu (refer to the list of predetermined settings in the Detailed Operating Instructions under Menu Functions). Under normal applications it is not necessary to use the Initialization feature, but it may be helpful.

The HOLD NEXT feature (HOLD NEXT key is shown in Fig. 2-37) is used to capture and hold digitized waveforms. Generally speaking, when HOLD NEXT is in effect, your 7D20 will accept the next trigger event, digitize a full memory of waveform data, then automatically enter HOLD. However, its actual operation varies somewhat depending upon the digitizing mode that is being used. (The digitizing modes are demonstrated in EXERCISE 9.) The HOLD NEXT mode can be used to acquire data from a single trigger event just as the single sweep mode is used in a conventional oscilloscope.

Initialize the 7D20 and perform the following steps to illustrate how HOLD NEXT is used in the different digitizing modes.

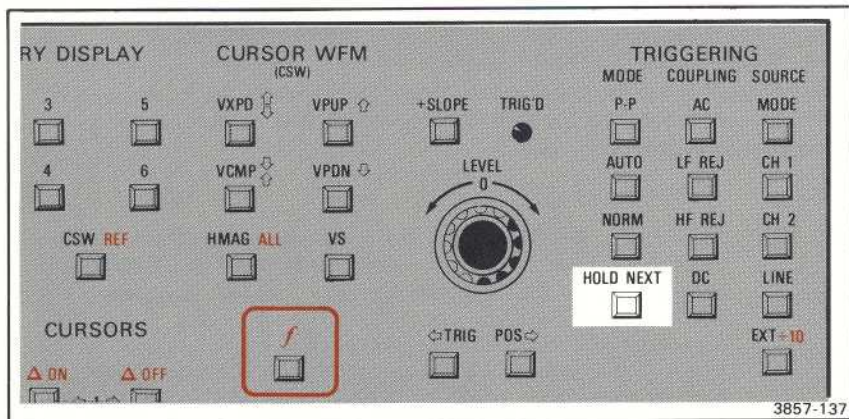


Figure 2-37. The HOLD NEXT key.

**REAL-TIME AND EXTENDED REAL TIME (50 ms  $\geq$  Time/Div  $\geq$  2  $\mu$ s):**

<b>Press</b>	<b>Remarks</b>
HOLD NEXT	Note that the 7D20 goes into HOLD and that the TRIG'D light goes out. When you first pressed the HOLD NEXT key the 7D20 simply accepted one more trigger, digitized a full memory of waveform data, then entered HOLD. Each subsequent press of the HOLD NEXT key arms the circuits to accept a new trigger and repeats the process. The TRIG'D light illuminates when the trigger occurs. This is true for the real-time digitizing mode (50 ms/div to 500 $\mu$ s/div) and for the extended real-time digitizing mode (200 $\mu$ s/div to 2 $\mu$ s/div).

**EQUIVALENT TIME MODE (1  $\mu$ s  $\geq$  TIME/DIV  $\geq$  50 ns):**

For this part set the TIME/DIV for 100 ns/DIV and adjust the incoming sine-wave signal to 1 MHz.

<b>Press</b>	<b>Remarks</b>
HOLD NEXT	Notice the 7D20 display momentarily updates before entering HOLD. In the equivalent time mode (1 $\mu$ s/div to 50 ns/div), when the HOLD NEXT key is pressed, the 7D20 accepts enough repetitive triggers to build a complete representation of the waveform before entering HOLD. The TRIG'D light stays illuminated during each triggered digitizing period. Refer to the Operational Theory, Section 3 for further information.

**ROLL MODE AND EXTERNAL CLOCK (20 s - 100 ms)**

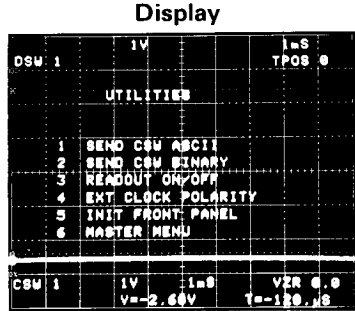
Select DC coupling on CH 1 and set the TRIGGERING MODE for NORM with DC COUPLING and HF REJ. Set the TIME/DIV control for 100 ms/div and change the input signal to a 1-Hz sine wave.

<b>Press</b>	<b>Remarks</b>
HOLD NEXT	Observe that the HOLD NEXT key illuminates when pressed but the display continues to roll for a short time. Then, the TRIG'D light comes on and a full 10-divisions of waveform rolls past before the 7D20 enters HOLD and the TRIG'D light goes out. Subsequent pressing of the HOLD NEXT key will produce the same result. Notice in doing so, that the 7D20 triggers on the same point and slope of the waveform each time. When HOLD NEXT is activated in the roll mode, a hold off period occurs while the circuits are armed. This allows about a screen full of data to roll across before a trigger can be accepted. When a trigger is accepted, the TRIG'D light comes on and the displayed waveform continues to roll across the screen until the selected TRIG POS is reached. If the TRIG POS is 0 then a full screen of data will cross the screen before HOLD is entered and the TRIG'D light goes out. Set the TRIG POS to center screen and only half a screen of data will roll across after a trigger is accepted. HOLD NEXT operates this way throughout the range of roll digitizing mode (20 s/div to 100 ms/div and EXT clock).

## EXERCISE 12 USING THE EXTERNAL CLOCK

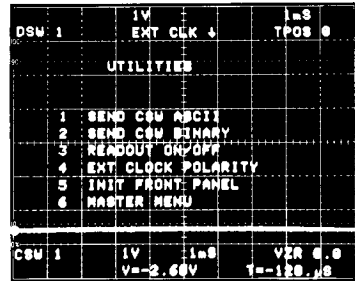
EXTERNAL CLOCK digitizing mode permits you to digitize signals using an external clock to determine when digitized point values are stored in waveform memory. Polarity of the EXTERNAL CLOCK is set by using the MENU feature of your 7D20 as instructed below:

<b>Press</b>	<b>Remarks</b>
MENU, 4	Selects the UTILITIES menu.

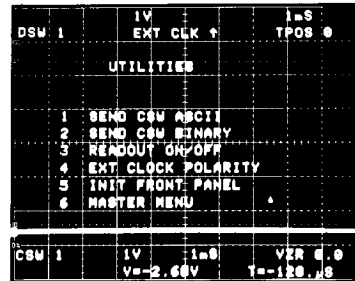


3857-138

4 Select EXT CLOCK POLARITY from the UTILITIES menu. EXT CLK ↓ prompt will appear in the display. The prompt arrow indicates the polarity setting of the EXTERNAL CLOCK. Each subsequent press of the 4 key will reverse the polarity setting so long as the UTILITIES menu is being displayed. Set the EXT CLK polarity to positive (↑). You can clear the prompt by pressing any other key. However, if you press the menu key the prompt and the displayed menu will both be cleared from the display.



3857-139

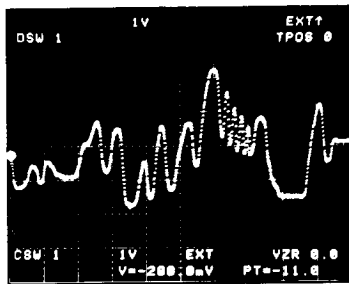


3857-140

Now, connect your signal generator to the EXT CLOCK input located in the lower part of the front panel of the 7D20. Apply a 5 volt signal.

Press	Remarks	Display
4 (cont)	Select 100 Hz frequency, this can be either a square or sine wave. Rotate the TIME/DIV control counter-clockwise until "EXT ↑" is displayed in the upper right corner of the readout.	
	The waveform display should be rolling. If not, be sure you are not in HOLD. While turning the CH 1 POSITION control back and forth, vary the signal generator frequency about 2 or 3 times slower or faster and notice that the speed of the ROLLING also varies.	

HOLD	Enters HOLD and the waveform stops moving. Observe the cursor's horizontal coordinate no longer reads in seconds. Since the EXT CLOCK may not always be a calibrated or constant frequency source, the cursor readout simply indicates point numbers instead of time.
------	---



3857-142

This concludes the GET-ACQUAINTED EXERCISES. For additional information, refer to the DETAILED OPERATING INSTRUCTIONS that follows in this section.

# DETAILED OPERATING INSTRUCTIONS

## INTRODUCTION

Detailed information concerning the controls and operation of the 7D20 is given in the following pages.

## POWER-UP

At power-up the 7D20 automatically performs an internal self test. Upon completion of its self test, the 7D20 restores many of its control settings which were retained in memory when power was last removed. Table 2-1 shows what settings are restored and which ones return to a power-up default condition.

**TABLE 2-1**  
**Control Settings at Power Up**

- 
- I. Control settings restored upon power-up:
- A. CH1/CH2
    - VOLTS/DIV
    - COUPLING
    - INVERT (CH2)
  - B. TRIGGERING
    - MODE
    - COUPLING
    - SOURCE
    - SLOPE
    - POSITION
  - C. HORIZONTAL
    - TIME/DIV
    - EXTERNAL CLOCK POLARITY
  - D. MEMORY DISPLAY
    - DISPLAY 1
    - DISPLAY 2
    - CURSOR WFM # (1 or 2)
    - REFERENCE WFM**
  - E. CURSOR WFM (If WFM 1 or WFM 2 at power down.)
    - VPUP, VPDN (For HMAG and VS with **REF** operation.)
    - HMAG, VS
  - F. ACQUISITION
    - AQR MODE
  - G. CURSORS
    - MODE (**INDEP** or **ALIGN**)
    - △ON/△OFF**
    - CURSOR POSITIONS
- 

TABLE CONTINUED



TABLE 2-1 (CONT)  
Control Settings at Power Up

---



---

H. GPIB	ADDRESS
	MODE
	TERMINATOR
I. OTHER	VECTOR
	SET N VALUE
II. Defaults of control settings not restored at Power Up.	
A. CH1/CH2	VARIABLE GAIN CONTROLS:ACTIVE
	POSITION CONTROLS:ACTIVE
B. TRIGGERING	TRIGGER LEVEL CONTROL:ACTIVE
C. HORIZONTAL	HORIZ POSITION CONTROL:ACTIVE
D. MEMORY DISPLAY	DISPLAY 3-6:OFF
	CSW: If CURSOR WFM was 3-6 it is reset to CH1 if AQR MODE is CH1, BOTH, or ADD, and to CH2 if AQR MODE is CH2.
	COPY:OFF
E. CURSOR WFM	VXPD, VCMP:OFF
	VPUP, VPDN:OFF (Except separation for HMAG or VS)
	For all WFM's other than CSW, VXPD, VCMP, VPUP, VPDN, HMAG and VS are off.
F. ACQUISITION:	No default settings.
G. CURSORS:	No default settings.
H. GPIB:	See GPIB section.
I. OTHER	AVE, AVE N:OFF
	ENV, ENV N:OFF
	MENU:OFF
	RQS, RQS#:OFF
	ID:OFF
	HOLD:OFF
	<i>f</i> :OFF

---

## MENU FUNCTIONS

The 7D20 MENU allows you to select a number of functions using the numbered (1-6) MEMORY DISPLAY keys. When pressed, the MENU key will illuminate and cause the last selected MASTER MENU (see Fig. 2-38) or submenu to be displayed. When initially selected after power-up it displays the MASTER MENU. Selecting MENU terminates TEST menus, ID, and clears any displayed text. The MENU is terminated by pressing the illuminated MENU key or by TEST, ID, or TEXT from GPIB. The illuminated MENU key will extinguish upon termination and the displayed menu will disappear. The following shows the 7D20 menus and explains their associated function selections.

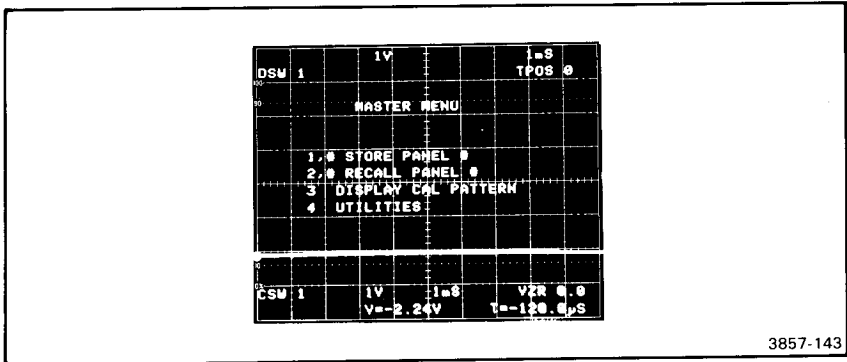


Figure 2-38. 7D20 MASTER MENU.

### MASTER MENU SELECTIONS

#### 1,# STORE PANEL #

This Menu function (see Fig. 2-38) permits you to store up to six sets of front panel settings. These settings are retained in memory at power down and can be recalled using menu item 2, which is described next. Table 2-2 gives the items that are stored, other items retain their current or most recent status before a recall command.

To store front panel settings, press MEMORY DISPLAY key 1. The prompt display field will read: STORE #, with the # sign blinking. At this point, you can press any of the numbered (1-6) MEMORY DISPLAY keys and the current front panel settings will be stored in the respective front panel memory. This in no way interferes with the storage of waveform data. Settings previously stored in the selected memory are replaced with the current settings. The store function is terminated upon completion of the command.

**TABLE 2-2**  
**Front Panel Settings that can be Stored and Recalled.**

A. CH1/CH2 VOLTS/DIV COUPLING INVERT (CH2)	D. MEMORY DISPLAY DISPLAY 1-6 CURSOR WFM#
B. TRIGGERING MODE COUPLING SOURCE SLOPE POSITION	E. CURSOR WFM NONE
C. HORIZONTAL TIME/DIV EXTERNAL CLOCK POLARITY	F. ACQUISITION AQR MODE
	G. OTHER VECTOR SET N VALUE

**2.# RECALL PANEL #**

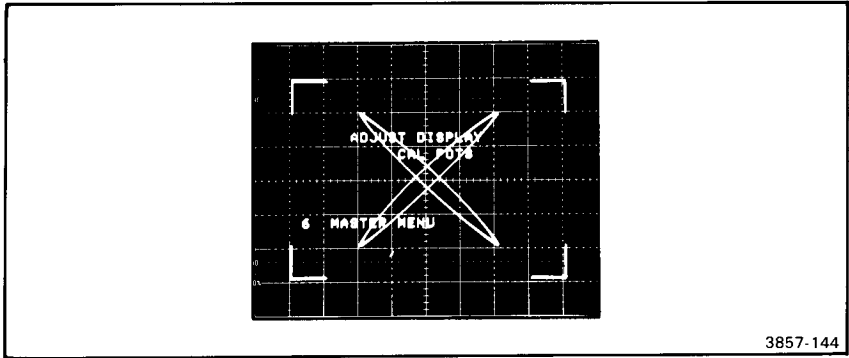
This Menu function allows you to recall any set of front panel settings which have been stored as described previously. Select this function by pressing MEMORY DISPLAY key 2. The prompt: RECALL # will appear in the display. The # sign will blink indicating that you should now select the numbered MEMORY DISPLAY key corresponding to the settings you wish to recall. Upon pressing the desired key, the 7D20 will immediately reset to the settings stored in the selected memory. The recall function terminates upon completion of the command.

**3 DISPLAY CAL PATTERN**

This menu selection is activated by pressing MEMORY DISPLAY key 3, which causes the DISPLAY CAL PATTERN shown in Figure 2-39 to be displayed. The DISPLAY CAL PATTERN is used with the DISPLAY CAL adjustments on the front panel to compensate the 7D20 display for the calibration of the host mainframe. Adjustment of these controls affect only the display and has no effect on the digitizing accuracy of the 7D20. Included in the DISPLAY CAL PATTERN is a VECTOR display that forms a flower pattern. Optimum VECTOR displays are achieved when the flower pattern is adjusted (using the VECT LIN adjustment) for a closed X pattern. The complete procedure for adjusting the DISPLAY CAL adjustments is given in the preliminary set-up portion of the "Get-Acquainted Exercises". To terminate the DISPLAY CAL PATTERN press MEMORY DISPLAY key 6 which returns the MASTER MENU to the display or press the illuminated MENU key which terminates the MENU mode. Instrument status and waveform data is retained during this process.

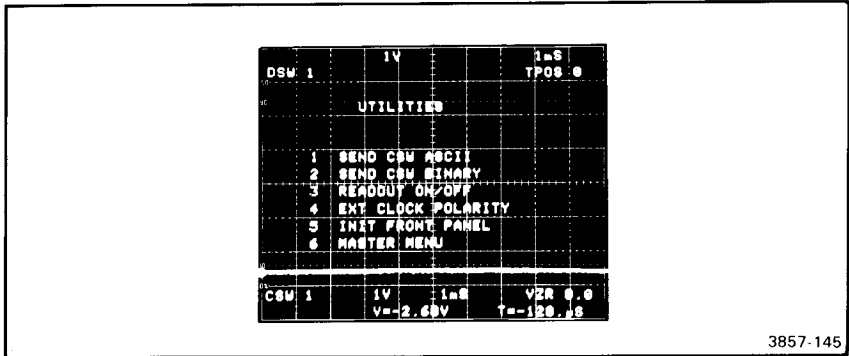
**4 UTILITIES**

Pressing MEMORY DISPLAY key 4 causes the UTILITIES submenu, shown in Figure 2-40, to be displayed. As with the MASTER MENU functions, selections from the UTILITIES MENU are made by pressing the numeric MEMORY DISPLAY key that corresponds to the menu selection you want. The UTILITIES MENU can be terminated by pressing MEMORY DISPLAY key 6, which returns the MASTER MENU, or by pressing the illuminated MENU key which terminates the MENU mode.



3857-144

Figure 2-39. DISPLAY CAL PATTERN.



3857-145

Figure 2-40. UTILITIES MENU.

### Utilities Menu Selections

**1 SEND CSW ASCII.** The selection of this function from the UTILITIES menu transmits the cursor waveform preamble and ASCII curve data over the GPIB. This command is provided for use with other GPIB instruments in a listen only mode where no controller is present in the system (refer to the GPIB Section 4). 7D20 must be in talk only mode.

**2 SEND CSW BINARY.** The selection of this function transmits the cursor waveform preamble and binary curve data over the GPIB. This command is also provided for use with other GPIB instruments in a listen only mode where no controller is present in the system (refer to GPIB Section 4). 7D20 must be in talk only mode.

**3 READOUT ON/OFF.** This selection from the UTILITIES menu will turn off or on the display readout (lines 1, 2, 15, & 16). Menu and text lines 3-14 are not affected.

**4 EXT CLOCK POLARITY.** This UTILITIES menu selection displays the current status of the EXTERNAL CLOCK polarity in the prompt field as shown in Figure 2-41. The message "EXT CLK ↓" or "EXT CLK ↑" is displayed (depending on current polarity status) and the arrow will be blinking. The EXT CLOCK polarity will reverse with each additional press of MEMORY DISPLAY key 4.

**5 INIT FRONT PANEL.** This MENU feature is included mainly to simplify the Get-Acquainted Exercises but may be used any time that you wish to set the front panel controls to a known state. Its selection, either from the UTILITIES menu or from a command over the GPIB will clear WFM memories 4 through 6, and will initialize the 7D20 front panel to the following predetermined settings:

- |  |  |
|--|--|
| <p>A. CH1/CH2<br/>           VOLTS/DIV:1<br/>           POSITION CONTROLS:ACTIVE<br/>           VARIABLE CONTROLS:ACTIVE<br/>           COUPLING:AC<br/>           CH2 INVERT:OFF</p>                                  | <p>E. CURSOR WFM<br/>           CSW:1<br/>           VXPDP, VCMP:OFF<br/>           VPUP, VPDN:OFF<br/>           HMAG, VS:OFF</p>   |
| <p>B. TRIGGERING<br/>           MODE:P-P<br/>           HOLD NEXT:OFF<br/>           COUPLING:AC<br/>           SOURCE:MODE<br/>           SLOPE:POS<br/>           LEVEL CONTROL:ACTIVE<br/>           POSITION:0</p> | <p>F. ACQUISITION<br/>           AQR MODE:CH1</p>  |
| <p>C. HORIZONTAL<br/>           TIME/DIV:1 mS<br/>           POSITION CONTROL:ACTIVE<br/>           EXT CLOCK POLARITY:POS</p>   | <p>G. CURSORS<br/>           MODE:ALIGN<br/>           ΔOFF:ON<br/>           CURSOR 1:POINT 1<br/>           CURSOR 2:POINT 1024</p>  |
| <p>D. MEMORY DISPLAY<br/>           DISPLAY 1:ON<br/>           DISPLAY 2-6:OFF<br/>           CSW:OFF<br/>           COPY:OFF<br/>           REF:OFF</p>  | <p>H. GPIB<br/>           NONE</p>   |
|  | <p>I. OTHER<br/>           AVE, AVE N:OFF<br/>           ENV, ENV N:OFF<br/>           MENU:OFF<br/>           RQS, RQS #:OFF<br/>           ID:OFF<br/>           HOLD:OFF<br/>           f:OFF</p> |

**6 MASTER MENU.** This selection from the UTILITIES MENU returns the MASTER MENU to the display.

## TEST MENU

Each time the 7D20 is powered up it automatically executes an internal "Self Test". This self test can, however, be run anytime at the discretion of the operator. To manually initiate SELFTEST, first press the *f* (shift function) key, then press the MENU TEST key. This will cause the TEST MENU shown in Figure 2-42 to be displayed. Next, press the 1 key (MEMORY DISPLAY) to select EXECUTE SELFTEST from the TEST MENU. The SELFTEST will then execute (front panel keys will blink, etc.) and, upon completion, the SELFTEST PASS message will appear in the display as shown in Figure 2-43 and the TEST MENU will extinguish. The 7D20 will reset to

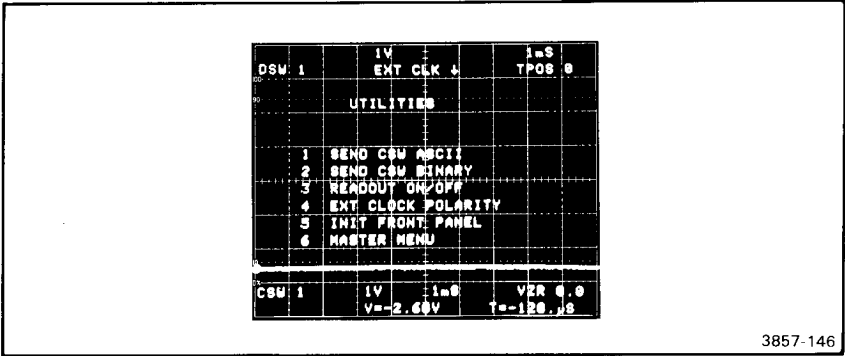


Figure 2-41. External Clock prompt.

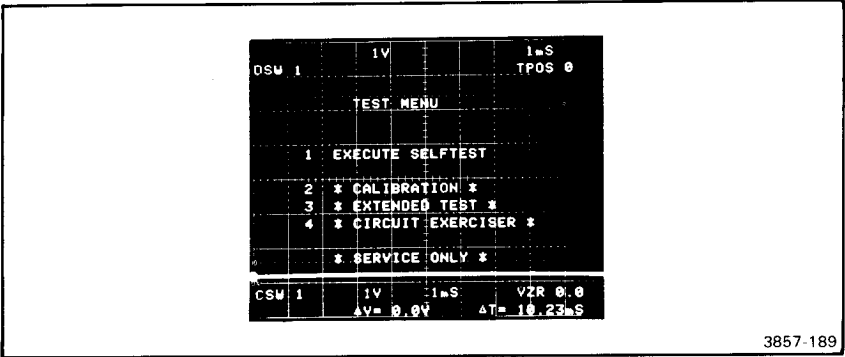


Figure 2-42. TEST MENU.

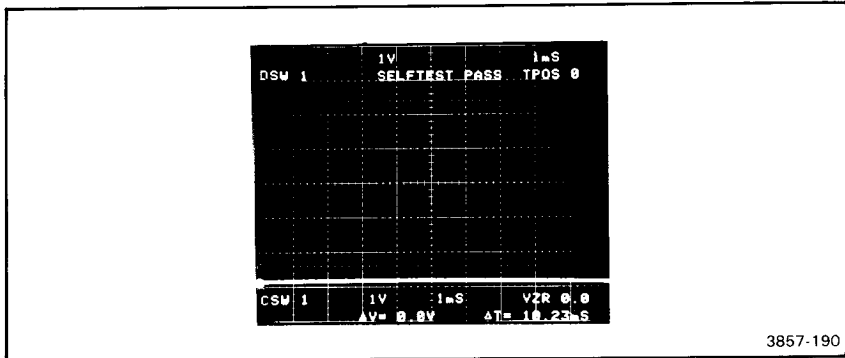


Figure 2-43. SELFTEST PASS message.

the state it was in before SELFTEST was initiated and all waveform memories will be initialized.

Should the SELF TEST fail, the appropriate message will also appear in the display along with the number of the failed test as shown in Figure 2-44. This information should be reported to a qualified service person for repair.

SELF TEST can be continued by pressing MEMORY DISPLAY key 1 in response to the Failure menu displayed when a self test failure occurs. This will continue the Self Tests but the number of the failed test will remain in the display. To leave the Self Test mode after a failure occurs, press MEMORY DISPLAY key 2 to EXIT SELFTEST. The Prompt and Error Message Field will display the SELFTEST FAIL message.

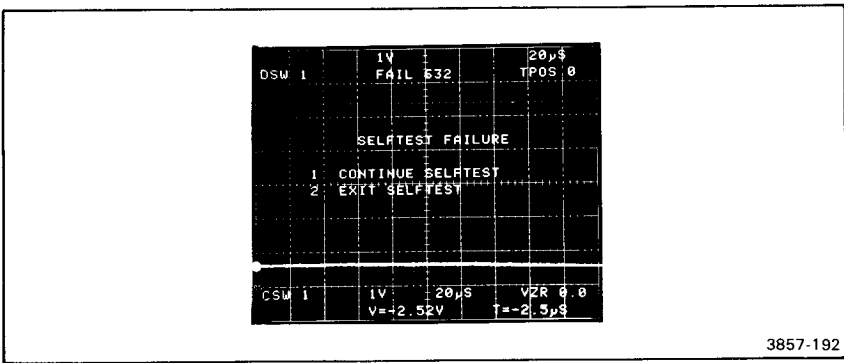


Figure 2-44. Self Test Fail.

Items 2, 3, and 4 on the TEST MENU are service related items. Use of these tests is intended only for qualified service personnel and are explained in the 7D20 Service Manual. Should you somehow get the 7D20 into one of these test routines and can't get out, you may need to turn the power off then back on to restore normal operation.

## PROMPTS AND WARNINGS

The 7D20 displays prompts and warnings in response to certain actions or conditions. The following is a complete list of these prompts and warnings along with descriptions of why they will be displayed.

Prompt	Description
SET N=XXX?	Appears in response to pressing the SET N key. The X's will indicate the current value of N. Press and hold the key to cycle through the possible values of N.

Prompt	Description
COPY #→#	Appears in response to pressing the COPY key. This requires entry of the number of the waveform to be copied and the number of the waveform destination.
CSW #	Appears when the CSW key is pressed. This requires the number of the desired cursor waveform.
CSW X VS #	Is displayed when the VS key is pressed. This requires that you enter the number of the waveform you wish to display versus the CSW. The cursor waveform number is X.
STORE #	This prompt appears in response to the selection of item 1 on the MASTER MENU. Press the number of the memory in which you wish to store the front-panel settings.
RECALL #	This prompt appears in response to the selection of item 2 on the MASTER MENU. Press the number of the MEMORY DISPLAY key where front-panel settings were previously stored and those settings will be automatically restored.
EXT CLK↑ or EXT CLK↓	Appears when item 4 on the UTILITIES menu is selected. The direction of the arrow indicates the current polarity setting of the EXTERNAL CLOCK. Polarity will reverse upon each subsequent press of MEMORY DISPLAY key 4 so long as the UTILITIES menu is displayed.
RQS #	This prompt appears in response to the selection of $f$ RQS #. The user completes the command by pressing 1-6 which gives RQS X prompt.
MODE=mode?	In this prompt, mode = OFF, T (talk), L (Listen), or T/L (Talk/Listen).
TERM=term?	In this prompt, term = EOI or LF/EOI.
ADDR=XX	In this prompt, XX = 0 - 30.
0-10 TPOS REQD	This warning is given in AVE or ENV ROLL Mode (or the EXTERNAL CLOCK) if a negative trigger position is selected.
HOLD REQD (HOLD REQUIRED)	This warning appears if VPUP $\uparrow$ , VPDN $\downarrow$ , VCMP $\frac{\circ}{\times}$ , or VXP $\frac{\circ}{\times}$ are pressed while the 7D20 is actively digitizing into the cursor waveform and REFERENCE is off.
HMAG, VS REQD	This warning is given when you attempt to select the REFERENCE waveform when the 7D20 is not displaying a HMAG cursor waveform (magnified cursor waveform) or is not in the VS (versus) mode.



Prompt	Description
CSW REQD	This warning is displayed if you attempt to turn off the cursor waveform.
UPDATE IGNORED	This warning appears if you attempt to change any of the parameters in the ID menu when the 7D20 is in the remote with lockout state.
RQS OFF	This warning occurs when the RQS or probe identify button is pressed and the RQS mask is turned off. The RQS mask can be turned off by the RQS OFF, PID OFF, or USER OFF GPIB commands. Refer to Section 4, GPIB.
TALK ONLY REQD	This warning occurs when send CSW Binary, or send CSW ASCII command is selected if the GPIB mode is not Talk Only.

The following messages are results of self test and diagnostic checks. In the event of a test failure message, refer the problem to a qualified service person.

SELFTEST PASS  
SELFTEST FAIL  
GPIB PASSED  
GPIB FAILED  
EAROM PASSED  
EAROM FAILED\*  
FAIL XX (where X = failed circuit number).

## DISPLAY CALibration

The 7D20 contains five (5) controls to compensate the 7D20 for the calibration of the host mainframe. These controls must be adjusted by the operator each time the 7D20 is installed in a new or different host mainframe. To adjust these controls it is necessary to call up a waveform that is permanently stored in the memory of the 7D20 and is accessed by using the MENU. This procedure is given at the beginning of this section in the preliminary set-up part of the "Get-Acquainted Exercises". Adjustment of these controls affects only the display and has no effect at all on the digitizing characteristics of the 7D20.

**VERT CTR**—This screwdriver control adjusts the 7D20 display output to match the vertical position tolerance of any Tektronix 7000-series mainframe.

**VERT GAIN**—This screwdriver control adjusts the 7D20 display output to match the vertical gain tolerance of any Tektronix 7000-series mainframe.

**HORIZ CTR**—This screwdriver control adjusts the 7D20 display output to match the horizontal position tolerance of any Tektronix 7000-series mainframe.

---

\*The EAROM FAILED message is also given when the RECALL command is given and the memory being retrieved contains bad data.

**HORIZ GAIN**—This screwdriver control adjusts the 7D20 display output to match the horizontal gain tolerance of any Tektronix 7000-series mainframe.

**VECT LIN**—This screwdriver control optimizes the quality of the VECTOR display from one host mainframe to another by compensating for their various delay line lengths.

## READOUT DISPLAY

The readout display of the 7D20 is generated internally and displayed on the crt of the host mainframe. There are four lines of characters reserved for the readout display. Two lines of characters are centered in the top graticule division (lines 1 and 2, for the current control settings and prompt field), and two lines of characters are centered in the bottom graticule division (lines 15 and 16, for the cursor waveform parameters). The functions of these four lines are detailed below.

### CURRENT CONTROL SETTINGS AND PROMPT FIELD

These readout lines (lines 1 and 2) are centered in the top graticule division; they display the current front-panel control settings and prompt messages, positioned as follows:

LINE 1	Not Used by 7D20	CH 1 VOLTS/DIV Setting	CH 2 VOLTS/DIV Setting	TIME/DIV Setting
LINE 2	Displayed Waveform Number	Prompt Field		Trigger Position

### CURSOR WAVEFORM PARAMETERS

These readout lines (lines 15 and 16) are centered in the bottom graticule division; they display information about the cursor waveform, and are positioned as follows:

LINE 15	Cursor Waveform Number	Cursor Waveform Vertical Scale Factor in VOLTS/DIV*	Cursor Waveform Horizontal Scale Factor in Time or VOLTS/DIV*	Cursor Waveform Vertical Zero Reference *
LINE 16	Not Used by 7D20	Cursor Vertical Coordinate	Cursor Horizontal Coordinate	

\*Also affected by waveform modifiers (e.g., VXPDP, VPUP, etc.)

## VERTICAL CONTROLS

With the exception of the channel 2 invert function all controls for both input channels are identical so only channel 1 controls will be discussed.

### SIGNAL CONNECTION

In general, probes offer the most convenient means of connecting signals to the 7D20 inputs. The 7D20 supports all Tektronix probe readout encoding. Refer to the Tektronix, Inc. catalog for probe selection.

Coaxial cables may also be used to connect signals to the 7D20 input connectors. However, cables can have a considerable effect on the accuracy of the displayed waveform. To maintain the original frequency characteristics of an applied signal, use only low-loss, high-quality coaxial cable. Also, cables should be terminated in their characteristic impedance. If this is not possible, use suitable impedance matching devices.

### INPUT CONNECTORS

These bnc connectors provide signal connection for their respective channels.

### VOLTS/DIV (VARIABLE)

The vertical signal component is determined by the signal amplitude, the attenuation factor of the probe, the setting of the VOLT/DIV switch, and the setting of the VARIABLE control. The VOLTS/DIV switch (both channels) selects calibrated vertical sensitivity settings from 5 V/DIV (counterclockwise) to 5 mV/DIV (clockwise) in a 1,2,5 sequence. The knob settings are displayed on the crt of the host mainframe in the positions shown in Figure 2-45. The VOLTS/DIV settings shown in the display apply only when the VARIABLE control is in the calibrated (detent) fully-clockwise position.

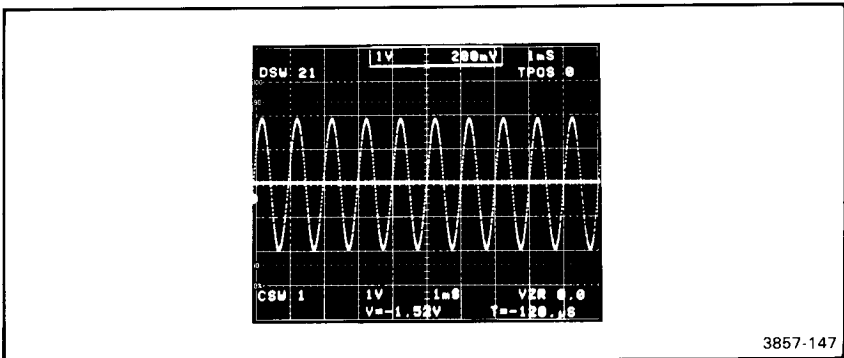


Figure 2-45. VOLTS/DIV readout.

The VARIABLE control provides continuously variable, uncalibrated settings between the calibrated steps of the VOLTS/DIV switch. With the VARIABLE control fully counterclockwise and the VOLTS/DIV set to 5 volts/div, the uncalibrated vertical sensitivity is extended to at least 12.5 volts/division. By applying a calibrated voltage source to the input connector, any specific vertical sensitivity value can be set within the range of the VARIABLE control. When the VARIABLE control is moved from its detent position, a ">" sign (see Fig. 2-46) will appear in the display next to the VOLTS/DIV readout to indicate an uncalibrated setting with greater attenuation.

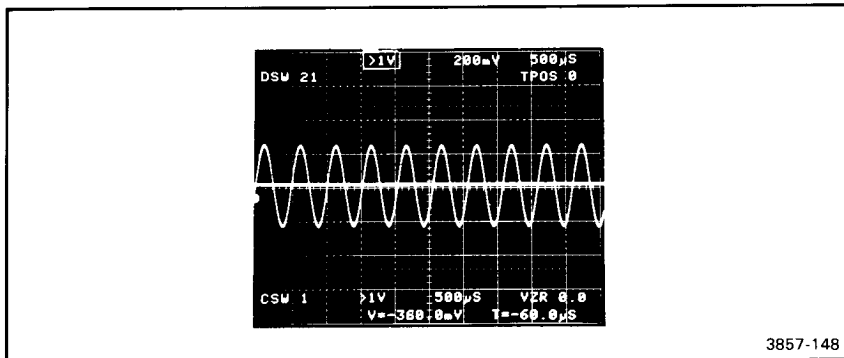


Figure 2-46. VARIABLE VOLTS/DIV readout indicator.

## POSITION

This control vertically positions the signal being acquired into channel 1 and channel 2 during digitizing. Its range is  $\pm 10$  div from center screen.

## COUPLING (INPUT)

The channel 1 and channel 2 coupling (AC-GND-DC) keys allow a choice of input coupling methods. The type of display desired and the applied signal will determine the coupling to use.

**AC**—This key illuminates when selected. With AC coupling the DC component of the applied signal is blocked by a capacitor in the input circuit. AC coupling provides the best display of signals with a DC component larger than the AC components above 30 Hz.

**DC**—This key illuminates when DC coupling is selected. DC coupling must be used to display the DC component of a signal. It must also be used to display AC signals below about 30 hertz (10 hertz with a 10X probe) and square waves with low-frequency components as these signals are attenuated with AC coupling.

**GND**—This key illuminates when GND coupling is selected. Ground coupling provides a ground reference at the input of the 7D20 without externally grounding the input connectors. The signals connected to the inputs are not grounded, and the

same DC load is presented to the signal source. Use GND coupling to determine VZR (vertical zero reference) level for the digitizer. When GND coupling is selected, the VZR reading updates in the display as the POSITION control is changed. To ensure that the VZR setting remains valid the POSITION control must not be moved after leaving GND position. This only affects the cursor readings with one cursor on.

### AQR (ACQUIRE) GAIN

This screwdriver adjustment varies the vertical gain of the signal being acquired (before the signal is digitized). To check the gain of either channel, set the VOLTS/DIV switch to 10 mV and connect a 40 mV, 1-kHz signal from the mainframe calibrator to the input of the channel being checked. Turn on cursor 2 and position the cursors so that cursor 1 is on the bottom of the waveform and cursor 2 is on the top as shown in Figure 2-47. The readout should indicate exactly 40 mV difference between cursor 1 and cursor 2. If not, adjust the CURSOR GAIN control till this reading is obtained. The 7D20 must not be in HOLD when making this adjustment. You may find it helpful to use the AVE mode when making this adjustment since this will average out any noise that might be present on the input signal.

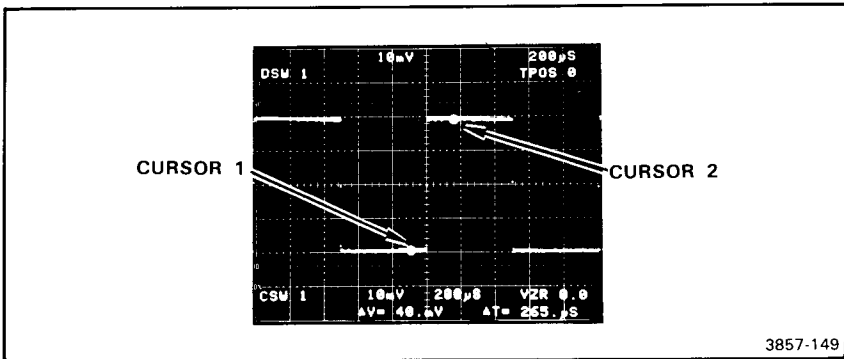


Figure 2-47. Cursor placement for AQR GAIN adjustment.

### CH2 INV

The Channel 2 INV key is used to electrically invert signals acquired through the channel 2 input connector. This function is selected by pressing the CH2 INV key. The key will illuminate and a down arrow "↓" will appear in the readout (see Fig. 2-48) as indications that the waveform being acquired through channel 2 is being inverted. When the invert function is not selected, the signal acquired through channel 2 will have the same polarity as the applied signal and a positive dc voltage will cause the displayed signal to move up in the display. When the invert function is selected a positive-going waveform at the channel 2 input will be acquired and displayed in inverted form and a positive dc voltage will move the displayed waveform down. The invert function is particularly useful in "added" operation when differential measurements are being made.

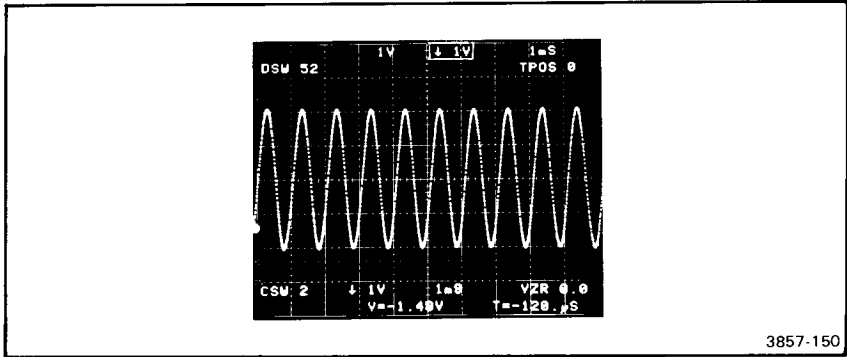


Figure 2-48. CH 2 INVert readout.

## ACQUIRE MODE

These four keys select the vertical acquisition mode by which your 7D20 acquires signal information. These keys illuminate when pressed to indicate the mode selected. Only one mode selection can be active at a time so the selection of any one of the AQR MODEs will cancel any AQR MODE previously selected.

### CH1 ⇨1

When selected, this mode enables your 7D20 to digitize the signal present at the CH 1 input connector into waveform memory 1. The selection of this mode automatically cancels a previously selected AQR MODE, the CSW is set to 1 (only if CSW was CH 2) and CH 2 display is off.

### BOTH

When this key is pressed your 7D20 is enabled to digitize signal information simultaneously, with full bandwidth, record length, and sample rate from both channels 1 and 2 inputs. Channel 1 signals are digitized into waveform memory 1 and Channel 2 signals are digitized into waveform memory 2. The selection of this mode automatically cancels a previously selected AQR MODE, and memories 1 and 2 are turned on.

### ADD ⇨1

This mode enables your 7D20 to acquire the signals connected to both channel 1 and 2 inputs added together. This added waveform is digitized and stored in waveform memory 1. The added waveform is shown in the mainframe display. Channel 2 waveform can be inverted for differential mode. Selection of this mode also cancels any previously selected AQR MODE. The CSW is set to 1 (only if CSW was CH 2) and the CH 2 display is off.

**CH2 ⇨ 2**

If selected while the 7D20 is actively digitizing (not in HOLD), this mode causes your 7D20 to digitize the signal present at the CH2 input connector into waveform memory 2. The selection of this mode automatically cancels a previously selected AQR MODE. The CSW is set to 2 (only if CSW was CH 1) and CH 1 is off.

**HOLD**

When this key is pressed the digitizing process is instantly terminated and the contents of both waveform memories 1 and 2 are simultaneously held unchanged. The display appears to freeze motion during HOLD because the waveform memory is not being updated with new data. The HOLD function is terminated by pressing the HOLD key a second time or by pressing AVE **N**, ENV, ENV **N**, and HOLD NEXT (the light will go out).

***f* (shift function)**

Certain keys have shift functions that are labeled with orange markings. These functions are selected by first pressing the *f* key. When pressed, the *f* key blinks to indicate that a shift function can be selected. Once the shift function is selected the *f* key light will go out and the selected function will be implemented.

If the *f* key is pressed and you then press a key that has no shift function, the *f* key will cancel (the light will go out) and the selected function will be implemented. The *f* key function can also be cancelled by pressing the blinking *f* key. The light will go out, indicating that the function is no longer active.

**TIME/DIV**

The TIME/DIV control establishes the digitizing rate and digitizing mode (see Table 2-3) of the 7D20. The TIME/DIV setting appears in the upper right corner of the display as shown in Figure 2-49. The TIME/DIV setting can be adjusted, in a 1, 2, 5 sequence from 20 seconds to 50 nanoseconds. When the TIME/DIV control is rotated counterclockwise past 20 seconds/div, the 7D20 is set for EXTERNAL CLOCKING, and EXT ↓ or EXT ↑ will appear in the display in place of the TIME/DIV readout as shown in Figure 2-50. Further rotation of the TIME/DIV control past EXT in the counterclockwise direction, or past 50 nanoseconds/division in the clockwise direction will have no effect on the TIME/DIV setting.

**DIGITIZING MODES**

The 7D20 uses four digitizing modes that are automatically determined by the TIME/DIV setting. Different digital sampling techniques are used in each digitizing mode to optimize the 7D20 performance capabilities depending upon the range of the TIME/DIV setting. The Roll, Real-Time, and Extended Real-Time modes

TABLE 2-3  
7D20 DIGITIZING MODES

DIGITIZING MODE TIME/DIV	Number of points per waveform	Number of triggers required for full waveform	Number of points sampled per single trigger
EQUIVALENT TIME 1 $\mu$ s—50 ns	1024	Multiple	50 ns $\approx$ 10 points 100 ns $\approx$ 20 points 200 ns $\approx$ 40 points 500 ns $\approx$ 100 points 1 $\mu$ s $\approx$ 200 points
EXTENDED REAL TIME 200 $\mu$ s—2 $\mu$ s	820	1	820
REALTIME 50 ms—500 $\mu$ s	1024	1	1024
ROLL, EXT CLK 20 s—100 ms	1024	1*	1024

\*Not required except to terminate HOLD NEXT.

sequentially digitize incoming waveform signals. In these digitizing modes a single trigger will produce a full memory of waveform data points in the HOLD NEXT mode. The Equivalent Time digitizing mode uses random sampling techniques to digitize higher frequency signals. Consequently, the Equivalent Time digitizing mode requires repetitive triggers to build a complete waveform in waveform memory. Table 2-3 gives the TIME/DIV range for each of these digitizing modes and the

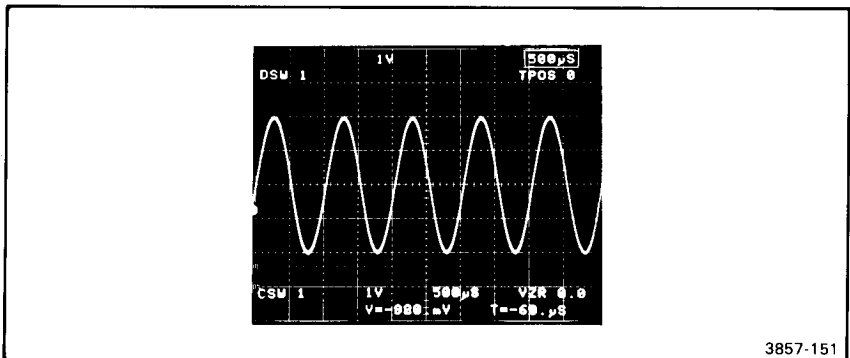
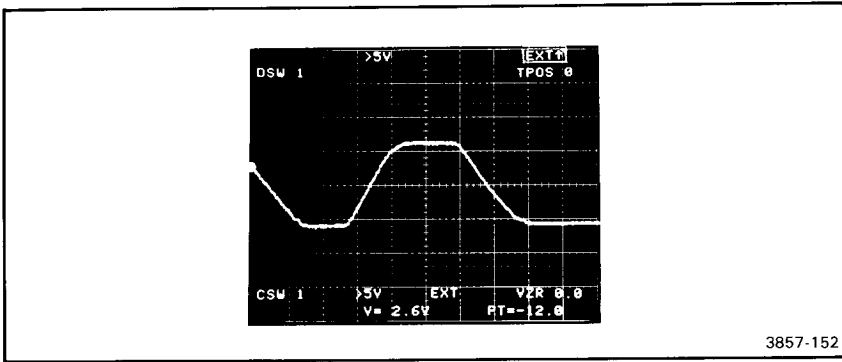


Figure 2-49. TIME/DIV readout.





3857-152

Figure 2-50. EXTERNAL clock readout.

number of waveform data points acquired for each for single trigger events. Refer to the Operational Theory, Section 3 for a more detailed discussion of the 7D20 digitizing modes.

### Extended Real-Time Digitizing (ERD)

In the ERD mode, in order to extend the real-time digitizing capability of the 7D20, the resolution is set to 80 points/division and digitizing is accomplished by interlaced differential sampling. (This is described more fully in Operational Theory,

TABLE 2-4  
Digitizing Mode Bandwidth

Digitizing Mode	Bandwidth (-3 dB)	Nyquist Frequency
ROLL	70 MHz	*
REAL-TIME	70 MHz	*
EXTENDED REAL TIME: TIME/DIV		
200 μs	100 kHz	200 kHz
100 μs	200 kHz	400 kHz
50 μs	400 kHz	800 kHz
20 μs	1 MHz	2 MHz
10 μs	2 MHz	4 MHz
5 μs	4 MHz	8 MHz
2 μs	10 MHz	20 MHz
EQUIVALENT TIME	70 MHz	*

$$* \text{Nyquist frequency} = \frac{100 \text{ points/div}}{(2)(\text{time/div})} = \frac{50}{\text{TIME/DIV}}$$

Section 3.) The resultant effect on bandwidth is shown in Table 2-4. It is important to note that in all digitizers, digitized signal frequencies which exceed the Nyquist limits will be aliased. The Nyquist frequencies for each digitizing mode are indicated in Table 2-4.

## TRIGGERING

The triggering features of the 7D20 are functionally grouped into three categories: SOURCE, COUPLING, and MODE. The associated triggering keys are arranged in a sequence which places the most-often used selections at the top of each column of keys. With this arrangement, stable acquisition and display can usually be obtained by pressing the top keys: P-P, AC, and MODE. When an adequate trigger signal is applied and the LEVEL control is correctly set, triggered acquisition is indicated by an illuminated TRIG'D indicator. If the TRIG'D light is not on, the trigger signal amplitude is inadequate, or its frequency is below the lower frequency limit of the AC coupling, or the 7D20 may be in HOLD. If proper acquisition is not obtained with these keys, other selections must be made.

In the ROLL digitizing mode the TRIG'D light does not illuminate because triggering is not used. An exception to this is when HOLD NEXT is used in the ROLL mode. Refer to the discussion of the HOLD NEXT function later in this section. The following discussions explain the triggering functions and how they are used.

### SOURCE

Keys in this column select the source of the trigger signal. The keys illuminate when selected.

**MODE**—The MODE key causes the trigger source to be determined by the AQR MODE selection. If the AQR MODE is channel 1, then the trigger source is also channel 1. If the AQR MODE is set for channel 2, then the trigger source will automatically set to channel 2. If the selected AQR MODE is ADD or BOTH the trigger source is forced to be channel 1. When MODE triggering is selected the appropriate CH1 or CH2 key will also illuminate. SOURCE MODE will cancel if another selection; CH1, CH2, LINE, or EXT (+10) is made.

**CH1**—This key permits you to select the channel 1 input for the trigger source. The key illuminates when selected and causes the trigger signal to be obtained from the signal applied to the channel 1 input. This provides a stable display of the signal applied to the channel 1 input.

**CH2**—This key permits you to select the channel 2 input for the trigger source. The key illuminates when selected and causes the trigger signal to be obtained from the signal applied to the channel 2 input. This provides a stable display of the signal applied to the channel 2 input.

**LINE**—Selection of LINE SOURCE connects a sample of the power-line voltage from the host mainframe to the trigger circuit. Line triggering is useful when the

input signal is time-related (multiple or submultiple) to the line frequency. It is also useful for providing a stable display of a line-frequency component in a complex waveform. The key illuminates when this function is selected.

**EXT**—Selection of this TRIGGERING SOURCE connects the signal from the EXT TRIG connector to the trigger circuit. However, the external signal must be time-related to the displayed waveform for a stable display. An external trigger signal can be used to provide a triggered display when the internal signal is either too low in amplitude for correct triggering or contains signal components on which triggering is not desired. It is also useful when signal tracing in amplifiers, phase-shift networks, wave-shaping circuits, etc. The signal from a single point in a circuit can be connected to the EXT TRIG connector through a probe or cable. This way acquisition is always triggered by the same signal which permits amplitude, time relationship, or waveshape changes of signals at various points in a circuit to be examined without resetting the triggering controls.

$f$  **EXT+10**—The EXT+10 function attenuates the external trigger signal by a factor of 10. Attenuation of high amplitude external trigger signals is desirable to increase the effective range of the LEVEL control. This function is selected after first pressing the  $f$  key, then the EXT+10.

## COUPLING

The triggering COUPLING keys select the way the trigger signal is connected to the trigger circuits. Each key permits selection or rejection of some frequency components of the trigger signal and each illuminates when selected. These coupling choices are discussed in the following paragraphs.

**AC**—AC coupling blocks the dc component of the trigger signal. Signals with low-frequency components below about 40 hertz (nominally -3 dB) are also attenuated. In general, AC coupling can be used for most applications. If, however, the signal contains unwanted frequency components or if acquisition is to be triggered at a low repetition rate or dc level, one of the other coupling choices will provide a better display.

**LF REJ**—This key is used with AC coupling to reject dc, and attenuate low-frequency trigger signals below about 40 kilohertz. Acquisition is, therefore, triggered by the higher-frequency components of the trigger signal. Low frequency rejection coupling is particularly useful for providing stable triggering when the trigger signal contains line-frequency components. Selecting LF REJ automatically selects AC coupling and illuminates the AC coupling key.

**HF REJ**—High frequency reject coupling rejects high-frequency signals above about 40 kilohertz. This can be selected with either AC or DC coupling.

**DC**—DC coupling is used to provide stable triggering from low-frequency signals. DC coupling can be used to trigger acquisition when the trigger signal reaches a dc level set by the LEVEL control.

## MODE

Keys in this column select the TRIGGERING MODE. The P-P mode is used for most applications because of the ease of obtaining triggered acquisition of waveforms. The AUTO, NORM, and HOLD NEXT modes are used for special situations and applications. The triggering mode functions are discussed in the following paragraphs.

**NORM**—This key selects the NORMAL mode of triggering. Whenever an adequate trigger signal is applied and the LEVEL control is correctly set, the NORMAL mode will provide triggered signal acquisition. When the trigger signal is not adequate or the LEVEL control is not properly set, acquisition is halted (TRIG'D light off). Even though acquisition is halted, contents of the waveform memory continue to be displayed. It is important to understand that when this happens, the displayed waveform memory contents are from the last triggered acquisition. Consequently, the signal currently being input is not the signal being displayed. The TRIG'D light is the best indication of this condition since it will not be illuminated. A new trigger will initiate a new acquire cycle which will update the waveform memory and illuminate the TRIG'D light. This triggering mode must be used to acquire signals with repetition rates below about 30 hertz.

**AUTO**—This key selects the AUTO triggering mode. AUTO triggering provides triggered signal acquisition when the LEVEL control is correctly set and when an adequate trigger signal is applied. The TRIG'D light indicates when signal acquisition is triggered.

The AUTO triggering mode is similar to the NORMAL mode except that loss or lack of an adequate trigger signal or an incorrect LEVEL control setting will not halt signal acquisition and the updating of waveform memory. In other words, what would appear as a HOLD condition (inadequate trigger signal) in the NORMAL triggering mode appears as free-run acquisition in the AUTO mode. When this happens, the free-run condition is visible in the mainframe display but the display doesn't appear stable because the trigger doesn't relate to the input waveform. An adequate trigger signal ends the free-running condition. A free-running condition can be useful when it is desired to measure only the peak-to-peak amplitude of a signal without observing the waveshape (such as bandwidth measurements).

**P-P**—This key selects peak-to-peak automatic triggering. This mode is the same as the AUTO TRIGGERING MODE except the range of the LEVEL control is restricted to within the peak-to-peak value of the trigger signal. This is the most frequently used triggering mode since it provides triggered acquisition for almost any setting of the LEVEL control whenever an adequate trigger signal is applied. The 7D20 internally ac couples the signal in this mode. High frequency and low frequency may still be selected.

The range of the LEVEL control in the P-P mode is between approximately 20% and 80% of the peak-to-peak amplitude of the trigger signal. The LEVEL control can be set so that the displayed waveform starts at any point within this range on either slope. The SLOPE control is discussed later in this section. The trigger circuits automatically compensate for a change in trigger signal amplitude. Therefore, if the

LEVEL control is set to start waveform acquisition at a certain percentage level on the leading edge of a low-amplitude signal, it triggers at the same percentage level on the leading edge of a high-amplitude signal if the LEVEL control is not changed. In this mode an inadequate trigger signal results in free-running acquisition just the same as described above for the AUTO triggering mode. When an adequate trigger signal is again applied, the free-run condition ends and triggered acquisition resumes. The P-P mode is particularly useful when observing a series of waveforms, since it is not necessary to reset the LEVEL control for each acquisition. However, P-P triggering is sometimes ineffective on low repetitive rate signals such as may be encountered in ROLL modes so NORMAL triggering should be used.

**HOLD NEXT**—The HOLD NEXT feature of the 7D20 allows you to capture and digitize single occurring events. The function is activated by pressing the HOLD NEXT key, which illuminates to indicate that the circuits are enabled to accept a trigger. Generally, once enabled, the 7D20 accepts the next occurring trigger event, digitizes a full memory of waveform data, then enters HOLD. However, operation of the HOLD NEXT function varies somewhat with the different digitizing modes as explained in the paragraphs that follow.

In the roll digitizing mode (20 S/div to 100 ms/div and EXT CLOCK) a hold-off period occurs when HOLD NEXT is activated, which allows the timing circuits to be enabled. During the hold-off interval a screen full of waveform data will roll past before a trigger will be accepted. Once the trigger event is accepted, the TRIG'D light comes on and stays lighted until the acquisition is complete and HOLD is entered. When HOLD is entered is determined by the TRIG POSition. If the TRIG POS is set for 0 a full 10-divisions of waveform will roll across the display before HOLD is entered. The entire waveform will then be post-trigger. Setting the TRIG POS at 5 will give 5 divisions of pre-trigger so when the trigger event occurs, only 5 divisions of waveform data will roll across the screen before HOLD is entered.

When the 7D20 is in the real time (50 ms/div to 500  $\mu$ s/div) or extended real time (200  $\mu$ s/div to 2  $\mu$ s/div) digitizing modes and HOLD NEXT is active (the HOLD NEXT key illuminated); the 7D20 will accept the next trigger event, (the TRIG'D light illuminates when this occurs), digitize a full memory of waveform data, then automatically enter HOLD. When HOLD is entered, the HOLD NEXT key light extinguishes and the HOLD key illuminates.

In the equivalent time digitizing mode (1  $\mu$ s/div to 50 ns/div) repetitive triggers are required to build a complete representation of the waveform. The HOLD NEXT feature allows for this by accepting multiple triggers till enough waveform data points are accumulated to build a complete waveform, at which time HOLD is entered. The TRIG'D light stays illuminated (unless repetition rate is slow) during the accumulation interval and turns off when the process is complete (when HOLD is entered). Table 2-5 gives the number of triggers required to build a waveform for the different TIME/DIV settings in the equivalent time digitizing range.

When HOLD NEXT is activated over the GPIB, the 7D20 issues a service request (SRQ) when HOLD is entered. This provides notice to the controller that a HOLD NEXT acquisition has been completed. Refer to Section 4, GPIB for more information.

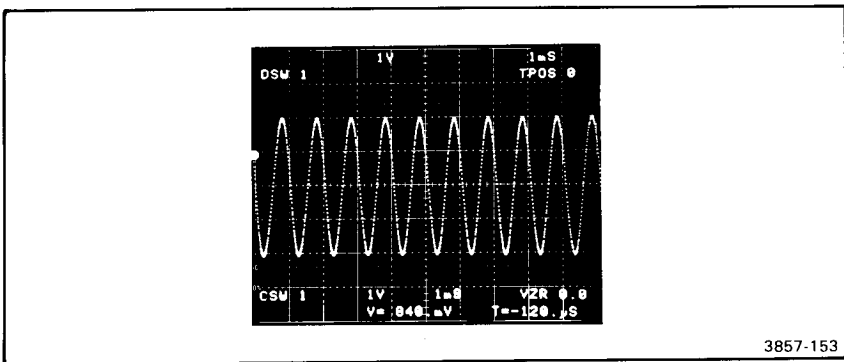
**TABLE 2-5**  
**Equivalent Time Digitizing Mode Trigger Requirements**

TIME/DIV	Number of points per trigger	MINIMUM triggers to build a waveform	Average number of triggers to build a waveform
1 $\mu$ s	204/205	5	11
500 ns	102/103	10	29
200 ns	40/41	25	95
100 ns	20/21	50	224
50 ns	10/11	100	516

When using the HOLD NEXT function it is very important to check that the trigger LEVEL is adjusted so the trigger will be accepted. Otherwise, you may miss the event you want to capture.

### +SLOPE

This key illuminates when selected and causes the trigger circuit to respond to the positive-going portion of the trigger signal. Conversely, when the key is not illuminated the trigger circuits respond to the negative-going portion of the trigger signal (see Fig. 2-51). When several cycles of a signal appear in the display, the setting of this switch is often unimportant. However, if only a certain portion of a cycle is to be acquired and displayed, correct SLOPE selection is necessary to ensure that acquisition begins on the desired slope of the input signal.



**Figure 2-51. Triggering on negative slope.**

## TRIG'D

This indicator light illuminates when signal acquisition is triggered. In the ROLL digitizing mode triggers are not required so the TRIG'D light doesn't illuminate except when the HOLD NEXT function is active (refer to the discussion of the HOLD NEXT mode). In the absence of a trigger event the digitizer ceases to update and the last triggered acquisition is displayed. If the trigger light is on but the display is unstable, this usually indicates an aliasing condition. Turn the TIME/DIV control clockwise until a stable display is obtained. Refer to the Operational Theory, Section 3 for a discussion of aliasing.

## LEVEL

The LEVEL control adjusts the voltage level on the trigger signal at which triggering occurs. When the LEVEL control is set in the clockwise direction, the trigger circuit responds at a more positive point on the trigger signal. When the LEVEL control is set in the counterclockwise direction, the trigger circuit responds at a more negative point on the trigger signal.

The LEVEL control range is approximately  $\pm 6.4$  divisions in the AUTO and NORM TRIGGERING MODEs. Refer to TRIGGERING MODE in this section for level range information in the P-P MODE.

To set the LEVEL control, first select the triggering MODE, COUPLING, SOURCE, and +SLOPE. Turn the LEVEL control fully counterclockwise then, rotate it clockwise until the display of the acquired signal begins at the desired point.

## ↔ TRIG POS ↔

The TRIG POS (trigger position) slewing keys allow you to increment  $\rightarrow$  or decrement  $\leftarrow$  the trigger position by whole divisions from its zero position on the left vertical graticule line. The trigger position appears in the upper right corner of the display as shown in Figure 2-52. A full 10-divisions of pre-trigger can be displayed by moving the trigger position fully to the right of its zero position. In the opposite direction, the trigger position can be moved to the left of zero to achieve up to 1500 divisions of post trigger. When a TRIG POS key is pressed and held, the trigger position automatically stops incrementing or decrementing when its limit is reached. Likewise, when moving it back toward zero by pressing and holding the appropriate key, the trigger position automatically stops at zero. Trigger position 0 on the graticule includes several points of pretrigger data prior to the graticule line. In the extended real-time mode 10 pretrigger points are shown, and in the digitizing modes, 12 points of pretrigger are displayed.

## HORIZ POSITION

This control adjusts the horizontal position of all the waveforms. The HORIZontal POSITION control has a calibrated detent at the clockwise end of rotation. This ensures that the TRIGger POSitions are lined up with the graticule lines as indicated in the readout. The range of the HORIZ POSITION control is plus or minus about 2 divisions.

## VECTOR

Acquired waveforms are stored in waveform memory as data points. These data points are displayed as a series of dots that represent the digitized waveform data points. As these dots become separated vertically it sometimes becomes difficult to determine which dot follows which. The VECTOR function, when selected connects time adjacent dots and the key illuminates. This allows you to better determine their time relationships. The VECTORS are designed such that you can still easily discern the waveform data points because they appear intensified with the VECTORS on. VECTORS are particularly useful when viewing waveforms in the ENVelope mode.

## MEMORY DISPLAY

MEMORY DISPLAY keys 1 through 6 control the display of the 7D20 waveform memories. Six memory registers in the 7D20 are dedicated to the storage of waveform data. Each waveform memory register is functionally divided into three segments. One segment stores the waveform data points. A second segment of the waveform memory stores the original conditions under which the waveform data was acquired, such as the vertical sensitivity and the time/division. A third segment of the waveform memory stores information concerning the "display conditions" for

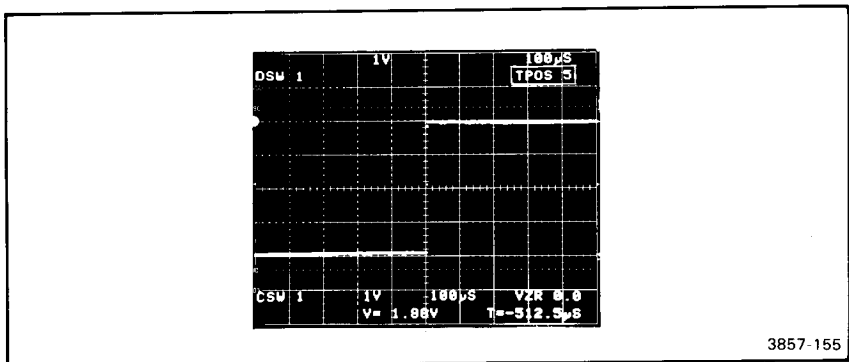
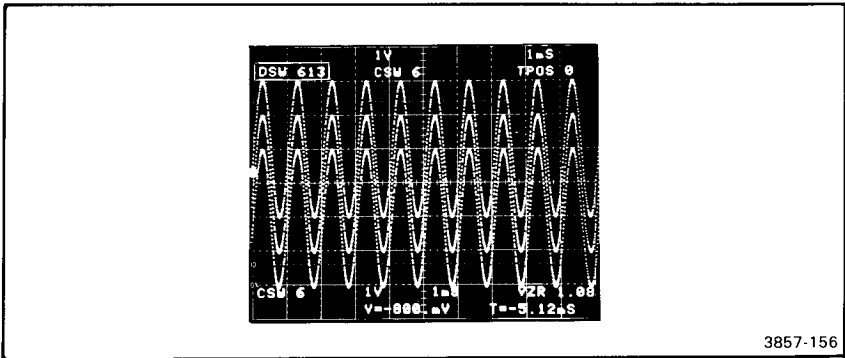


Figure 2-52. Trigger POSition readout.

3857-155



the waveform such as the vertical position (up or down), vertical expansion or compression, versus, cursor position and horizontal magnification. The contents of any waveform memory can be displayed by pressing its corresponding (1 through 6) MEMORY DISPLAY key. When pressed, the numbered MEMORY DISPLAY keys illuminate to indicate that the memory is being displayed and the respective waveform memory number appears in the upper left corner of the display in the DSW (Display Waveform) list as shown in Figure 2-53. Pressing the key a second time will extinguish the light and clear the waveform from the display. However, this is not true of the cursor waveform which cannot be turned off. The numbered MEMORY DISPLAY keys also have a second function. They are used for making selections from the MENU, discussed earlier in this section.



3857-156

Figure 2-53. DSW (displayed waveforms) readout.

## COPY

This key permits you to copy a waveform from one memory to another. When a waveform is copied from one waveform memory to another, all the associated acquire conditions and display conditions are copied too. To do this, press the COPY key: the COPY key will start blinking and a prompt will appear in the display. The prompt (shown in Fig. 2-54) will read "COPY #->#" with the first number sign blinking. Next, press the numbered MEMORY DISPLAY key corresponding to the memory you wish to copy. The MEMORY DISPLAY key will illuminate, the contents of the associated memory will be displayed, and the prompt will show the number of the waveform you are copying in place of the blinking number sign. The second number sign will now blink. Next, press the numbered (1-6) MEMORY DISPLAY key where you wish to copy the waveform. At this point, the waveform will have been copied into the chosen waveform memory and the COPY key will cease blinking and stay off. The prompt will now show the number of the waveform memory into which you just copied the selected waveform in place of the blinking, second number sign. Press any other key and the copy prompt will disappear from the display. A copied waveform will not be displayed unless the memory it was copied into was being displayed before the copy was made.

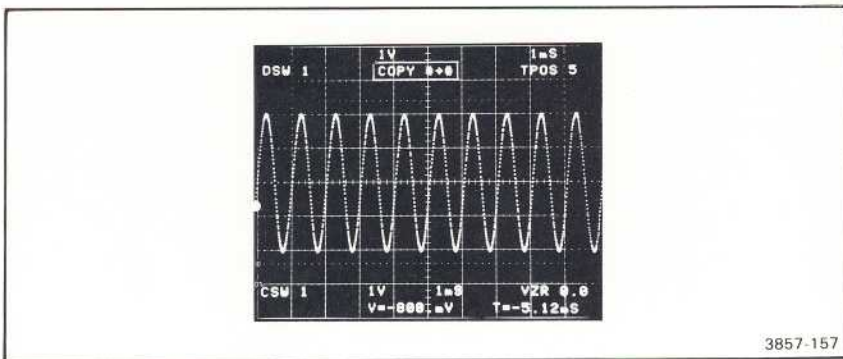


Figure 2-54. 7D20 COPY prompt.

It should also be noted that when a waveform is copied into a waveform memory, any information previously stored in that memory will be cleared and consequently lost.

## CSW

The CSW (CurSor Waveform) select key allows you to select any one of the six waveform memories as the cursor waveform. When pressed, the CSW key will begin to blink and the previous cursor waveform number in the CSW prompt will be replaced with a blinking # sign. You can then select the desired cursor waveform by pressing the appropriate MEMORY DISPLAY key. This instantly results in the display of the new cursor waveform and its associated display information. The CSW prompt will reflect the newly selected CSW number and will disappear from the display when another key is pressed.

## *f* CSW REF

When viewing a waveform in the HMAG or VS modes, this function allows you to also view the waveform as it was previous to the HMAG or VS mode as a reference. The REF function can only be used when the 7D20 is in the HMAG or VS modes and cannot otherwise be selected. Refer to the discussion of HMAG and VS later in this section.

As a shift function of the CSW key, the REF key is activated after first pressing the *f* key. Once selected, the REF key will illuminate and the reference waveform will be displayed and an R will appear in the DSW readout as shown in Figure 2-55. Cursors on the displayed reference waveform will be positioned on the exact same data points as on the displayed HMAG or VS waveform. And, any movement of the cursors will occur on both waveforms.

Termination of the HMAG or VS mode will extinguish the REF key illumination, however, the REF state is retained in memory and will be enabled if the HMAG or VS mode is entered again. The REF state can be terminated by pressing the *f* key then pressing the REF key. This also extinguishes the REF key illumination.

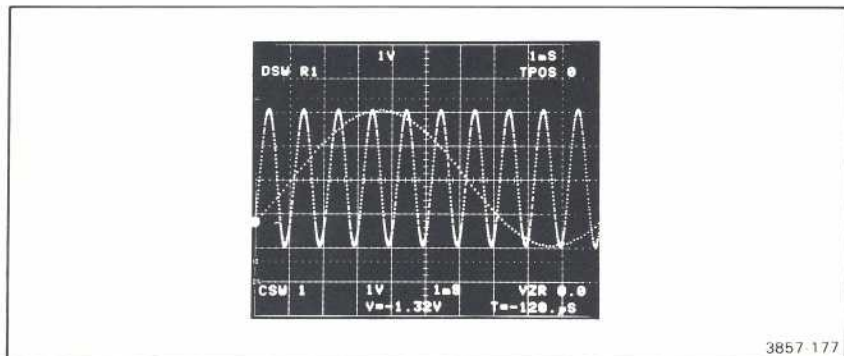


Figure 2-55. DSW readout with R (Reference) Waveform displayed.

## CURSOR WFM

The CURSOR WFM keys allow you to manipulate the cursor waveform in a variety of ways. Keep in mind that each waveform memory has associated with it three separate memory segments; one for waveform data, a second for acquire conditions, and a third for display conditions. Use of these keys affect only the display condition segment of the waveform memory and will not in any way affect the waveform data nor the original acquire conditions which remain stored, unchanged in waveform memory. Using the CURSOR WFM keys, you can vertically reposition, expand, or compress the cursor waveform. Using these keys you can also magnify the cursor waveform horizontally or display it "versus" another waveform or itself. These keys are operational on the cursor waveform only. Waveforms 3 through 6 are always in a hold condition. These keys may also set the cursor waveform display to be used as the **RE**ference waveform independently of the HMAG or VS cursor waveform display when the cursor waveform is in HOLD.

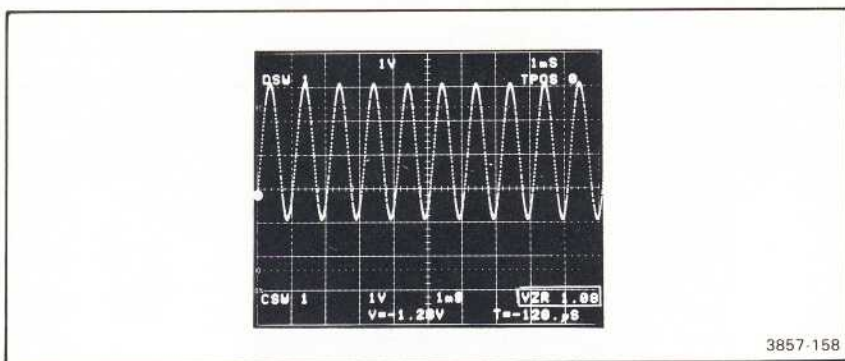
The use of these keys is described in the following paragraphs.

### VPUP $\uparrow$ AND VPDN $\downarrow$ (Vertical Position Up and Vertical Position Down)

These slewing keys, when in HOLD, allow you to move the cursor waveform vertically about the acquired vertical zero reference. A single press of a key moves the waveform one vertical display resolution increment (0.04 division). If you press and hold the key, continuous movement of the cursor waveform will result. Movement stops when the maximum of about 5 divisions (in either direction) from the acquired zero level is reached.

Movement of the cursor waveform display, either up or down with respect to its original display position will result in the illumination of the appropriate (up or down) key. Press and hold the non-illuminated, opposing key to return the cursor

waveform to its original display position. Waveform movement will automatically stop at its original display position, at which time neither key will be illuminated. To continue waveform movement past its original position, release the movement key momentarily, then press and hold it again. The position of the cursor waveform GND (plus or minus) with respect to center screen is shown in the VZR display (see Fig. 2-56). Maximum movement in either direction is approximately 5 divisions (further if expanded). When the REFERENCE waveform is present (which means the cursor waveform must be in HMAG or VS mode) these keys allow separation control between the cursor waveform and its reference. These keys operate in this mode even if the 7D20 is not in HOLD.



3857-158

Figure 2-56. Cursor waveform with VZR (Vertical Zero Reference) altered by VPUP.

### VCMP $\updownarrow$ AND VXP $\updownarrow$ (Vertical Compress and Expand)

With these keys you can vertically compress or expand the cursor waveform about the center of the acquisition window. The cursor waveform can be expanded or compressed, two increments (in a 1,2,5 sequence) from its original acquired vertical scale factor.

When the cursor waveform is expanded or compressed, the respective VXP $\updownarrow$  or VCMP $\updownarrow$  key illuminates and the change in vertical scale factor will appear in the display in the second from the bottom line labeled CSW. Because these keys have opposing action, the opposite key is used to return the cursor waveform to its original acquired vertical scale factor. At which time, neither key will be illuminated.

Waveforms being digitized cannot be vertically compressed or expanded and are not affected by these keys except when HMAG or VS modes are selected and the REF waveform is displayed. An attempt to do this results in the display of the warning message "HOLD REQD". Refer to prompts and warnings discussion given earlier in this section. VXP $\updownarrow$  occurs about crt center whereas VCMP $\updownarrow$  occurs about VZR (Vertical Zero Reference). This feature allows the user to position the expanded waveform anywhere within the graticule area.



## HMAG (Horizontal Magnify)

The cursor waveform can be magnified (expanded) horizontally from its acquired state by a factor of 10 times. In other words a 100 points/division waveform is displayed at 10 points/division as shown in Figure 2-57. By using the cursors as described later under cursors and magnified waveforms, any portion of the magnified cursor waveform can be positioned within the viewing area of the display. It is often helpful to view the **REF** (unmagnified) waveform (also shown in Fig. 2-57) at the same time. Since the cursors appear on exactly the same data points on both versions of the waveform.

The HMAG key illuminates when the function is active. Press the key to select. Press again to terminate—the light will extinguish. The HMAG function will also terminate if the VS mode is selected.

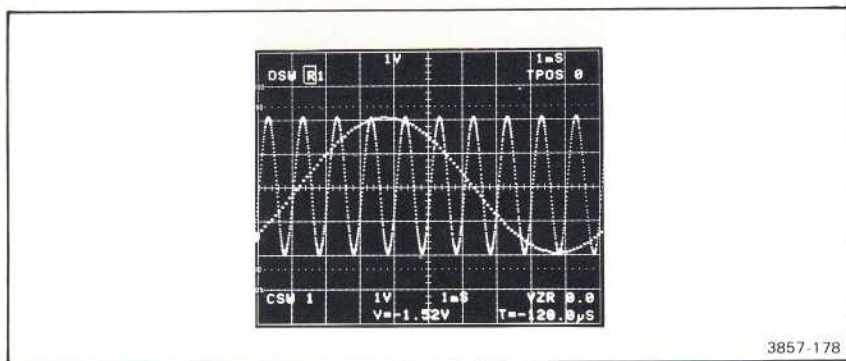


Figure 2-57. HMAG Waveform displayed with its reference waveform.

## **f**, HMAG ALL

This function causes all displayed waveforms to be horizontally magnified 10 times. This is a shift function of the HMAG key. As such, the **f** key must be pressed and illuminated and the HMAG key blinking, before this function can be selected. When the **ALL** function is active and the HMAG key is pressed without first pressing the **f** key, only the cursor waveform will be unmagnified. Since the cursor waveform is not magnified the HMAG key light will extinguish. Press the key again (without first pressing the **f** key) and the cursor waveform will again be displayed magnified and the HMAG key will again illuminate. To cancel the **ALL** function, press the **f** key first, then press the **ALL** key. The cursor waveform will also cease to be magnified if the VS mode is selected.

## VS

This key allows you to display the cursor waveform versus another waveform (or versus itself) rather than versus time. When the function is active, the 7D20 displays the cursor waveform on the Y-axis (vertical) versus the selected waveform on the X-axis (horizontal).

To select the VS function, first press the VS key. The VS key will illuminate and the prompt "CSW X VS #" will appear in the display as shown in Figure 2-58. The X in the prompt will be the number of the cursor waveform and the # will be blinking. Next, press the number of the selected waveform. The cursor waveform will be displayed vs the selected waveform, and the CSW information at the bottom of the display will indicate that the cursor waveform is being displayed VS the selected waveform (see Fig. 2-59). The cursor waveform information at the bottom of the display will change to indicate the volts/div of the two waveforms. Refer to the discussion of the display earlier in this section.

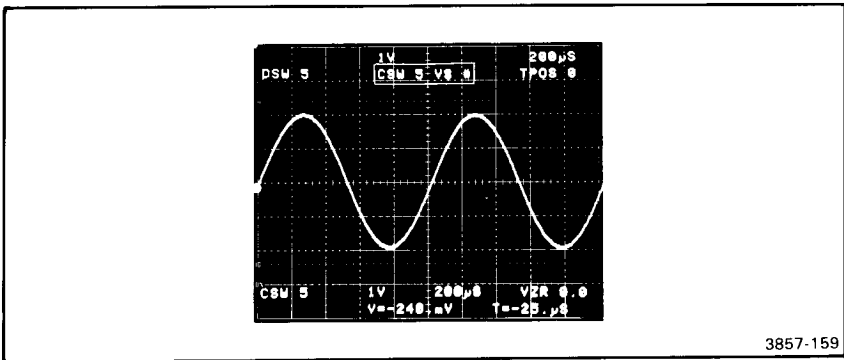


Figure 2-58. Cursor Waveform VS (Versus) prompt.

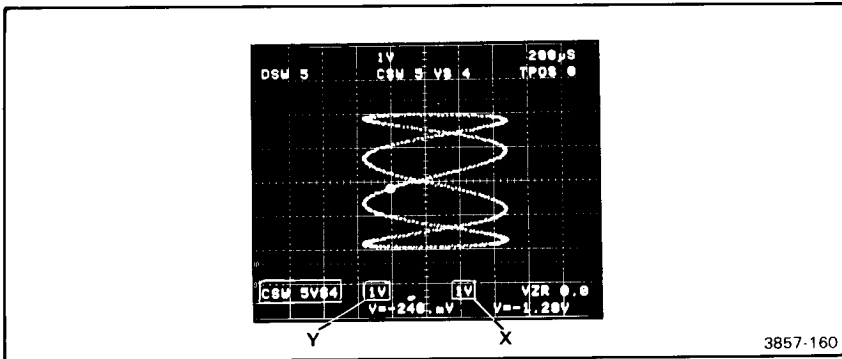


Figure 2-59. CSW versus readout.

## CURSORS

Each waveform memory has associated with it two cursors where one or both may be displayed when the waveform is selected as the cursor waveform. The cursors appear in the display as intensified dots that coincide with waveform data points. When in HMAG or VS modes and with the REFerence waveform displayed, cursors appear on the same data points on both versions of the waveform.

At power-up, when the ALIGN mode is selected, cursors for all waveforms are aligned with those of the cursor waveform. And, consequently, any movement of the cursors on the cursor waveform will cause identical movement of the cursors on all other waveforms. Of course, only the cursors on the CSW are visible. Independent movement of the cursors is made possible by selecting the INDEPENDent mode as explained later.

Cursors are moved by use of the slewing keys  $\leftarrow$ 1, 1 $\rightarrow$ ,  $\leftarrow$ 2, and 2 $\rightarrow$ . The arrows indicate the direction of cursor movement. A single press of a cursor movement key will cause the associated cursor to move to the next adjacent data point on the waveform. Continuous movement of the cursor will result if the key is pressed and held. The following rules apply to the cursors and their use:

- Cursor 1 is always on.
- Cursor 1 can be moved to any cursor waveform data point.
- Cursor 1 is always coincident with or closer to the beginning of the waveform than cursor 2.
- Cursor 1 pushes cursor 2 when the two are coincident and cursor 1 is moved toward the end of the waveform.
- Cursor 2 movement is halted when it becomes coincident with cursor 1 as you attempt to move cursor 2 toward the beginning of the waveform.

### NOTE

*If cursor 2 is sufficiently separated from cursor 1, it may not appear on the magnified portion of a waveform in the HMAG mode. This can be more easily observed if the REFerence waveform is turned on.*

## CURSOR COORDINATES

The kind of cursor coordinate information displayed on the crt depends on whether cursor 1 is on by itself or if cursor 2 is on also. When cursor 2 is off, the displayed cursor coordinates give both the vertical position and the horizontal time value of the waveform data point where cursor 1 is located as seen in Figure 2-60. The vertical position is given in volts with respect to the VZR (vertical zero reference) value. The horizontal time value is given with respect to the trigger position. This is also true whenever the REF waveform is displayed. In the VS mode (without the REF waveform), the display gives only the vertical coordinates of the cursors on

waveforms as shown in Figure 2-61. Again, these coordinates are given in volts relative to the respective vertical zero reference values of the two waveforms:

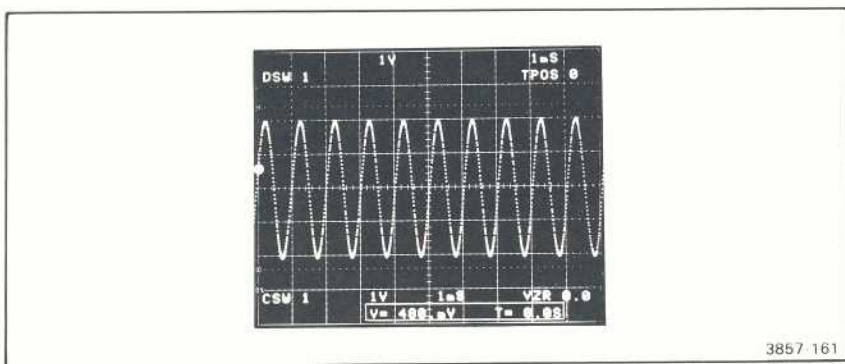


Figure 2-60. Cursor coordinate readout.

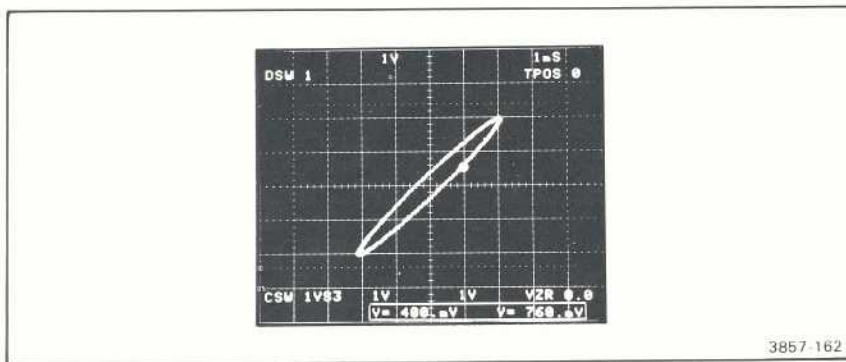


Figure 2-61. Cursor coordinate readout with VS Waveform displayed.

In the  $\Delta$ ON cursor mode, cursor 2 is turned on. The display gives the vertical value of cursor 2 relative to cursor 1 and the horizontal time value of cursor 2 relative to cursor 1 as shown in Figure 2-62. When the VS mode is selected, the display gives the vertical value of cursor 2 relative to cursor 1 for both X and Y waveforms.

## CURSORS AND MAGNIFIED WAVEFORMS

When viewing an unmagnified cursor waveform, the entire length of the waveform is visible within the display area and cursor 1 can be moved to any data point on the waveform. However, when HMAG is selected only part of the waveform remains



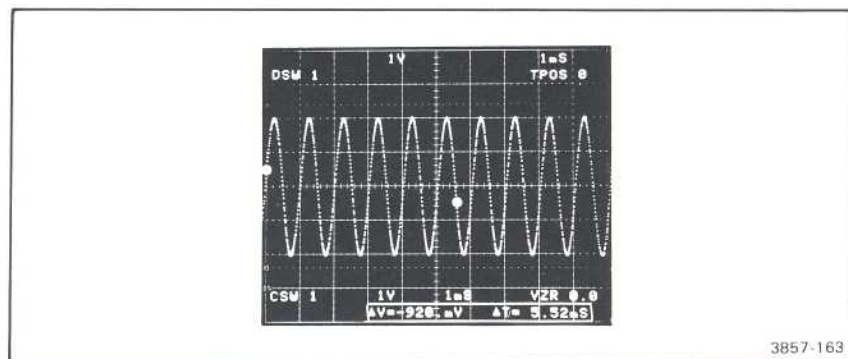


Figure 2-62. Cursor coordinate readout when both cursors are on ( $\Delta$ ON).

visible. So as to still be able to examine the entire magnified waveform, the following occurs: As cursor 1 is moved toward the end of the waveform, it becomes fixed at graticule line 2 (see Fig. 2-63). Then the waveform moves relative to cursor 1 until the last data point comes into the display area. Cursor 1 then resumes movement till it reaches the end of the waveform. Similarly, the reverse is true when cursor 1 is moved back toward the beginning (left) of the waveform. This can be more easily observed if the REF (reference) waveform is displayed at the same time since the cursors appear on both the magnified cursor waveform and the REF waveform.

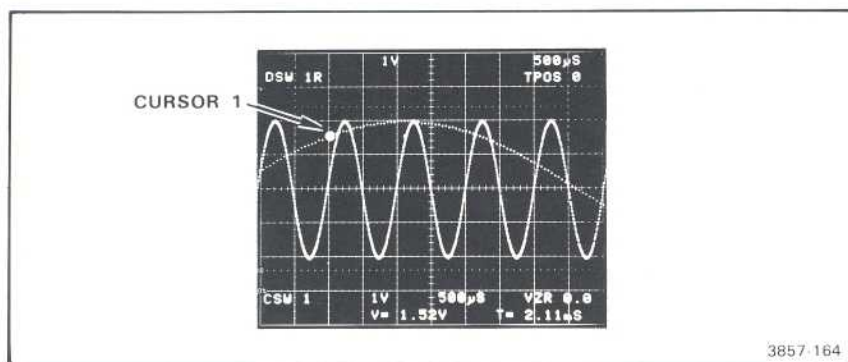


Figure 2-63. Cursor 1 location on HMAG'd Waveform.



These two slewing keys move cursor 1 in the direction indicated by the arrows. Pressing either of the keys once will cause cursor 1 to move to the next data point in the direction indicated. Continuous movement of cursor 1 results when a cursor movement key is pressed and held. When the last data point is reached, in either direction, cursor 1 will come to a halt.

## ⇄2⇄

These two slewing keys move cursor 2 in the direction indicated by the arrows. Pressing either of the keys once will cause cursor 2 to move to the next data point in the direction indicated. An exception is when cursor 2 is co-incident with cursor 1 or when cursor 2 is at the end of the waveform (last data point). Continuous movement of cursor 2 will result when one of the movement keys is pressed and held. Cursor 2 cannot be positioned to the left (toward the beginning of a waveform) of cursor 1 nor can it be positioned beyond the last waveform data point.

## ΔON ΔOFF

These keys turn cursor 2 **ΔOFF** and **ΔON**. As shift functions of the cursor 1 movement keys, you must first press the **f** key to activate these functions. When cursor 2 is turned on, the displayed cursor coordinates will reflect the vertical value of cursor 2 relative to cursor 1 and the horizontal time value of cursor 2 relative to cursor 1. Each of these functions is cancelled upon the selection of the other.

## INDEPENDent

This is a shift function of the ⇄2 key. Selection of the **INDEPENDent** mode permits you to move the cursors on the cursor waveform without affecting the position of the cursors on any other waveforms. Selection of this mode requires that the **f** key be pressed first, the key illuminates when selected. The **INDEPENDent** mode is cancelled by selecting the **ALIGN** mode.

## ALIGN

This is a shift function of the cursor 2⇄ key and its selection must be preceded with a press of the **f** key. Selection of the **ALIGN** mode illuminates the **ALIGN** key and causes the cursors on all waveforms to align with the cursors on the cursor waveform. Thereafter, movement of the cursors on the cursor waveform will cause identical movement of the cursors on all other waveforms. The **ALIGN** mode can be cancelled by selecting the **INDEPENDent** mode described above.

## SET N

SET N is used with the AVE **N** and ENV **N** modes to AVERage or ENVELOpe an incoming waveform N number of times. With this key, you can set the value of N to: 8, 16, 32, 64, 128, or 256. The SET N key illuminates when pressed and "SET N=(current value of N)" appears in the display as shown in Figure 2-64. Press the SET N key again and the value of N will increment to the next higher value. If the SET N key is pressed and held, the value of N will cycle through the above sequence of numbers in the display. Simply release the key when the desired value of N appears. SET N is terminated by selecting any other key and the value of N is retained.

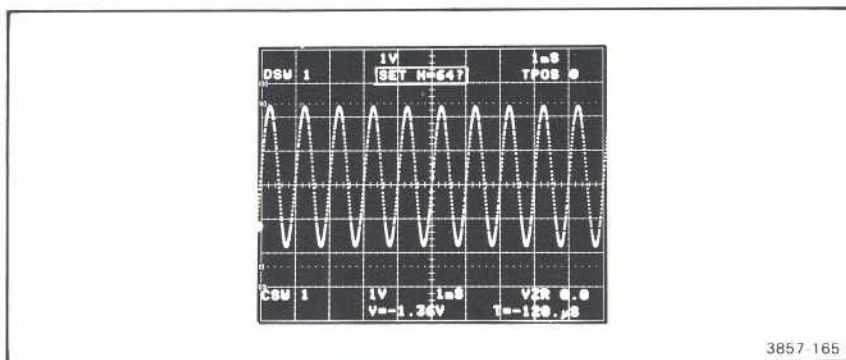


Figure 2-64. SET N readout prompt.

## AVE

AVEraging is a method of removing noise from a signal and presenting a better representation of the actual waveform.

When the AVE key is pressed it will illuminate and the 7D20 will begin to continuously display averaged waveform data. The value of N will adjust the responsiveness of the accumulated average to any changes in waveforms acquired after the N<sup>th</sup> waveform. Each additional waveform is weighted by 1/N. After N acquisitions the prompt field will indicate that the number of averages is greater than N (see Fig. 2-65).

When AVEraging in the ROLL mode (TIME/DIV set to 100 ms or slower) the waveform is sampled at a constant 2 kHz rate (1 kHz at 20 s/div). This allows the

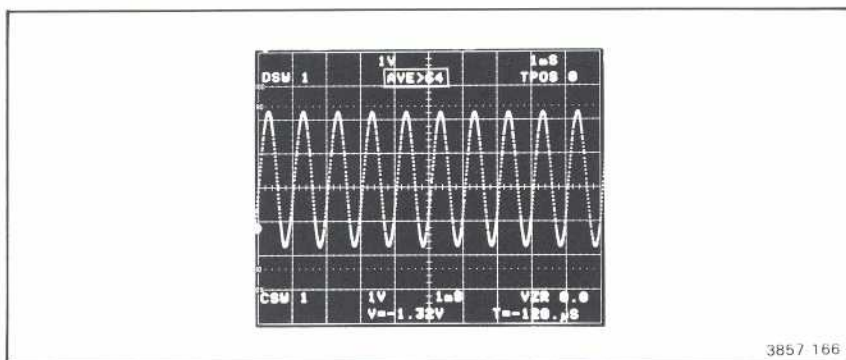


Figure 2-65. The AVE readout.

7D20 to capture and process a number of samples between each pair of displayed points. The next point displayed is the average of these "in between" samples. See Table 2-6.

**TABLE 2-6**  
**Number of Points Averaged for**  
**Each Displayed Point in ROLL Mode**

TIME/DIV	Digitized Points/Displayed Point
100 ms	2
200 ms	4
500 ms	10
1 s	20
2 s	40
5 s	100
10 s	200
20 s	200
EXT	100

In the ROLL Mode AVEraging is performed in real time, therefore repetitive waveforms are not required. AVEraging in the ROLL mode using the EXTERNAL CLOCK input allows 100 samples to be taken between displayed points.

The AVE function can be cancelled by pressing the AVE key a second time. Or, selecting HOLD, HOLD NEXT, AVE N, ENV, or ENV N will also terminate AVE with the following results:

**HOLD**—Immediately stops signal acquisition, terminates the AVE function and extinguishes the AVE key. The display readout will indicate the total number of averages accumulated, or greater than N.

**HOLD NEXT (TRIGGERING MODE)**—Allows one more triggered acquisition then enters HOLD with the same effects as the HOLD key.

**GND**—Terminates the AVE function and redefines VZR.

**AVE N**—Clears the display then executes the AVE N function.

**ENV and ENV N**—Both terminate the AVE function, extinguish the AVE key and implement the selected function.

If the VOLTS/DIV, TIME/DIV, or TRIG POS are changed, the AVEraging restarts and discards previous information.



**Algorithm:** For  $50 \text{ ms/div} \geq \text{TIME/DIV} \geq 50 \text{ ns/div}$  (Not ROLL mode).

$$W_n^a(m) = 0 \text{ For } m = 0, 1, \dots, 1023$$

For  $n = 1, 2, \dots, N$

$$W_n^a(m) = W_{n-1}^a(m) + [W_n(m) - W_{n-1}^a(m)] / q.$$

where  $q = 2^{**} \text{INT} [\text{Log}_2 (1.443 * n)]$

$W_n^a$  is the averaged stored waveform after  $n$  acquisitions

$W_n$  is the  $n^{\text{th}}$  acquired waveform

For  $n > N$

$$W_n^a(m) = W_{n-1}^a(m) + W_n(m) - W_{n-1}^a(m) / N$$

## f, AVE N

This shift function of the AVE key is selected after first pressing the **f** key. AVE **N** is the same as the AVE function except when **N** (current setting of SET **N**) averages have been made, the 7D20 automatically enters HOLD. Also, when AVE **N** is activated, the readout prompt begins at the current value of **N** and counts down to zero as averages accumulate. When HOLD is entered the prompt will then give the total number of averages. Should you enter HOLD before **N** averages occur, the readout prompt will give the total reached. AVE **N** accumulates complete waveform records, therefore Roll operation is suspended for TIME/DIV settings  $\geq 100 \text{ ms}$ . When the AVE **N** function is initiated over the GPIB, the 7D20 issues a service request (SRQ) over the bus when HOLD is entered. Refer to Section 4, GPIB.

## ENV

ENVeloping is a method of constructing a waveform using the minimum and maximum values of the sampled data points. ENVeloping makes many waveform aberrations visible that might otherwise be overlooked altogether, for example, frequency drift.

Press the ENV key and the 7D20 will begin to display envelope processed data. The source of the data is determined by the AQR MODE. An envelope waveform is constructed from the appropriate source and displayed following each triggered acquisition. The display is updated after each acquisition in a continuous process. The number of processed acquisitions is shown in the display (see Fig. 2-66). Selecting the ENV function will cause the ENV key to illuminate and, at the same time terminates the AVE, AVE **N**, SET **N**, ENV **N**, and HOLD functions if either are active.

When ENVeloping in the ROLL mode (TIME/DIV set to 100 ms or slower) the waveform is sampled at a constant 2 kHz rate (1 kHz at 20 s/div). This allows the capture and processing of a number of samples between each displayed point (minimum and maximum). The next two points displayed are the maximum and minimum of these "in between" samples.

TABLE 2-7  
Number of Points Averaged for Each Max and  
Min Pair of Displayed Points in ROLL Mode

TIME/DIV	Digitized Points/Max-Min Pair
100 ms	4
200 ms	8
500 ms	20
1 s	40
2 s	80
5 s	200
10 s	400
20 s	800
EXT	200

This mode can be used to capture and process a signal pulse width of 500  $\mu$ s or less. Refer to Section 5, Application 3, Monitoring Intercellular Neuronal Discharge.

ENVeloping in the ROLL mode using the EXTernal CLOCK input allows 100 samples to be taken between displayed points.

To cancel the ENV function press the key a second time. Or, the ENV function can be terminated by pressing the HOLD, HOLD NEXT, GND, or  $f$  ENV  $N$  keys with the following results:

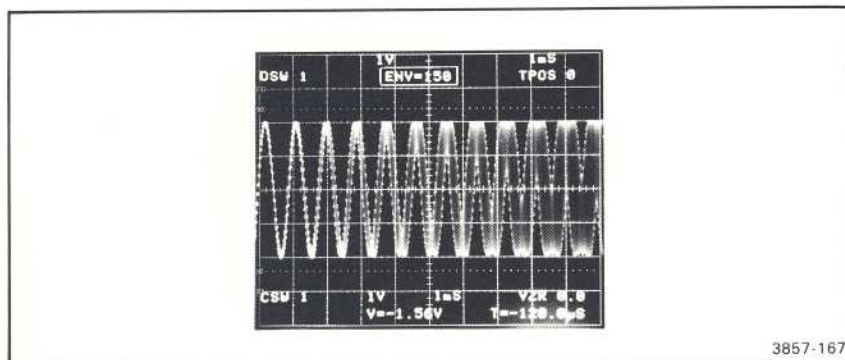


Figure 2-66. The ENV readout.

**HOLD**—Immediately terminates the ENV function and signal acquisition. The number of waveform ENV is displayed in the readout prompt field.

**HOLD NEXT**—Same effects as HOLD except termination is delayed until after the completion of the next acquisition.

**GND**—Terminates the ENV function and redefines VZR.

**f, ENV N**—Terminates the ENV function, clears the displayed waveform and initiates the AVE, AVE N, and ENV function.

If the VOLTS/DIV, TIME/DIV, or TRIG POS are changed, the ENVELOping restarts and previous information is discarded.

**Algorithm:** For  $100 \text{ ms/div} \geq \text{TIME/DIV} \geq 50 \text{ ns/div}$  (Not ROLL mode).  
For  $n = 2$  to infinity.

$$W_n^a = \text{Max} [W_{n-1}^a(m), W_n(m)] \quad m = 0, 2, \dots, 1022$$

$$W_n^a = \text{Min} [W_{n-1}^a(m), W_n(m)] \quad m = 1, 3, 5, \dots, 1023$$

where  $W_n^a(m)$  is the  $m^{\text{th}}$  point of the constructed envelope waveform after  $n$  acquisitions.

$W_n(m)$  is the  $m^{\text{th}}$  point of the  $n^{\text{th}}$  waveform acquisition.

## f, ENV N

This shift function of the ENV key is selected after first pressing the **f** key. ENV N operates the same as the ENV mode except when N (the current setting of SET N) envelopes are accumulated, the 7D20 automatically enters HOLD. When ENV N is activated the readout prompt begins counting at the current value of N and counts down to zero. At the zero count, HOLD is automatically entered and the readout prompt will indicate the total number of envelopes.

Should HOLD be initiated before N envelopes occur, the total number reached will appear in the readout prompt. ENV N accumulates complete waveform records, therefore Roll operation is suspended for TIME/DIV settings  $\geq 100 \text{ ms}$ . When ENV N is initiated over the GPIB, the 7D20 issues a service request (SRQ) when HOLD is entered. GPIB operating information is contained in Section 4, GPIB.

## f, TEST

Selects the Test Menus that are used for executing SELFTEST and servicing the 7D20. Refer to the Maintenance section of the service manual for this information.

## RQS (REQUEST SERVICE)

When the 7D20 is on bus in TALK/LISTEN mode, pressing this key will generate an SRQ (service request) over the GPIB. The key will illuminate when you do this. A poll by the GPIB controller will subsequently extinguish the illuminated key and cancel the SRQ.

A power-up SRQ is generated as part of the power-up sequence when power is applied to the 7D20. If the 7D20 is in the GPIB mode and ON LINE, the key will extinguish when the 7D20 is polled by the controller. Refer to Section 4, GPIB.

### RQS #

Allows selection of 6 unique SRQ event codes when the 7D20 is operating over the GPIB. This sends a predetermined status byte. Refer to Section 4, GPIB for information concerning GPIB operation.

## REMOTE ONLY (indicator)

This indicator light illuminates when the 7D20 is in the RWLS (Remote With Lock out State). When the light is on, the front panel controls can only be activated or changed over the GPIB. All front panel controls are inactive except for the RQS functions, the variable VOLTS/DIV, HORIZ POSITION, and ID monitoring of GPIB selections.

## ID (ADDRed Indicator)

The ID key, when pressed, causes the ID menu to appear in the display as shown in Figure 2-67. The ID menu is used for manually establishing the MODE TERMINATION, and ADDRESS for GPIB operation of the 7D20. The ADDR illuminates when the 7D20 has been addressed as a Talker or Listener. Refer to Section 4, GPIB, for further information.

### NOTE

*The firmware version displayed in these illustrations may differ from those displayed for your 7D20, and are shown for positional reference only.*



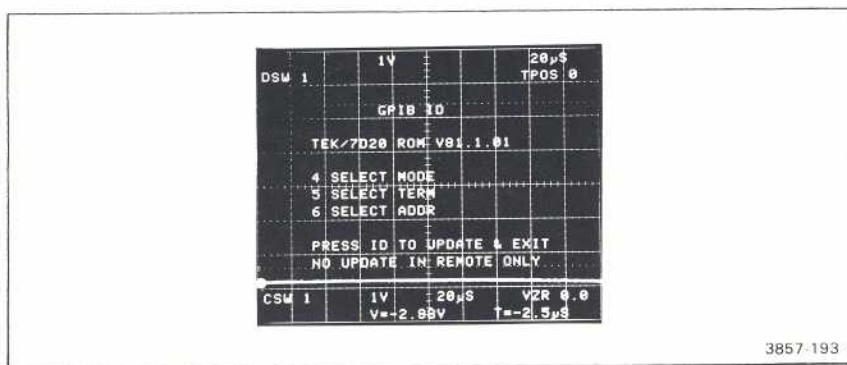


Figure 2-67. ID MENU.

## EXT CLOCK

This bnc connector permits the connection of an external clock signal to the 7D20. The 7D20 can accept a maximum clock rate of 10 kHz. In order for the EXT CLOCK connector to be functional, the TIME/DIV switch must be set to the EXT position by turning it counterclockwise until the readout indicates EXT  $\uparrow$  or EXT  $\downarrow$  in the upper right corner of the display (see Fig. 2-68). The arrow indicates the EXT CLOCK edge polarity. Clock edge polarity is selected via item 4 on the UTILITIES menu. Press MEMORY DISPLAY key 4 when the UTILITIES menu is displayed and the EXT CLK prompt will appear in the display. Press MEMORY DISPLAY key number 4 a second time and the arrow (indicating the EXTERNAL CLOCK polarity) will reverse direction. The indicator arrow will continue to reverse each time the number 4 MEMORY DISPLAY key is pressed so long as the UTILITIES menu is displayed.

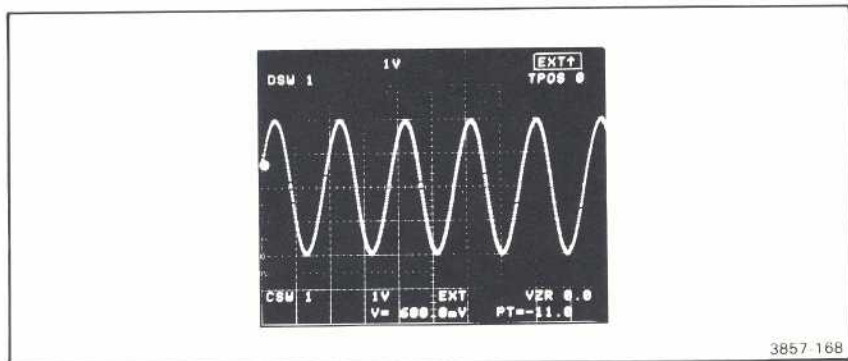


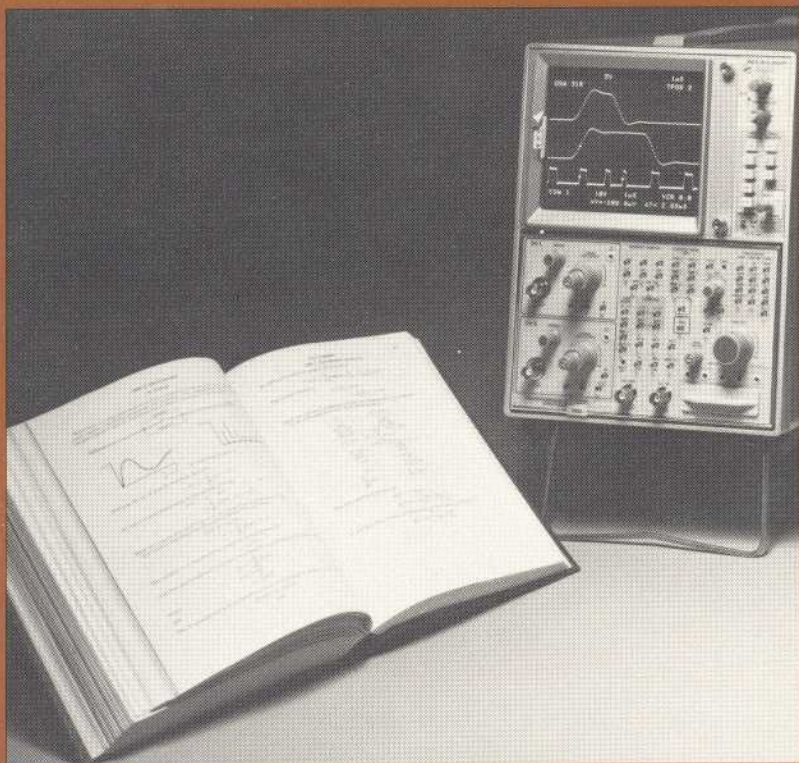
Figure 2-68. EXTERNAL clock readout.

## EXT TRIG

This bnc connector permits the connection of an external trigger signal to the 7D20. In order for this connector to be functional, the EXT or EXT +10 TRIGGERING SOURCE MODE must be selected.

## GPIB (connector)

This is the IEEE 488 connector port for connecting the 7D20 to operate over the GPIB. Information for GPIB operation is contained in Section 4, GPIB.



This section provides an overall view of the 7D20 and its operational capabilities. A general block diagram description of the input signal processing, digitizing, data storage, and the resultant displays will prepare you for the more detailed operational aspects explained later in this section. The information provided in this section is not required to operate the 7D20; however, it may help you understand more fully its internal processes. This understanding will help you utilize the 7D20 for your own specific applications.

# SECTION 3 CONTENTS

---

BLOCK DIAGRAM DESCRIPTION .....	3-1
OVERALL BLOCK DIAGRAM DESCRIPTION .....	3-1
CHARGE COUPLED DEVICE (CCD) OPERATION .....	3-4
DIGITIZING MODES .....	3-4
WAVEFORM DISPLAY .....	3-11

# OPERATIONAL THEORY

---

*THIS INFORMATION IS NOT REQUIRED FOR OPERATING THE 7D20*

## BLOCK DIAGRAM DESCRIPTION

The following description presents an overview of the operation of the 7D20. Figure 3-1 is an overall block diagram of the plug-in. A discussion of the operation of the charge coupled devices (CCDs) the Digitizing modes, and the display system is included at the end of the block diagram description.

### OVERALL BLOCK DIAGRAM DESCRIPTION

The 7D20 Programmable Digitizer performs analog to digital conversion, waveform storage, and digital to analog conversion for display (see Fig. 3-1). The input signal is applied to the Preamplifier circuitry through either or both the CH 1 or CH 2 input connectors. The Preamplifier circuitry attenuates the input signal according to the setting of the front-panel VOLTS/DIV control. It then amplifies the signal, converts it into a differential signal and applies it to the CCD circuitry.

The Charge Coupled Device (CCD) circuitry takes analog samples of the input signal, and passes them to the A/D converter for conversion into digital values. (See the following discussion of the operation of the CCDs.) The digitized samples are applied to the Memory circuitry. The CCD receives its timing and synchronization information from the Time Base circuitry.

The Memory circuitry contains the waveform memory, which consists of digital memory and control circuitry. The Waveform Memory is used to store acquired waveform data from the A-to-D converter, and as a source of data for display on the mainframe crt. In addition, the microprocessor accesses the Waveform Memory to perform operations such as waveform averaging, GPIB data transfer, cursor measurements and storage of data for readout displays.

The Display circuitry receives digital display data from the Waveform Memory and converts it to analog display signals (vertical and horizontal), which drive the mainframe vertical and horizontal display amplifiers, to produce the mainframe crt display. The Display circuitry produces four types of displays: Y versus time, Y versus X, alpha-numeric characters, and cursors. Display setup information from the microprocessor determines the display mode. Timing signals from the Time Base circuitry synchronize the Display with the Memory access time slots.

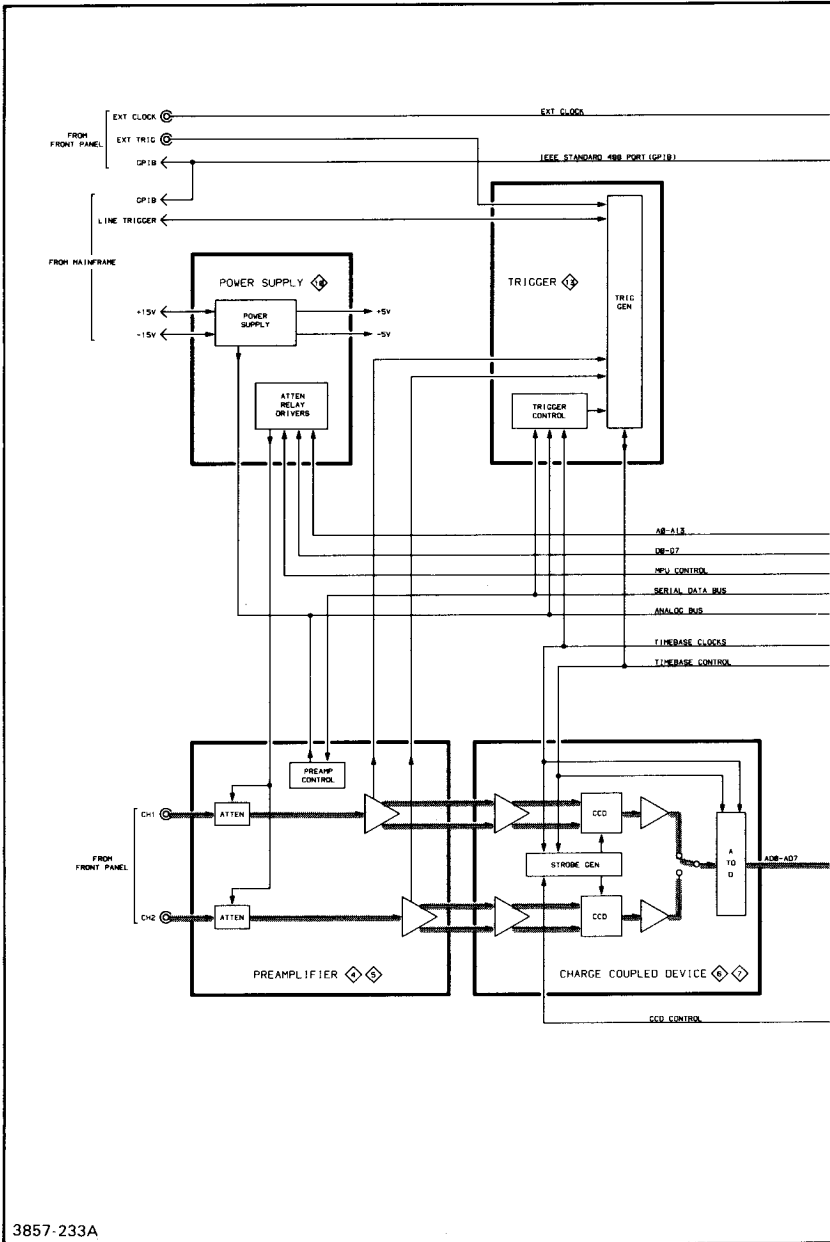


Figure 3-1. 7D20 Overall Block Diagram.

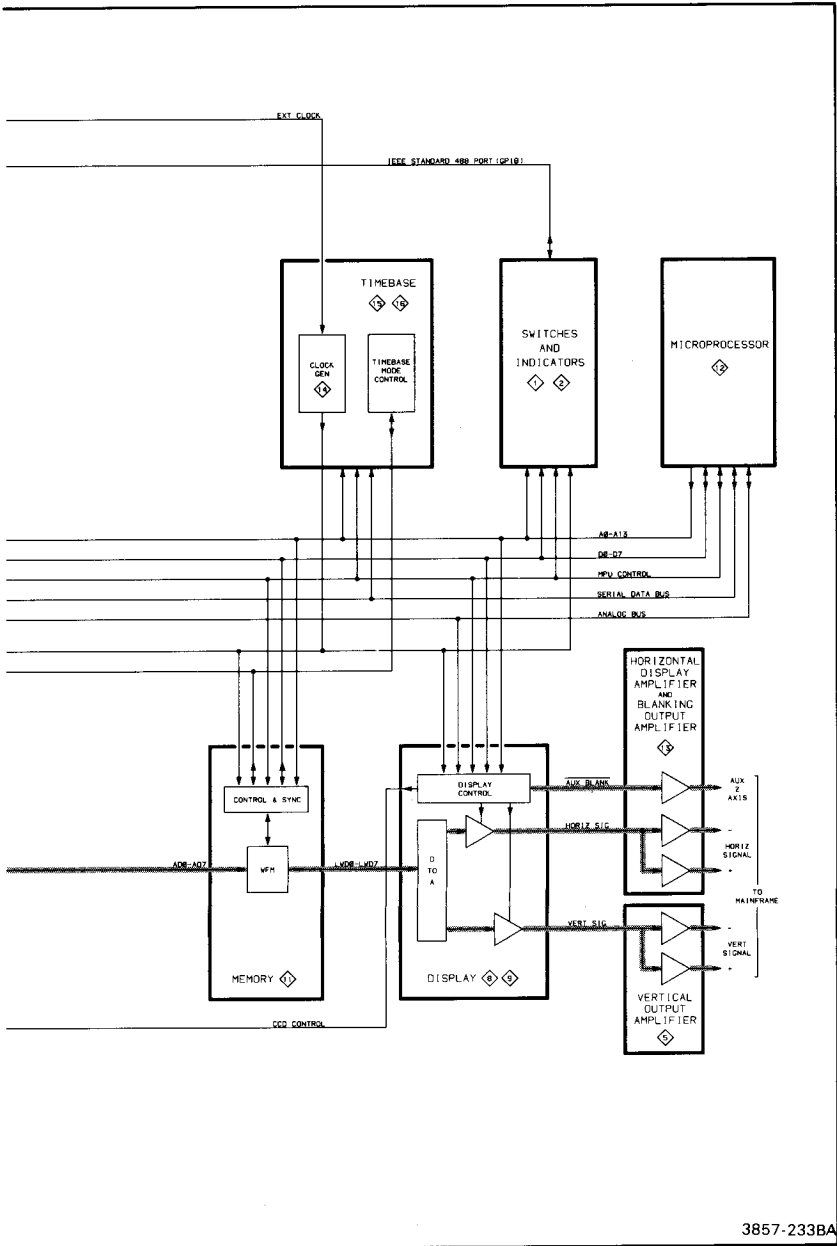


Figure 3-1 (cont). 7D20 Overall Block Diagram.

The Display circuitry also allows waveforms displayed in the Y versus time mode to be expanded, compressed, and offset and magnified horizontally without modifying the contents of the Waveform Memory.

The Trigger circuitry generates the trigger gate signals, which provides a trigger reference point to the Time Base circuitry for the acquired data. It can produce a trigger gate from four different trigger signal sources; Ch 1 input signal, Ch 2 input signal, line and external trigger signal. The trigger source, coupling, mode, level and slope can be selected from the front panel or externally programmed.

The Time Base circuitry provides the basic timing for the 7D20. It generates a number of clock and handshake signals which are used within the Time Base, CCD, Memory and Display circuitry to control the timing of waveform acquisition and display. These signals control the rate at which the CCDs sample the signal, and which of these samples are subsequently stored in the waveform memory. The Time Base circuitry receives setup and control information from the microprocessor, and in turn, transmits trigger and timing data back to the microprocessor for use in setting up and controlling the Memory and Display circuitry.

The Microprocessor circuitry controls the setup of the 7D20 for its various modes of operation, and initiates the acquisition and display of waveforms. It also performs calculations such as signal averaging, enveloping, delta-time and voltage measurements, and controls the self diagnostics and GPIB interface firmware.

## CHARGE COUPLED DEVICE (CCD) OPERATION

A charge coupled device (CCD) is an analog shift register. Each CCD used in the 7D20 contains two analog shift registers (see Fig. 3-2), which are driven differentially. Register A samples the (-) side of the differential input signal; Register B samples the (+) side of the signal.

On each sampling clock to a register, a sample is taken of the signal. This sample is then stored in the first cell of the analog shift register. On subsequent clocks, this sample is passed from cell to cell, until it is applied to the output amplifier and subsequently to an analog-to-digital (A-to-D) converter.

## DIGITIZING MODES

The Time Base circuitry has four basic modes of operation: roll, real-time digitizing (RD), extended real-time digitizing (ERD) and equivalent time digitizing (ETD). The setting of the TIME/DIV control determines the mode of operation of the Time Base.

The roll mode (see Fig. 3-3A) creates a display on the mainframe crt similar to that of a strip chart recorder. In this mode, the CCD circuitry continuously samples the input signal at a 400 kHz rate. Samples are continuously being digitized at the output of the CCD by the A/D converter. Selected samples are then stored in a 1K



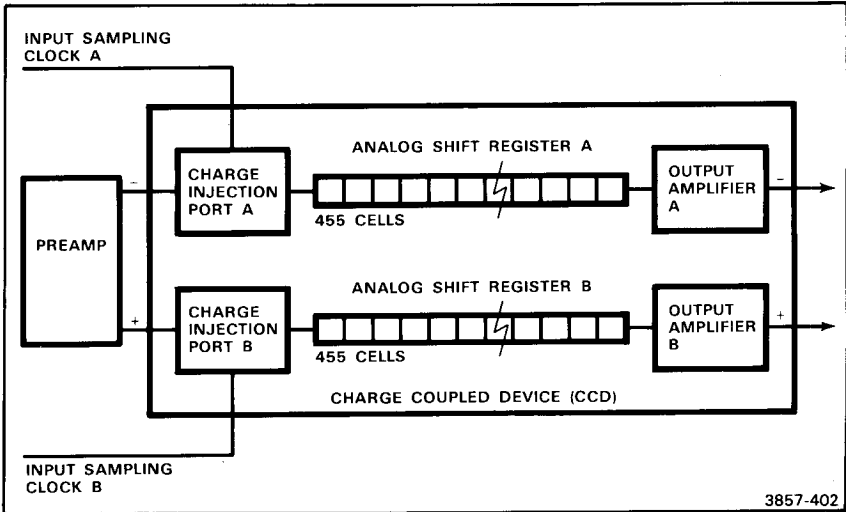


Figure 3-2. Simplified Block Diagram of Charge Coupled Device (CCD).

TABLE 3-1  
Summary of Time Base Modes

Time Base Mode	Time/Div Range	Samples Displayed/Div
Roll Continuous Real-Time Digitizing	20 s to 100 ms	100
Real-Time Digitizing	50 ms to 500 $\mu$ s	100
Extended Real-Time Digitizing	200 $\mu$ s to 2 $\mu$ s	80
Equivalent Time Digitizing	1 $\mu$ s to 50 ns	100

block of the Waveform Memory at a rate determined by the time/division setting. For example, at a time/division setting of 0.1 s/division, samples are stored in the Waveform Memory at a rate of 1 kHz:

$$\frac{100 \text{ samples}}{\text{div}} \div 0.1\text{s/div} = 1000 \text{ samples/sec} = 1 \text{ kHz}$$

Thus, every 400<sup>th</sup> sample that the CCD takes is stored in the Waveform Memory.

When the 1023<sup>rd</sup> memory location is filled, the Memory circuitry wraps around to location 0, such that the Waveform Memory is treated as an infinitely long register. The Display circuitry scans the Waveform Memory continuously, changing the start point in the Waveform Memory for each new sample acquired, such that the crt display is continuously updated with the 1024 most recently acquired samples, which in turn produces the strip-chart effect.

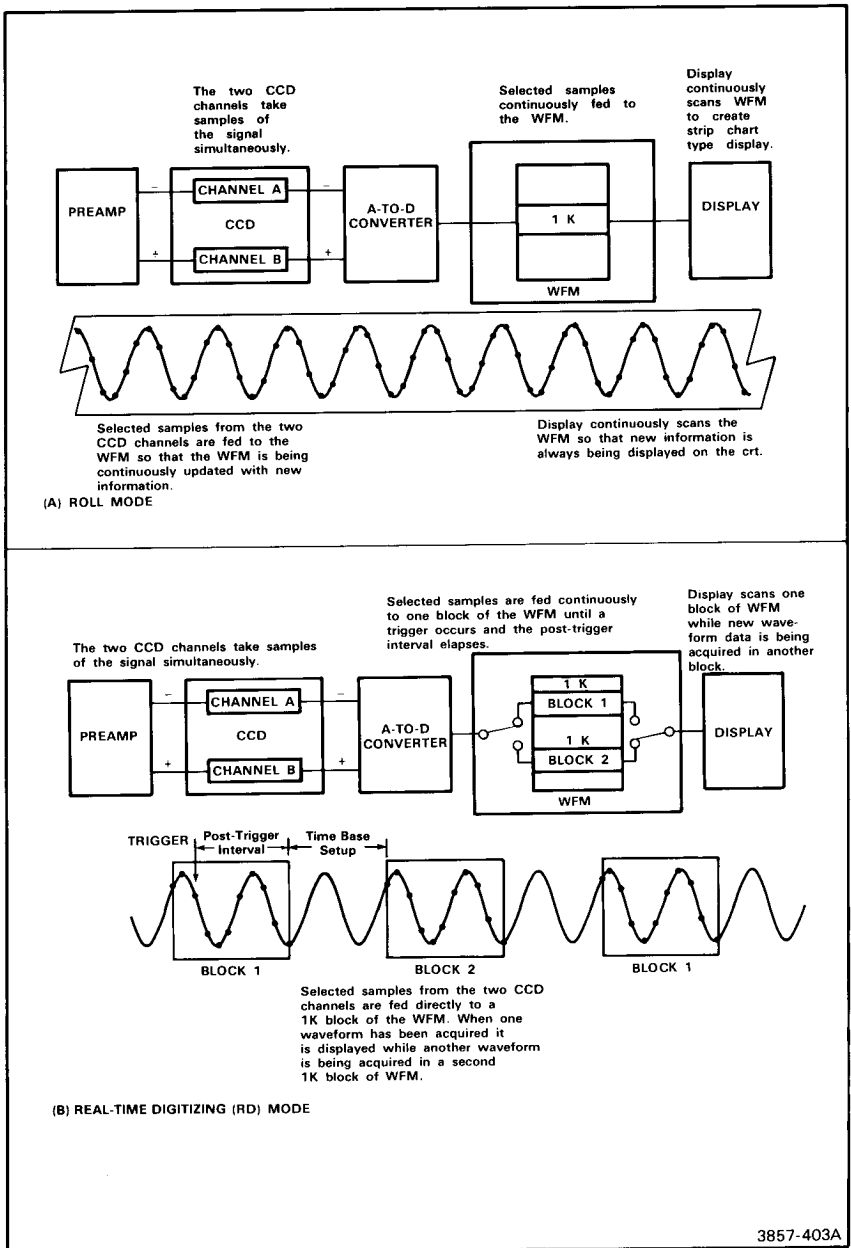
In the roll mode, a trigger is not required when the front-panel TRIGGERING MODE is set for P-P, AUTO or NORM, since the Waveform Memory is continuously being filled with new waveform information, which is in turn being displayed on the crt. In the HOLD NEXT mode, however, waveform acquisition can be halted a selected number of samples following a trigger event. The acquired waveform is then held in the Waveform Memory.

In the real-time digitizing (RD) mode (see Fig. 3-3B), the CCD also continuously samples the input signal at a 400 kHz rate. The A/D converter digitizes each sample and selected samples of the signal are stored in a 1K block of the the Waveform Memory, again at a rate determined by the time/div. When a trigger occurs, the storage of waveform samples continues until a complete waveform is stored at which time signal acquisition is halted. Compensation for the 455 sample delay through the CCD is included. The amount of post-trigger or pretrigger selected determines which portion of the signal is stored in the Waveform Memory with respect to the trigger event. Once the waveform has been stored in the Waveform Memory, the time base is reset and another waveform is acquired. This waveform, however, is stored in a second 1K block of Waveform Memory. While the second waveform is being acquired, the first waveform is displayed on the crt. This process is repeated continuously, switching back and forth between the two 1K blocks of Waveform Memory, such that the most recently acquired portion of the signal is always displayed.

In the extended real-time digitizing (ERD) mode (see Fig. 3-4A), a selected portion of the signal is first captured in the CCD, then written into the Waveform Memory. In this case, the two CCD registers continuously sample the input signal at a rate determined by the time/division setting. When the trigger event occurs, the CCD continues to sample the signal until the samples stored in the CCD correspond to the selected amount of pre-trigger or post-trigger desired. The samples held in the CCD are then shifted out at a 400 kHz rate, digitized by the A/D converter, and written into Waveform Memory. Once the waveform is stored in Waveform Memory, the time base is reset and another waveform is acquired. As with the RD mode, the second waveform is written into another block of the Waveform Memory, and the Display circuitry displays the most recently acquired waveform.

As the time/division settings are increased in the ERD mode, the CCD samples the signal at a much faster rate than the acquired waveform is digitized and written into memory. This mode of signal acquisition is thus called fast-in, slow-out.

Also, the two CCD channels sample the signal in a slightly different manner in the ERD mode. In the roll, RD and equivalent time digitizing modes, both CCD registers take a sample of the signal simultaneously. In the ERD mode, the two CCD registers take alternate samples of the signal.



3857-403A

Figure 3-3. Non-Fast In, Slow-Out Time Base Modes (A) Roll Mode; (B) Real-Time Digitizing (RD) Mode.

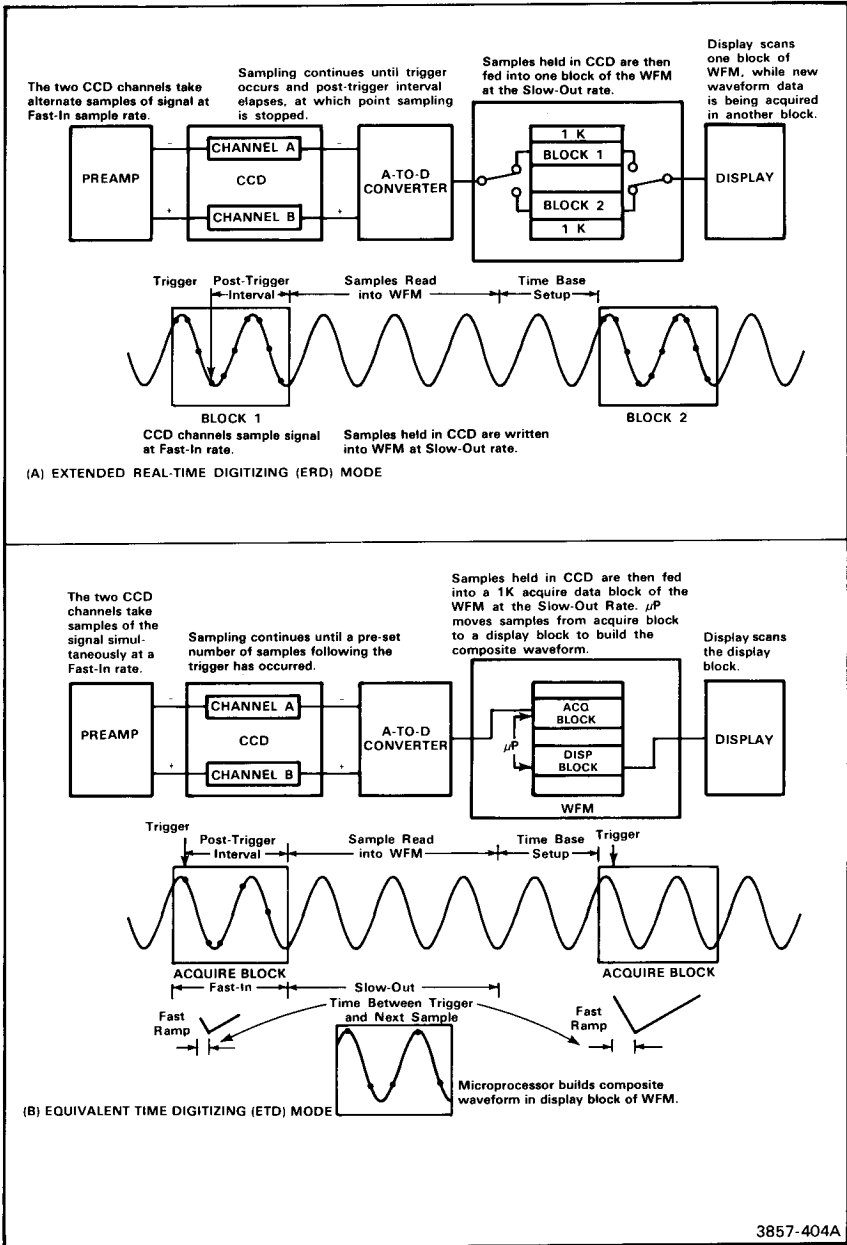


Figure 3-4. Fast-In, Slow-Out Time Base Modes; (A) Extended Real-Time Digitizing (ERD) Mode; (B) Equivalent Time Digitizing (ETD) Mode.

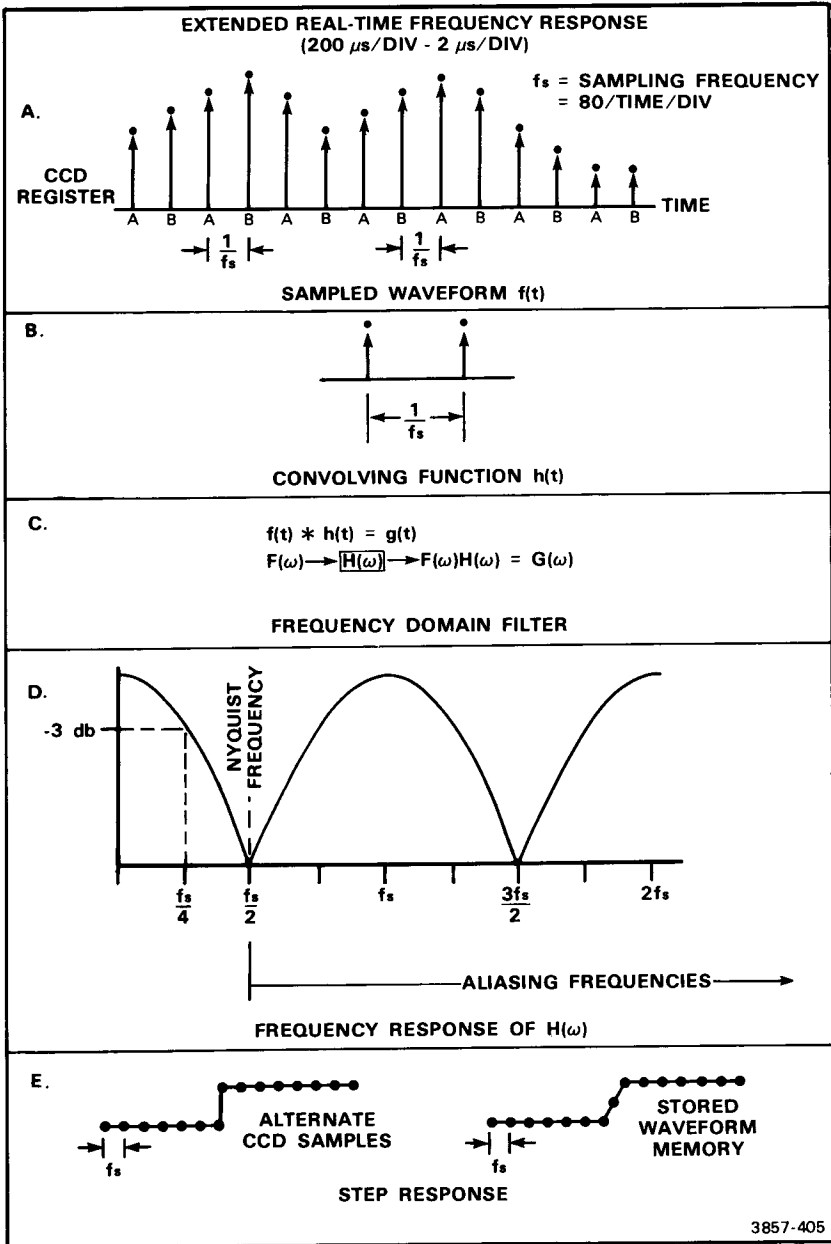


Figure 3-5. Extended Real-Time signal acquisition capability of the 7D20.

This technique is used to increase the real-time signal acquisition capability of the 7D20. Since the portion of the waveform that can be captured in this mode is dependent on the combined length of the two registers of the CCD, the time resolution in this mode is reduced from 100 samples per division to 80 samples per division. In addition, as the samples are alternately shifted out of the two CCD registers they are still combined differentially at the output and sent to the A/D converter for digitizing. This improves the CCD noise and linearity performance while introducing the characteristic cosinusoidal frequency response of a 2-point smooth function.

Shown in Figure 3-5A is a sampled waveform in the ERD mode where alternate samples are stored in each of the two registers of the CCD. Shifting samples alternately through the CCD and combining them differentially at the output is equivalent to convolving the sampled waveform with the two unit impulses shown in Figure 3-5B. As shown in Figure 3-5C this convolution is equivalent to passing the input frequency spectrum through a filter with the frequency response  $H(\omega)$ .  $H(\omega)$ , the Fourier transform of the convolving function  $h(t)$ , is shown in Figure 3-5D. Note that at one-half the Nyquist frequency, the frequency response is  $-3$  dB. Of course frequencies beyond the Nyquist frequency must be eliminated or aliasing will occur as with any digitizing system. As shown in Figure 3-5E this bandwidth limiting reduces the system rise time shown in the waveform step response.

In the equivalent time digitizing (ETD) mode (see Fig. 3-4B), a composite waveform is built up in a 1K block of the Waveform Memory from a number of waveform trigger events. As with the ERD mode, the CCD samples the signal at a fast rate, then writes the samples held in the CCD into Waveform Memory at a slower rate. In the ETD mode, the fast-in sample rate is fixed at 50 ns per point. In the ETD mode, however, only a limited number of samples are taken from each trigger event, and alternate sampling is no longer used (see Table 3-2). From subsequent triggers, more samples are taken until an accurate composite representation of the waveform is built up in the Waveform Memory.

**TABLE 3-2**  
**ETD Mode**

TIME/DIV	No. Points Acquired per Trigger Event
1 $\mu$ s	204-205
500 ns	102-103
200 ns	40-41
100 ns	20-21
50 ns	10-11

In order to determine where to store the set of samples in the Waveform Memory with respect to samples taken from other trigger events, the Time Base measures the time interval between the trigger event and the next sample taken with a fast

ramp time interpolator. The sample clock and the signal being measured are asynchronous, so a complete representation of the signal is built up after a number of trigger events.

Since a number of trigger events are required to create a waveform in the ETD mode, a complete 1024 point waveform can only be created when measuring repetitive signals. Since multiple points are taken from each trigger event the 7D20 can digitize a complete waveform from a low repetition-rate signal much faster than products that digitize only a single point per trigger.

## WAVEFORM DISPLAY

Waveform displays are produced by converting 8-bit vertical waveform memory data to analog signals. Hardware processing is applied for scaling, positioning, vector filtering, and conversion to a differential signal for driving the mainframe vertical interface amplifiers. Horizontal deflection is accomplished by digital to analog conversion of a 10-bit counter in the case of a Y versus T waveform or conversion of 8-bit X waveform data for a Y versus X waveform. Horizontal display magnification (X10) is accomplished digitally and is available for Y versus T waveforms. The analog horizontal signal is also processed for vector displays and conversion to a differential signal for driving the mainframe horizontal interface amplifiers.

Front-panel calibration controls permit adjusting the vertical and horizontal display gain and offset to match the calibration tolerance of all 7000-series mainframes. This procedure is given in Display Calibration, Section 2, Operating Instructions. Table 3-3 shows the position of the waveform display relative to the graticule of an ideal mainframe-crt system:

**TABLE 3-3**  
**7D20 Waveform Display Position**

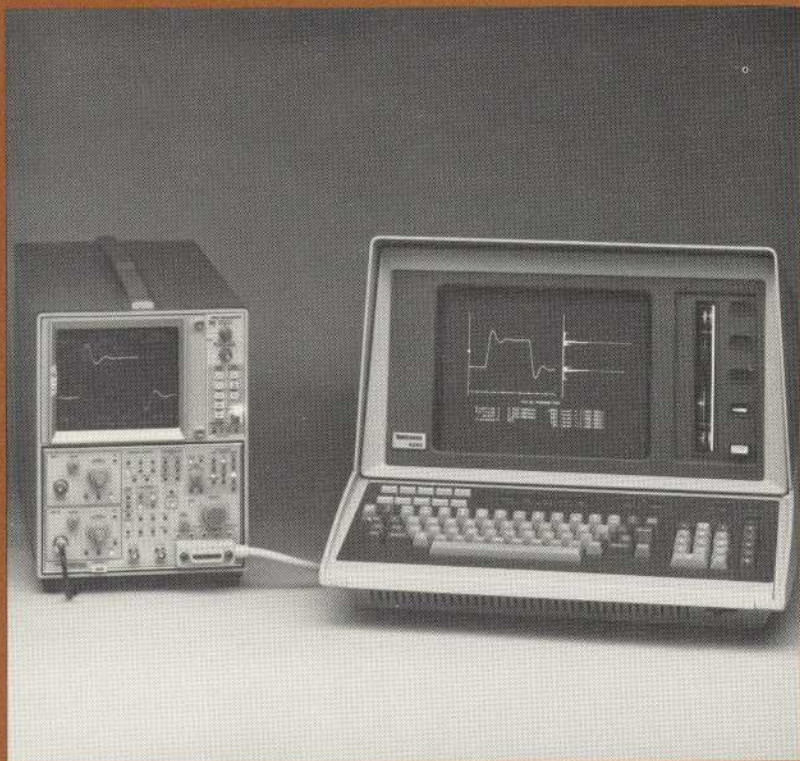
HORIZONTAL GRATICULE DIVISION	POINT #		HORIZONTAL BIT LEVEL (VS mode)
	1024 P/W	820 P/W	
-0.125	—	0	—
-0.12	0	—	0
0	12	10	3
1	112	90	28
2	212	170	53
3	312	250	78
4	412	330	103
5	512	410	128
6	612	490	153
7	712	570	178
8	812	650	203

TABLE CONTINUED ►

**TABLE 3-3 (CONT)**  
**7D20 Waveform Display Position**

HORIZONTAL GRATICULE DIVISION	POINT #		HORIZONTAL BIT LEVEL (VS mode)
	1024 P/W	820 P/W	
9	912	730	228
10	1012	810	253
10.08	—	—	255
10.11	1023	—	—
10.1125	—	819	—
VERTICAL GRATICULE DIVISION			VERTICAL BIT LEVEL (VXPD, VCMP OFF)
5.08			255
5			253
4	TOP GRATICULE		228
3			203
2			178
1			153
0	CENTER GRATICULE		128
-1			103
-2			78
-3			53
-4	BOTTOM GRATICULE		28
-5			3
-5.12			0





This section will introduce you to the increased versatility of the 7D20 Programmable Digitizer when interfaced with a computer, or other programmable device.

The section begins with a description of the GPIB system and the selection of various GPIB functions. Various waveform transfers, text, and readout formats are discussed in detail. A complete listing of all error codes with explanations, commands and queries, and sample programs using the Tektronix 4041 and 4050-series controllers, are included in this section.

# SECTION 4 CONTENTS

---

INTRODUCTION .....	4-1
GPIB DESCRIPTION .....	4-2
POWER UP .....	4-3
GPIB MODE, TERMINATOR, AND ADDRESS SELECTION .....	4-4
REMOTE AND LOCAL OPERATION .....	4-7
INDICATORS .....	4-8
GPIB INTERFACE FUNCTIONS .....	4-8
MESSAGES .....	4-9
GPIB INTERFACE MESSAGES .....	4-9
DEVICE-DEPENDENT MESSAGES .....	4-11
SERVICE REQUESTS .....	4-25
USING SRQ .....	4-25
COMMAND LIST .....	4-38
SAMPLE PROGRAMS .....	4-77



Digitally signed by  
<http://www.aa4df.com>

## INTRODUCTION

If you have read the previous sections of this manual, you are now familiar with the basic front panel operation of the 7D20. Reading the previous sections will help you make full use of the 7D20 GPIB command set capabilities. This section is intended to introduce you to the added capabilities that accrue from interfacing the 7D20 to a computer. The 7D20 uses the IEEE-488-1978 (GPIB) interface standard, which allows easy communication with most computers and other programmable devices. Table 4-2 later in this section lists the interface functions that the 7D20 implements.

In addition to the electrical standard followed by the 7D20 (IEEE-488-1978), the unit conforms to the GPIB CODES, FORMATS, CONVENTIONS AND FEATURES STANDARD that is used in the design of Tektronix, Inc. GPIB products. This standard provides precise language symbols that make programming the 7D20 no more difficult than operating it from the front panel.

### KEY FEATURES

The 7D20 has been designed with the end user in mind. To this end, a number of key features are included in the instrument:

- Text Messages
- Bidirectional Waveform Transfers
- Nine Separate User Requests
- Debug Command

**Text Messages.** To eliminate distracting written procedures, messages to the operator can be displayed on the crt. Other items of text, such as computer results and documenting remarks, can also be displayed by the same means.

**Bidirectional Waveform Transfers.** Waveform data may be sent to or from any one of the six waveform locations. This allows the acquired signal to be sent to a computer for processing or storage. Sending a stored waveform to the 7D20 allows the user to view the current signal against a known reference or against a limit waveform. The 7D20 is capable of transferring waveform data in both ASCII and Binary format (see Data Transfers in this section).

**Nine Separate User Requests.** The 7D20 can generate up to nine different user requests to the computer, which enhances the interactive capabilities and allows the program to better respond to the user's needs. Two of the user requests are generated by pressing the Probe Identify buttons on the body of the P6053B Probe.

One example where the user request functions are useful is for an operator to signal the computer to advance to the next test sequence, and set up the 7D20 as required. (See Service Requests in this section).

**Debug.** This is a highly useful command for the programmer, both during system development and while troubleshooting a problem. When the "DEBUG ON" command is received, all command data sent to the 7D20 is displayed on the crt. If a syntax or functional error is detected by the 7D20, an error statement indicating the detected problem is displayed just after the error.

### SECTION CONTENTS

This section consists of five major parts:

**GPIB Description.** This part highlights the basic operation of the interface. In addition, it contains information on getting started with the GPIB and selecting the various GPIB functions (address, terminator, etc.).

**Messages.** This part discusses the 7D20 command and query syntax. The various waveform transfers are covered in detail, along with the text and readout formats. In addition, the 7D20 response to IEEE-488-1978 defined interface commands is discussed.

**Service Request.** This part discusses the capability of the 7D20 to detect and report system status and user requests. A listing of all possible event codes is also included, along with explanations of each.

**Command List.** This part lists and explains all 7D20 bus commands and queries, and includes the exact syntax for each.

**Sample Programs.** To help the user get started in programming the 7D20, this part includes routines for both the Tektronix 4041 and 4050-series controllers. For each type of controller, both I/O and application routines are included.

## GPIB DESCRIPTION

The GPIB is an interface system that transfers signal information over a sixteen-line bus in a bit-parallel, byte-serial format. The bus consists of eight data lines, three handshake lines, and five bus management lines. The handshake lines permit asynchronous operation among instruments having different data transfer rates, so long as all conform to the handshake state diagrams and other provisions of IEEE-488. The data transfer rate is limited by the slowest active instrument connected to the bus.

The bus can be either a star or linear configuration, or a combination of both (see Fig. 4-1). To maintain the electrical characteristics of the bus, a device load must be connected for each two metres of cable, and at least one-half the devices on the bus must be operating.

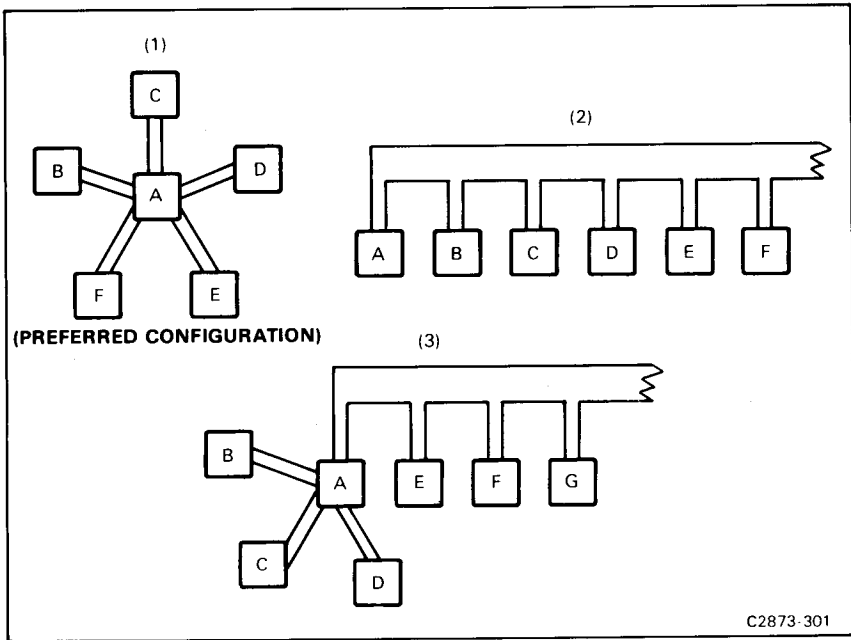


Figure 4-1. GPIB system configuration.

A minimum GPIB system for the 7D20 consists of the 7D20 and a GPIB controller (e.g., a Tektronix 4050-series controller). The controller directs all command and data transfers on the GPIB. A larger GPIB system may consist of up to 15 instruments distributed over a total cable length of no more than 20 metres. These instruments may include, in addition to the 7D20 and controller, other talkers (e.g., counter, digital multimeter, etc.) and other listeners (e.g., line printer, tape drive, programmable signal generator, etc.).

## POWER UP

When power is applied to the 7D20, a self-test routine is automatically invoked. At the completion of a successful test routine, the 7D20 comes on line with the same settings that existed when power was removed, with some exceptions. Power-up conditions are discussed more fully in Section 2 of this manual.

The 7D20 performs a self-test routine that tests 63 circuit modules. If a fault condition is detected, an SRQ (service request) is generated with an Internal Error status byte. An event query may be used to determine the specific module at fault. The controller may then issue a command to continue or exit the self-test routine. When the self-test is complete, an SRQ and an Operation Complete status byte are generated. Upon completion or exit from the self-test routine, the front-panel

settings are restored as before, and the waveform memory is initialized. The commands for the self-test routine are:

<b>TEST START</b>	Initiates self-test routine.
<b>CONTINUE</b>	Continues self-test routine after halting on fault condition.
<b>EXIT</b>	Exits self-test routine after halting on fault condition. This will not terminate an executing self-test.
<b>TEST</b>	Same as Test START.

Refer to the Operating Instructions and the Service Manual Maintenance Instructions for more complete information regarding self-test.

**NOTE**

- 1. The colored characters in the commands are what the 7D20 will accept as command abbreviations. See Command Usage later in this section for more information.*
- 2. The vertical line following TEST START AND CONTINUE means exclusive OR.*

**GPIB MODE, TERMINATOR,  
AND ADDRESS SELECTION**

**NOTE**

*The firmware version displayed in these illustrations may differ from those displayed for your 7D20, and are shown for positional reference only.*

The 7D20 provides for the selection of the mode, terminator, and bus address by means of the front-panel numeric keypad (located in the MEMORY DISPLAY section), in conjunction with the ID menu that is displayed by the oscilloscope in which the 7D20 is installed. The keys that are used for function selection are:

- Key #4: SELECT MODE
- Key #5: SELECT TERMINATOR
- Key #6: SELECT ADDRESS

The selection process is initiated by pressing the front-panel ID key, which changes the numeric keypad from waveform selection to function selection, and produces the identification message. The selection process is terminated by again pressing the front-panel ID button. When this occurs, all changes that were previously input are implemented, unless the 7D20 is in REMOTE ONLY. If so, the prompt "UPDATE IGNORED" will appear on screen. This prevents the bus from being changed during the selection process. Figure 4-2 illustrates the display when ID is pressed.

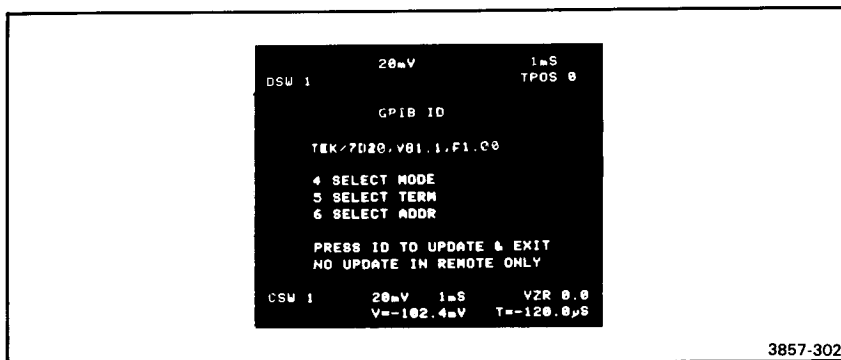


Figure 4-2. Function selection display.

**NOTE**

*All function choices are stored when power is removed, and restored when the power is resumed.*

**ADDRESS SELECTION**

To select the GPIB address, press front-panel key #6 (MEMORY DISPLAY). The prompt field will display the current GPIB address as "ADDR=<addr>?". The question mark and key #6 will both blink. Repeatedly pressing key #6 increments the address number, beginning with the current address. The address will increment to 30 and begin again at zero.

Figure 4-3 shows the display when the GPIB address is 13.

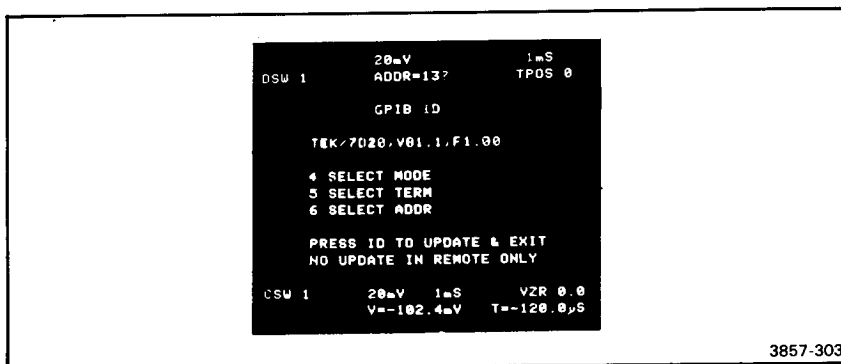


Figure 4-3. Address selection display.

### TERMINATOR SELECTION

To select the terminator, press key #5. The prompt field will display the current status of the GPIB terminator as "TERM=<term>?". Key #5 and the question mark will blink. Repeatedly pressing the key alternates the terminator between EOI and LF/EOI. Refer to controller manual for proper terminator choice.

Figure 4-4 illustrates the display when the LF/EOI terminator is selected.

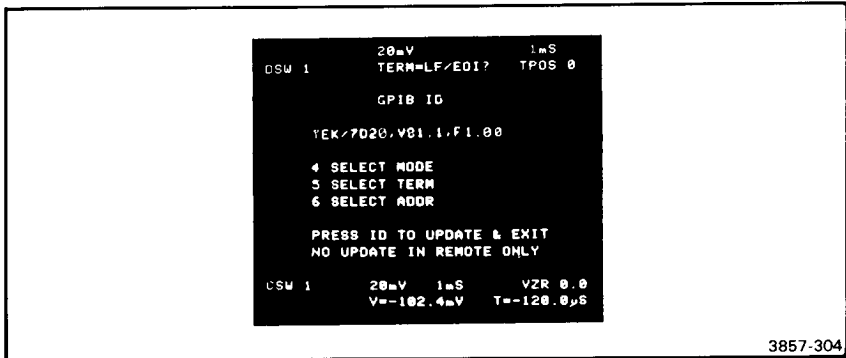


Figure 4-4. Terminator selection display.

### MODE SELECTION

To select the GPIB mode, press key #4. The prompt field on the oscilloscope display will show the current status of the GPIB mode as "MODE=<mode>?". (Key #4 and the question mark will blink.)

Repeatedly pressing the key will sequence the mode selection through off bus, talk only, listen only, and talk/listen.

Figure 4-5 illustrates the oscilloscope display when the talk/listen mode is selected.

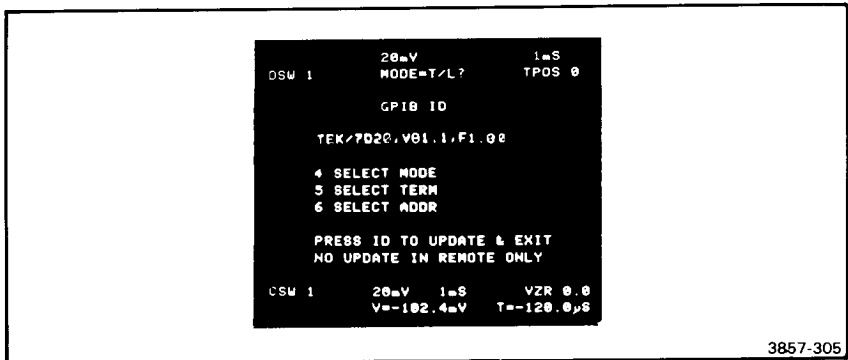


Figure 4-5. Mode selection display.



## ALTERNATE GPIB STATUS INDICATORS

In some cases, the oscilloscope display may not be visible, or may not be functioning correctly, in either case making normal function selection impossible. The light pattern of the numeric keypad displays the GPIB function status. Table 4-1 lists the function selection states. The table is arranged in 8421 binary code.

**TABLE 4-1**  
**MEMORY DISPLAY Keypad Code**

Indicated Function	Keypad					
	1	2	3	4	5	6
Select Mode						
OFF BUS	0	0	0	B	0	0
TALK ONLY	1	0	0	B	0	0
LISTEN ONLY	0	1	0	B	0	0
TALK/LISTEN	1	1	0	B	0	0
Select Terminator						
LF or EOI	0	0	1	0	B	0
EOI	0	0	0	0	B	0
Select Address						
30	0	1	1	1	1	B
29	1	0	1	1	1	B
Binary Address	A0	A1	A2	A3	A4	B
1	1	0	0	0	0	B
0	0	0	0	0	0	B

1 - ON; 0 = OFF; B = blinking

To terminate the function selection, press the illuminated ID key, which erases the displayed menu, implements the changes just made, and extinguishes the ID key. Also, the keypad is restored to the normal waveform selection state.

## REMOTE AND LOCAL OPERATION

The 7D20 complies with the requirements noted in IEEE-488 for the Remote-Local interface function, which provides the instrument with the capability to select between front-panel information and GPIB information sources. The 7D20 will, however, execute all GPIB commands and queries regardless of the remote-local state.

Each front-panel control on the 7D20 generates a local "rtl" message when used. If the 7D20 is in either LOCS (local state) or LWLS (local with lockout state), the front-panel function will be executed. If the 7D20 is in RWLS (remote with lockout state) only GPIB commands are executed except for the following functions: probe

encoding, probe identify, RQS, RQS#, and monitoring the status of the GPIB selection via the ID pushbutton. Also, the HORIZONTAL POSITION and VOLTS/DIV VARIABLE controls operate unless set to calibrated status via GPIB commands.

## **INDICATORS**

The front-panel ADDR (addressed) indicator lights whenever the 7D20 is addressed by the controller, which occurs in one of the following address states: TACS (Talker Active State), TADS (Talker Addressed State), LACS (Listener Active State), or LADS (Listener Addressed State). The front-panel SRQ indicator lights when a 7D20 service request is generated. This is cleared by a poll of the 7D20 by the controller. The REMOTE ONLY indicator lights when the 7D20 is in RWLS (Remote With Lockout State).

## **GPIB INTERFACE FUNCTIONS**

IEEE-488-1978 defines the GPIB interface functions and the allowed subsets of these functions. The subsets that apply to the 7D20 are listed in Table 4-2. Refer to IEEE-488-1978 for more definition regarding the functions.

**TABLE 4-2**  
**7D20 GPIB Interface Functions**

<b>Interface Function</b>	<b>Subset</b>	<b>Capability</b>
Source Handshake	SH1	Complete
Acceptor Handshake	AH1	Complete
Talker	T5	Complete (no secondary addressing)
Listener	L3	Complete (no secondary addressing)
Service Request	SR1	Complete
Remote Local	RL1	Complete
Parallel Poll	PPO	None
Device Clear	DC1	Complete
Device Trigger	DT1	Complete
Controller	C0	None
Electrical Interface	E2	Tri-state

# MESSAGES

GPIB messages fall into two major classes: Interface messages and device-dependent messages. Interface messages are sent by the GPIB controller to all instruments on the bus to control the functions of each instrument interface. Device-dependent messages (e.g., instrument commands, data, etc.) may be sent by an instrument on the bus to any other instrument on the bus. Refer to Figure 4-6, which illustrates the relationship of interface messages and device-dependent messages.

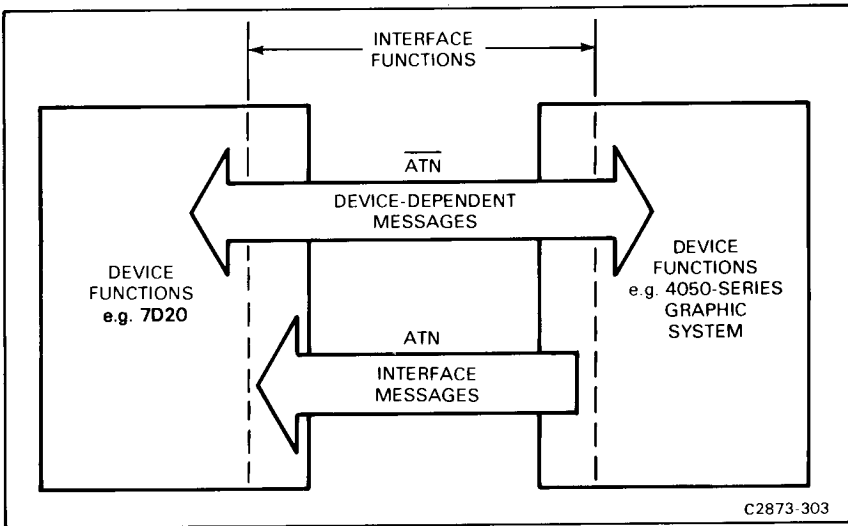


Figure 4-6. Device-dependent vs. interface messages.

## GPIB INTERFACE MESSAGES

The controller uses interface messages to manage the bus (e.g., designate talkers and listeners, etc.). Each instrument on the bus listens to every interface message. The interface messages in the controller's vocabulary are defined by IEEE-488-1978. The messages can be considered to be ASCII codes that assume a new meaning when sent by the controller with the Attention (ATN) line asserted.

### TALK AND LISTEN ADDRESSES (MTA, MLA)

Two of the interface messages are the talk and listen addresses. When a device sees its talk address (called My Talk Address or MTA) and ATN asserted simultaneously, it must become a talker. When the controller removes ATN, the device begins the source handshake to transmit its data. Similarly, My Listen Address (MLA) and ATN tells a device to listen to the data sent by a talker.

### **SERIAL POLL ENABLE COMMAND (SPE)**

The controller uses other kinds of interface messages for other tasks. One is the Serial Poll Enable command (SPE), which is used with the service request function. Assume that an instrument asserts SRQ when it completes acquiring data. The controller must poll all devices on the bus, since any one or all devices can assert SRQ. To conduct the serial poll, the controller sends SPE, a universal command, then addresses each device in turn as a talker and reads a status byte from each. Any device asserting SRQ also asserts DIO7 of the status byte to inform the controller that it is actively requesting service. The remainder of the status byte tells the controller why it is requesting service. (For more information on this topic, refer to the service request discussion later in this section.)

### **GROUP EXECUTE TRIGGER (GET)**

The Device Trigger function uses an addressed command: Group Execute Trigger (GET). Refer to Table 4-7 later in this section for more information.

### **DEVICE CLEAR (DCL)**

The controller issues the Device Clear message (DCL) to initialize internal functions of devices on the bus. DCL is a universal command that applies to all devices. Its effect on each instrument, however, is decided by the designer, who can choose to initialize any device function to any state that suits the purpose of the instrument.

The 7D20 complies with all requirements set forth by IEEE-488-1978 with regard to the Device Clear function. When Device Clear is accepted, the following occurs:

1. All bus communications in process are terminated without untalking or unlistening the 7D20.
2. All pending SRQ flags except POWER ON are cleared.
3. All pending event query responses except POWER ON are cleared.
4. Input and output buffers are cleared.

### **SELECTED DEVICE CLEAR (SDC)**

The 7D20, if listen-addressed, responds to a Selected Device Clear (SDC) in the same way as described for Device Clear.

### **INTERFACE CLEAR (IFC)**

The IFC (Interface Clear) message interrupts any data input or output, and places the 7D20 in the unaddressed state. If the 7D20 is talking when IFC is asserted, it will continue from the point of interruption when again addressed as a talker. Likewise, if listening when interrupted, the 7D20 will resume receiving data when again addressed as a listener.

## DEVICE-DEPENDENT MESSAGES

Device-dependent messages, which are 7D20 commands, data, etc., are sent from the controller or another device (talker) to one or more other instruments (listeners) on the bus. Device-dependent messages affect the functions of the instruments (e.g., commands), but they do not directly affect the actual GPIB interface.

### NOTE

*BNF notation is used herein to define communications with the 7D20. The symbols are defined as follows:*

- < > Defined element
- ::= Is Defined As
- { } Grouping
- [ ] Optional, May Be Omitted
- | Exclusive Or (one or the other, but not both)
- ... May be repeated one or more times.

### INTRODUCTION

The 7D20 Command set is divided into two classes: Commands, which direct an action to be taken, and Queries, which interrogate the 7D20 concerning the state of the 7D20.

### COMMAND USAGE

A command can be one of two types: The first and more complex is illustrated by Figure 4-7. Note that multiple arguments are permitted if separated by a comma. The command consists of a header, followed by a link argument, in turn followed by data. The term header is used for the class of words that select one of a group of

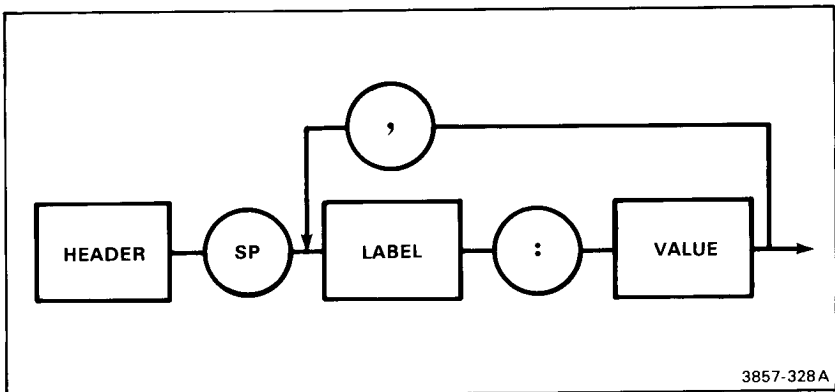


Figure 4-7. Complex command format diagram.

functions. For example, the Channel 1 Vertical Amplifier has four functions: Volts/Div, Position, Coupling, and Variable. The command:

**CH1 VOLTS: .2, POSITION: -1, COUPLING: AC, VARIABLE: ON**

specifies the four functions. The header is CH1, the link arguments are VOLTS, POSITION, COUPLING, and VARIABLE. The data is 0.2, -1, AC, and ON.

The second form of command consists of a header that is followed by an argument, having no data involved. Examples are:

**DEBUG ON  
RQS ON  
INIT ALL**

Refer to Figure 4-8, which illustrates the format of the simpler command.

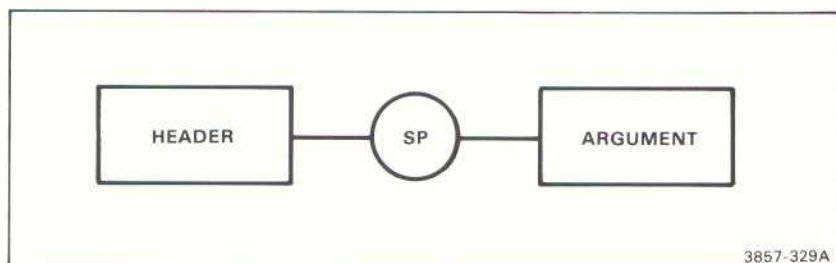


Figure 4-8. Simple command format diagram.

Multiple commands may be sent to the 7D20 in one GPIB message by inserting a semicolon (;) between the commands:

**CH1 POSITION:3.5,COUPLING:GND;CH2 VOLTS:.1**

This command list sets the channel 1 position to 3.5 divisions and coupling to GND, and the channel 2 volts/div to 0.1.

Header and argument may be abbreviated as long as the abbreviations are as specified herein. For example:

**TRIGGER MODE:P-P,COUPLING:ACHFREQ,SLOPE:PLUS  
TR MO:P-,COUP:ACH,SL:PL**

are identical commands. In this section, the minimum command required is printed in color, with the remainder of each word verified if present. Any additional characters will cause an error message to be sent.

The numeric arguments acceptable to the 7D20 comply closely with the ANSI 3.42 Standard on Representation of Numeric Values in Character Strings for Information

Interchange. The three data types are integer (NR1), floating-point (NR2), and exponential notation (NR3). In all three types, the 7D20 will not send numeric representations of zero that contain a minus sign. NR1, NR2, and NR3 data can be used interchangeably anywhere within the 7D20 syntax.

NR1 integer representation is most useful for numeric data where either limited interpretation or generation of data is required (e.g., restricted or fixed-range) and also where large volumes of fixed-format data or high-speed data are transferred between devices. In these applications, data rates are of primary significance. Figure 4-9 illustrates the NR1 data format.

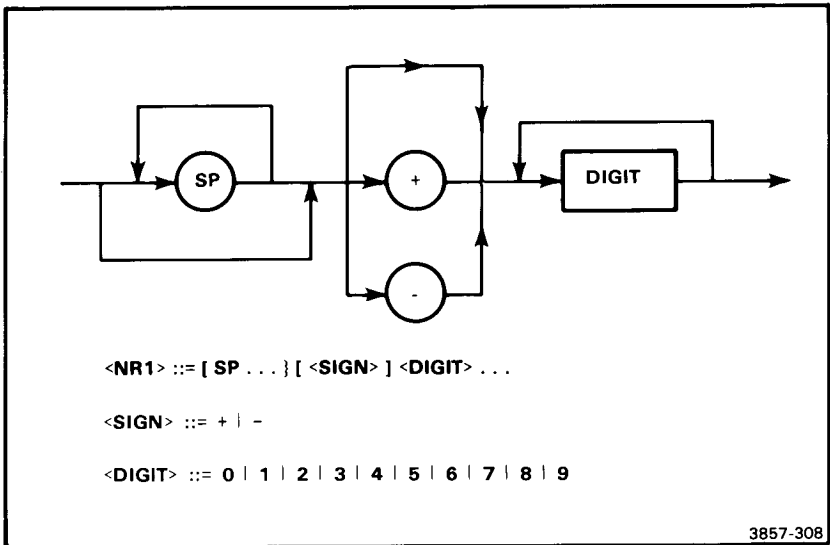


Figure 4-9. NR1 data format.

NR2 floating-point representation is most useful for numeric data where the range of data output is limited or data is intended to be used with devices where human interpretation prevails. Figure 4-10 illustrates the NR2 data format.

NR3 exponential notation is preferred where measurement or controller devices must accommodate a wide range of values or where the specific range of data to be output (or input) is unpredictable. Figure 4-11 illustrates the NR3 data format.

## QUERY USAGE

A query solicits information from the 7D20. Queries consist of either a header followed by a '?' or a header followed by a '?', then one or more arguments. A query consisting of only a header followed by a question mark addresses the entire group of information; a query consisting of a header followed by an argument addresses an individual subset of the group. Query responses are formatted using the same

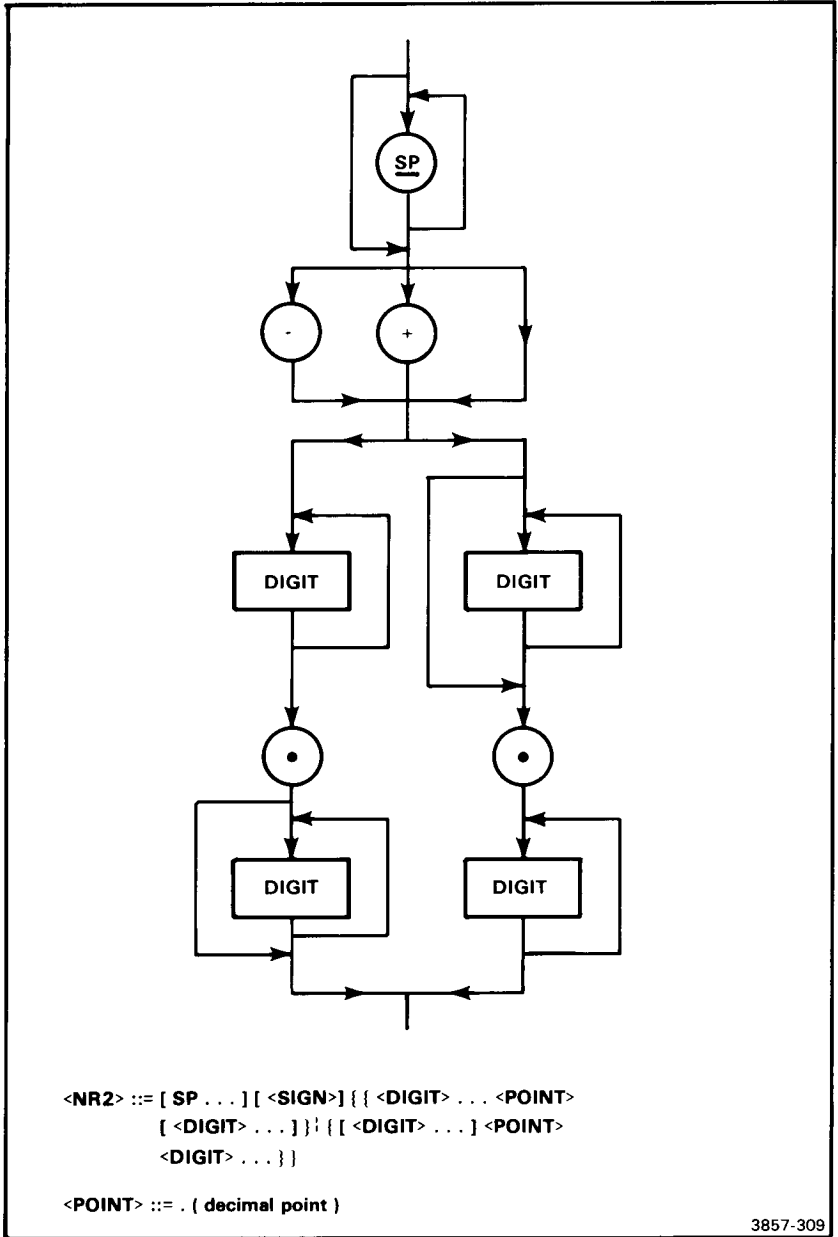


Figure 4-10. NR2 data format.



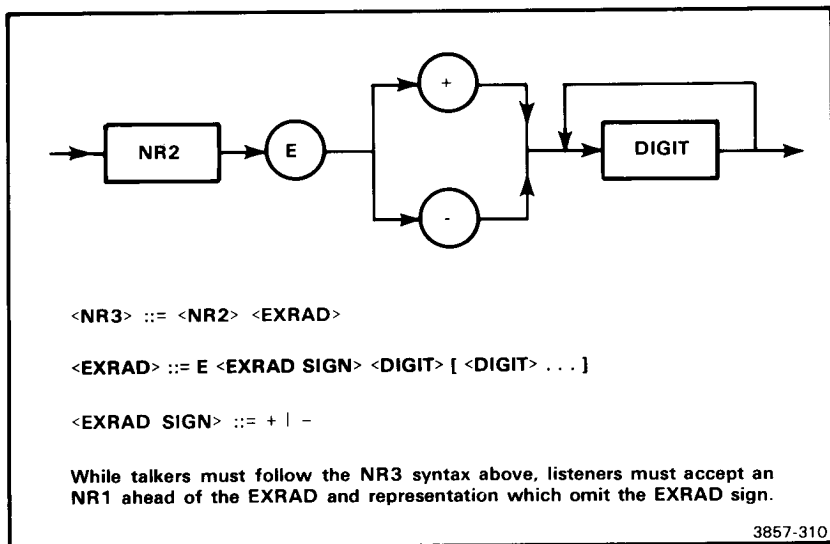


Figure 4-11. NR3 data format.

syntax as commands. Most responses can be sent directly back to the 7D20 as a command to set up the identical condition that existed when the response was generated. Some responses indicate the status of execution of 7D20 commands. These may be returned to the 7D20 as commands, but will be ignored by the instrument. The query format is depicted in Figure 4-12

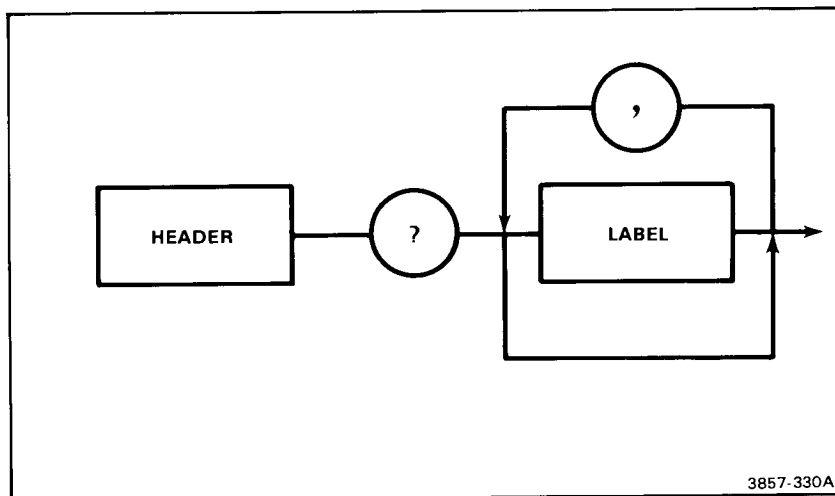


Figure 4-12. Query format diagram.

As an example, the header-only query, "CH1?" would solicit all the channel 1 settings. The response would be:

**CH1 VOLTS: .2, POSITION: -1, COUPLING: AC, VARIABLE: OFF, PROBE: 1**

if the channel 1 volts/div was 0.2 volts, position was -1 divisions, coupling was AC, channel 1 was calibrated (VARIABLE control was disabled), and channel 1 was connected to a X1 attenuation probe.

To solicit only the volts/div of channel 2, the query "CH2? VOLTS" would result in the following response:

**CH2 VOLTS: 2.0**

if the channel 2 volts/div setting was 2 volts.

Again, as with commands, the header and argument can be abbreviated as long as the abbreviations are as specified. For example,

**TRIGGER? MODE** and  
**TR? MO**

are identical queries. It should also be noted that the 7D20 must be talked once for each query with a terminator, but only once after a set of queries separated by a semicolon and followed by a terminator. The 7D20 will cancel any previous unread (?) Query responses upon receipt of a new command or query.

## DATA TRANSFERS

In the 7D20, data transfers take three major forms:

1. Readout, which conveys data about the operational settings of the 7D20, and the cursor waveform.
2. Text, which conveys messages to and from the crt by way of the GPIB.
3. Waveforms, which consist of curve information for transmission over the bus.

The 7D20 I/O Bus has the following characteristics:

1. The input buffer of the 7D20 is 128 characters deep. When the input buffer is full, the 7D20 will be unable to accept any more data until buffer space becomes available.
2. The output buffer of the 7D20 is 128 characters deep. When 128 characters are loaded into this buffer, it is considered full.
3. If a query is sent that fills the output buffer before the 7D20 has been talked and the input buffer is full, the output buffer is emptied of contents, and an execution error status byte is reported. An EVENT? at this time would evoke an event report of "I/O buffers full, output dumped", the code for which is 203. It is possible to send a form: Query, command, command... and fill the input buffer before talking the 7D20. This will create an I/O deadlock. To avoid loss of data, the controller should talk the 7D20 with each query sent; otherwise, an I/O deadlock may occur.

## Readout

When a readout query is received, the 7D20 responds by sending the information on the top two and bottom two lines of text displayed on the crt. Readout text can be transmitted to the 7D20, but it will be ignored. Syntax for the readout query is:

**RDOUT?**

and the response is:

**RDOUT “<line 1><CR><line 2><CR><line 15><CR><line 16><CR>”]**

## Text

The text is the information that is displayed on the center 12 lines of the crt display. Each line contains no more than 36 characters. Text can be sent or received by the 7D20. The characters that the 7D20 can display are listed in Figure 4-13, from character 32 (SP) through 95 (-), which form the center four columns of characters in the figure. If lower case letters (97 through 122) are received, they are converted to uppercase, then displayed. In addition, the characters in Table 4-3 can be sent or received:

**TABLE 4-3**  
**Supplemental Characters**

Character	ASCII Character	Decimal
Horizontal Tab	HT	9
Carriage Return	CR	13
(Up arrow)	^	94
(Down arrow)	ESC a	27 97
(Delta) $\Delta$	ESC d	27 100
(Milli) m	ESC m	27 109
(Micro) $\mu$	ESC u	27 117
(Nano) n	ESC n	27 110
(Pico) p	ESC p	27 112
(Right arrow)	ESC r	27 114

Note that the HT character is equivalent to sending five spaces in the text string.

When using the text feature, the user is cautioned that some limitations should be kept in mind:

1. The character set is formed with a  $5 \times 7$  standard dot matrix, and is imposed on a field of 1536 points. One can calculate the precise number of characters that can be displayed by summing the dots used in the characters until all 1536 points (dots) are used. More practically, the number of characters possible is approximately 100. Spaces or blanks do not count in the calculation of total characters.
2. A TEXT command will terminate a DEBUG display, or any MENU, TEST, or ID command and will also replace the text that is currently on the display.

# ASCII & IEEE 488 (GPIB) CODE CHART

BITS		0 0		0 0 1		0 1 0		0 1 1		1 0 0		1 0 1		1 1 0		1 1 1	
B4 B3 B2 B1		CONTROL				NUMBERS SYMBOLS				UPPER CASE				LOWER CASE			
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

## KEY

octal	25	PPU	GPIB code
	NAK		ASCII character
hex	15	21	decimal

3857-327

Figure 4-13. ASCII and IEEE-488 code.

3. Likewise, a DEBUG display, or any MENU, TEST, or ID command will terminate a TEXT command and replace the displayed text with that accompanying the new command.
4. Syntax for the TEXT command is:

```
TEXT "<line 3>[<CR><line 4><CR><line 5><CR>...
<line 12><CR><line 13><CR><line 14><CR>]"
```

The commands between brackets are optional.

5. If more than 36 characters are received for a single line, a carriage return is inferred and the remainder is placed on subsequent lines.
6. A TEXT query will always cause twelve lines to be sent. Thus, if only three lines of text characters are in the message, nine blank lines will accompany them.

## Waveforms

Up to six different waveforms may be acquired and stored in the 7D20 memory for later use. These can be selected at random, and can also be stored in any order.

The general format of the transmitted waveform data closely complies with the Tektronix Codes and Formats Standard, which specifies the form of transmission required for waveform data. The two parts of the waveform data are:

1. The preamble, which contains items such as waveform size, scaling information, format specifications, and similar items required to determine co-ordinate values, and auxiliary information like identification strings and units; and
2. The curve, a set of data that contains the curve co-ordinates and attributes. Waveform transmissions may include both preamble and curve data, or either one can be sent alone. Separate query messages may be used to elicit preamble data or curve data, or both.

The data received by the 7D20 may be in either the 1024 or the 820-point format. If insufficient points are sent by the CURVE command, the last point is repeated until the end of the waveform occurs.

Each curve, when interpreted in accordance with its corresponding preamble, constitutes a complete waveform transmission. For proper interpretation of curve data, a minimum preamble must be transmitted for the selected waveform destination before any waveforms are sent. Subsequent preambles sent need only transmit changed information. Curve data will be associated with the preamble previously stored at the same destination. Curve data may be sent in binary or ASCII format. Binary format is especially suited for transmitting large blocks of data at high speed, but is not easily readable in this form.

An example of binary encoding of a waveform curve follows:

Preamble:

```
WFMPRE WFID:W 1,ENCDG:BINARY,NR.PT:1024,PT.FMT:Y,
XINCR:1.0E-5,PT.OFF:1.2 E+1,XZERO:0,XUNIT:S,
YMULT:1.0,YZERO:0,YUNIT:V,BYT/NR:1,
BN.FMT:LF,BIT/NR:8,CRVCHK:CHKSMO
```

The ASCII format is more bulky than the binary format, so the transmission time is longer than that of binary, but the data is easily readable by comparison. An example of ASCII encoding of a waveform curve follows:

```
WFMPRE WFID:W 1,ENCDG:ASCII,NR.PT:1024,PT.FMT:Y,
XINCR:1.0E-5,PT.OFF:1.2E+1,XZERO:0,XUNIT:S,
YMULT:1.0,YZERO:0,YUNIT:V
```

**Waveform Preambles.** Following are the waveform commands, queries, and the definitions of each.

1. **Waveform Preamble Commands.** These commands provide the 7D20 with scale factor attributes, number of points per waveform, and the trigger reference location to be assigned to waveform data sent to the 7D20 by the CURVE command.

<b>WFMPRE NR.PT:</b> 820 1024	Sets number of points that are input from the CURVE command to 820 or 1024.
<b>WFMPRE XINCR:</b> <x increment>	Sets the time between points input from the CURVE command to <x increment>. <x increment> ::= time / div / 80 points/division for 820-point waveforms, and time / div / 100 points/division for 1024-point waveforms.
<b>WFMPRE PT.OFF:</b> <point #>	Selects the waveform point at which the cursor reads time zero. The range is $-150,000 \leq \text{point \#} \leq 1023$ , where point 0 is the left most displayed point on the waveform. An execution error warning is issued if out of range condition occurs.
<b>WFMPRE XUNIT:</b> S	Sets displayed horizontal scale factor unit for waveforms input from the CURVE command.

An example of binary encoding of a waveform curve follows:

Preamble:

```
WFMPRE WFID:W 1,ENCDG:BINAR,NR.PT:1024,PT.FMT:Y,
XINCR:1.0E-5,PT.OFF:1.2 E+1,XZERO:0,XUNIT:S,
YMULT:1.0,YZERO:0,YUNIT:V,BYT/NR:1,
BN.FMT:LF,BIT/NR:8,CRVCHK:CHKSMO
```

The ASCII format is more bulky than the binary format, so the transmission time is longer than that of binary, but the data is easily readable by comparison. An example of ASCII encoding of a waveform curve follows:

```
WFMPRE WFID:W 1,ENCDG:ASCII,NR.PT:1024,PT.FMT:Y,
XINCR:1.0E-5,PT.OFF:1.2E+1,XZERO:0,XUNIT:S,
YMULT:1.0,YZERO:0,YUNIT:V
```

**Waveform Preambles.** Following are the waveform commands, queries, and the definitions of each.

- 1. Waveform Preamble Commands.** These commands provide the 7D20 with scale factor attributes, number of points per waveform, and the trigger reference location to be assigned to waveform data sent to the 7D20 by the CURVE command.

**WFMPRE NR.PT:** 820|1024      Sets number of points that are input from the CURVE command to 820 or 1024.

**WFMPRE XINCR:** <x increment>      Sets the time between points input from the CURVE command to <x increment>.

<x increment> ::= time / div / 80 points/division for 820-point waveforms, and time / div / 100 points/division for 1024-point waveforms.

**WFMPRE PT.OFF:** <point #>      Selects the waveform point at which the cursor reads time zero. The range is  $-150,000 \leq \text{point \#} \leq 1023$ , where point 0 is the left most displayed point on the waveform. An execution error warning is issued if out of range condition occurs.

**WFMPRE XUNIT:** S      Sets displayed horizontal scale factor unit for waveforms input from the CURVE command.



**WFMPRE YMULT:** 1 <y multiplier> Sets the vertical scale factor (volts/div) for waveforms to input from the CURVE command to <y multiplier>.

**WFMPRE YZERO:** <y zero> Sets the vertical value of the center horizontal graticule division for waveforms input from the CURVE command. This corresponds to curve data of zero divisions for ASCII curves and 80<sub>hex</sub> for binary curves.

<y zero> ::= -volts/div \* VZR (vertical zero reference)

**WFMPRE YUNIT:** V Sets the displayed vertical scale factor unit for waveforms input by the CURVE command.

2. **Waveform Preamble Queries.** These queries provide the controller with the waveform attributes associated with the waveform curve data specified as the source waveform from the DATA command. The responses apply to curve data sent from the 7D20 in response to a CURVE? or WAVFRM? query.

**WFMPRE? WFID**<sup>1</sup> Responds with the current source waveform number and indicates if interpolation will be applied to curve data: **WFMPRE WFID:**<wfmid>

**WFMPRE? ENCDG**<sup>1</sup> Responds with the current curve encoding: **WFMPRE ENCDG:**<encdg>

**WFMPRE? NR.PT** Responds with the current source waveform's points/waveform: **WFMPRE NR.PT:**<p/w>

**WFMPRE? PT.FMT**<sup>1</sup> The 7D20 responds: **WFMPRE PT.FMT:**Y. This indicates that the curve data is vertical data with respect to time.

**WFMPRE? XINCR:** Responds with the current source waveform's time between points: **WFMPRE XINCR:**<x increment>

**WFMPRE? PT.OFF** Responds with the trigger point number: **WFMPRE PT.OFF:**<point #>

**WFMPRE? XZERO**<sup>1</sup> The 7D20 responds: **WFMPRE XZERO:**0. This indicates that the curve data horizontal origin has not been shifted.

<sup>1</sup>These queries have no corresponding 7D20 command. Query responses are ignored whenever returned to the 7D20 in command form, and no error message is sent.



<b>WFMPRE? XUNIT</b>	The 7D20 responds: <b>WFMPRE XUNIT:S PT</b> . This indicates that the curve data horizontal scale factor unit is seconds or points.
<b>WFMPRE? YMULT</b>	Responds with the current source waveform vertical scale factor: <b>WFMPRE YMULT:&lt;y multiplier&gt;</b>
<b>WFMPRE? YZERO</b>	Responds with the current source waveform vertical zero: <b>WFMPRE YZERO:&lt;y zero&gt;</b>
<b>WFMPRE? YUNIT</b>	The 7D20 responds: <b>WFMPRE YUNIT:V</b> . This indicates that the curve data vertical scale factor unit is volts.
<b>WFMPRE? BYT/NR<sup>1</sup></b>	The 7D20 responds: <b>WFMPRE BYT/NR:1</b> . This indicates that the curve data for binary transmission is one byte per point.
<b>WFMPRE? BN.FMT<sup>1</sup></b>	The 7D20 responds: <b>WFMPRE BN.FMT:LF</b> . This indicates that the curve data is sent in a left-justified binary fraction format.
<b>WFMPRE? BIT/NR<sup>1</sup></b>	The 7D20 responds: <b>WFMPRE BIT/NR:8</b> . This indicates that each binary byte has eight bits.
<b>WFMPRE? CRVCHK<sup>1</sup></b>	The 7D20 responds: <b>WFMPRE CRVCHK:CHKSMO</b> . This indicates that the last byte of the binary curve is a checksum.
<b>WFMPRE?</b>	Responds with the current source waveform's preamble:  if ASCII: <b>WFMPRE WFID:&lt;wfmid&gt;,ENCDG:ASCII, NR.PT:&lt;p/w&gt;,PT.FMT:Y,XINCR:&lt;x increment&gt;, PT.OFF:&lt;point #&gt;,XZERO:0,XUNIT:S, YMULT:&lt;y multiplier&gt;,YZERO:&lt;y zero&gt;, YUNIT:V</b>  if binary: <b>WFMPRE WFID:&lt;wfmid&gt;,ENCDG: BINARY,NR.PT:&lt;p/w&gt;,PT.FMT:Y, XINCR:&lt;x increment&gt;,PT.OFF: &lt;point #&gt;,XZERO:0,XUNIT:&lt;x unit&gt;, YMULT:&lt;y multiplier&gt;,YZERO:&lt;y zero&gt;, YUNIT:V,BYT NR:1,BN.FMT:LF, BIT NR:8,CRVCHK:CHKSMO</b>

<sup>1</sup>These queries have no corresponding 7D20 command. Query responses are ignored whenever returned to the 7D20 in command form, and no error message is sent.

where:

<wfm #> ::= 1 | 2 | 3 | 4 | 5 | 6

<encdg> ::= ASCII | BINARY

<p/w> ::= 820 | 1024

<x increment> ::= <NR3> = time/div / points/div

<point #> ::= <NR1>

<y multiplier> ::= <NR3> = vertical scale factor

<y zero> ::= <NR3> = -(VZR \* vertical scale factor)

<wfmid> ::= W <wfm #> | W <wfm #> ,

where I indicates interpolated data.

## Waveform Curve

1. **Curve Commands.** These transmit waveform point data to the 7D20. The CURVE command takes two different forms:

**CURVE:** <ASCII curve> | <binary curve>

<ASCII curve> ::= <ASCII point>, <ASCII point>...

<binary curve> ::= %<binary count><binary point>...

<checksum> (<checksum> is defined below.)

Waveform memory extends from -5.12 divisions to +5.08 divisions. When transmitting an ASCII curve, data is sent in divisions. When transmitting a binary curve, data is sent as point numbers with  $00000000_2$ ,  $10000000_2$ ,  $11111111_2$  corresponding to -5.12 divisions, 0 divisions (center screen), and +5.08 divisions, respectively. A short curve is filled out with a reiteration of the last point. Note that an ASCII curve must have at least two points. If only one point is sent, a syntax error code will be generated.

2. **Curve Queries.** These queries transmit waveform data from the 7D20 to the GPIB controller. The DATA command specifies whether ASCII or binary format is to be used.

### CURVE?

Responds with the current source waveform's curve:

if ASCII:

**CURVE** <ASCII point>, <ASCII point>...

if binary:

**CURVE** %<binary count><binary point>...<checksum>

where:

<ASCII point> ::= <NR3>

<binary count> ::= two bytes representing number of data points +1 (checksum)

<binary point> ::= 8-bit byte (00 | 01 | ... | FE | FF)

<checksum> ::= two's complement of the modulo 256 sum of the preceding binary data bytes and the binary count, but not the "%" preceding the binary byte count.

<wfmid> ::= W <wfm #> | W <wfm #>, (I indicates interpolated data.)

### Waveform Preamble and Curve

The complete waveform may be transmitted from the 7D20 using the query "WAVFRM?". For convenience, the response includes both preamble and curve.

**WAVFRM?**      Responds with  
                   <preamble><curve>  
                   <preamble> ::= response to **WFMPRE?**  
                   <curve> ::= response to **CURVE?**

**Waveform Data Commands.** To transmit waveforms from the 7D20, first specify the waveform data source. This is done through the command: **DATA MEMORY:<wfm #>**, which sets the memory source for reception or transmission. The <wfm #> options are memory sources 1 through 6. The second command, "**DATA ENCDG: <ASCII | BINARY>**", chooses between the ASCII format or the binary format. When receiving waveforms into the 7D20, the same kinds of operations apply as when transmitting from the instrument.

The third command, "**DATA INTERPOLATE: <ON | OFF>**", sets 820 to 1024-point interpolation mode for all extended real-time waveforms transmitted from the 7D20. An 820-point waveform format is used for the extended real-time waveforms that are digitized when the 7D20 TIME/DIV is 200  $\mu$ s/div to 2 s/div. For the most convenience with some packaged waveform processing programs, the 7D20 will interpolate the waveform data into a 1024-point data length if desired. The algorithm for the interpolation process, which maintains the value of the trigger point without alteration, is as follows:

$$\begin{aligned} Q_0 &= 3/5 P_0 + 2/5 P_1 \\ Q_1 &= 4/5 P_1 + 1/5 P_2 \\ Q_2 &= P_2 \\ Q_3 &= 1/5 P_2 + 4/5 P_3 \\ Q_4 &= 2/5 P_3 + 3/5 P_4 \\ Q_5 &= 3/5 P_4 + 2/5 P_5 \\ Q_6 &= 4/5 P_5 + 1/5 P_6 \\ Q_7 &= P_6 \\ Q_8 &= 1/5 P_6 + 4/5 P_7 \\ Q_{1022} &= P_{818} \\ Q_{1023} &= 1/5 P_{818} + 4/5 P_{819} \end{aligned}$$

where:       $Q_n$  = Value of point n of the 1024-point waveform  
                    $P_m$  = Value of point m of the 820-point waveform  
                   n = 0 to 1023  
                   m = 0 to 819

Note that Q2 and Q7 of the interpolated waveform are not fabricated. This is true for every fifth point on the waveform.

## SERVICE REQUESTS

Service requests allow the 7D20 to report operating status in detail to the controller, which encourages a synergistic relationship to exist among the system components. Additionally, the user will find service requests a very powerful aid in debugging programs and in high-order troubleshooting.

SRQ responses fall into four major classes. The first reports on the health of the 7D20, by means of such status bytes as Power On, Fatal Error, and Internal Error. The second reports on Execution Status, using status bytes such as No Status to Report and Operation Complete. The third class consists of User Requests, including RQS and RQS#. The fourth reports on errors and warnings, using Execution Warning and Execution Error as a means of conveying status.

Responses to the controller divide into two discrete levels. The first, the status byte, is used as a general response. For example, Power On, Internal Error, Command Error, and Execution Error are all status bytes, and are meant to indicate the general class of condition that initiated the SRQ.

The second response level is the event code, which in the 7D20 is used to subclassify SRQ responses at a level that is much more specific than that of the status byte. For example, event codes 108 through 162 (see Table 4-6) only deal with command errors, and each of the codes specifies a certain kind of command error. Event code 108, for example, states that a checksum error has occurred; event code 109 points out a byte-count error; and so forth.

### USING SRQ

The SRQ function is device-initiated, in that the 7D20 asserts SRQ when an event occurs that is included in the event code list, and the corresponding service request mask is enabled. (See the discussion on Service Request Masks on the following pages.) The controller responds to the SRQ by polling all devices on the bus, which clears the SRQ, and receives the status byte that indicates the class of event, such as Command Error, etc. The status bytes are listed and defined in Table 4-4.

Refer to Figure 4-14, which is a block diagram of the Status Byte and Event Code Buffers. It depicts two paths, one for the status bytes, the other for the event codes. Each event code corresponds with a particular status byte.

When the 7D20 senses a condition that merits the operator's attention, a status byte is sent to the output interface and SRQ is asserted to alert the controller that its attention is required. At some time later, the controller interrogates each instrument on the GPIB by performing a serial poll. When it interrogates the 7D20, this instrument responds with the current status byte. If the controller then follows with an event query, the 7D20 will respond with the current event code (which corresponds with the current status byte). Figure 4-15 is a flow chart that illustrates the path that is followed when an event code is generated.

**TABLE 4-4**  
**7D20 Status Bytes**

Title	Binary	Decimal		Priority
		(not busy)	(busy)	
(Device Dependent: Fatal Error	1R1X 0011	227	243	1
(System:)				
Power On	010X 0001	65	81	1
Command Error	0R1X 0001	97	113	2
User Request	010X 0011	67	83	2
Execution Error	0R1X 0010	98	114	2
Internal Error	0R1X 0011	99	115	2
Execution Warning	0R1X 0101	101	117	2
Internal Warning	0R1X 0110	102	118	2
Operation Complete	0ROX 0010	66	82	2
No Status To Report	000X 0000	0	16	3

R is asserted with the GPIB command: RQS:ON. If the 7D20 is in the RQS:OFF mode and polled, the status byte is sent without DIO7 asserted. X is the busy bit, and is asserted only when the 7D20 is in AVE, AVEN, ENV, ENVN, or HOLDNEXT at the time of the SRQ. The first column of decimal values is without the busy bit set and the last column is with the busy bit set. Both columns assume RQS:ON.

It should be noted that status bytes are not stacked; that is, only the current status byte (in the interface) and the status byte of highest priority (status byte pending) are saved. All status bytes of lower priority than the one pending are discarded. Also, the status byte in the pending register is discarded and replaced if one of higher priority appears. Thus, with the exception of the status byte stored in the interface (status byte current), the highest priority status byte is stored in the pending register.

One exception to the above is the Fatal Error status byte, which acts as a system safeguard. If a Fatal Error occurs (which only happens if the 7D20 fails), this status byte goes directly to the interface, and the 7D20 will refuse to respond further to any bus commands, except serial poll. Thus, the fatal error status byte never has a corresponding event code, for none could be read. The reason for the shut down is to prevent transmission of erroneous bus commands that might adversely affect other instruments on the GPIB.

Unlike status bytes, as many as 40 event codes are saved for later examination. Referring again to Figure 4-14, note that except for the "LIFO" (last-in, first-out) buffer, the structure of event-code storage is similar to that of the status bytes. New event codes that occur are sorted to determine the priority of each; if one is of higher priority than that stored in the Event Pending buffer, the code in the buffer is

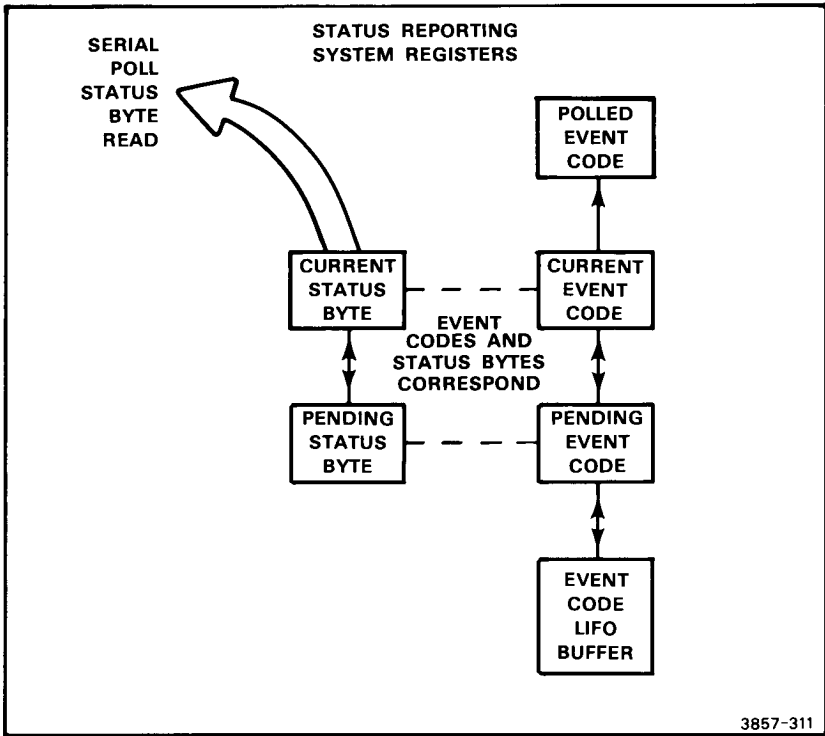


Figure 4-14. 7D20 status reporting block diagram.

replaced by the higher priority code, and the lower one moves to the LIFO buffer. New event codes of lower priority are placed directly into the LIFO buffer, and those that are oldest are discarded by overflow if the buffer is full.

Following a serial poll of the 7D20, the controller will usually send an event query to the instrument. If such occurs, the current event code is placed on the bus, and the event code pending becomes the current event code. See Figure 4-16, which illustrates the path of events following a serial poll.

#### NOTE

*If the 7D20 is polled twice with no intervening event query, the event code pending (from the first status byte) is discarded and lost. This occurs so that the relationship between the status byte and the corresponding event code will remain consistent.*

When RQS:OFF, status bytes are loaded without causing an SRQ to occur with the exception of Fatal Error, which interrupts 7D20 operation regardless of its state.

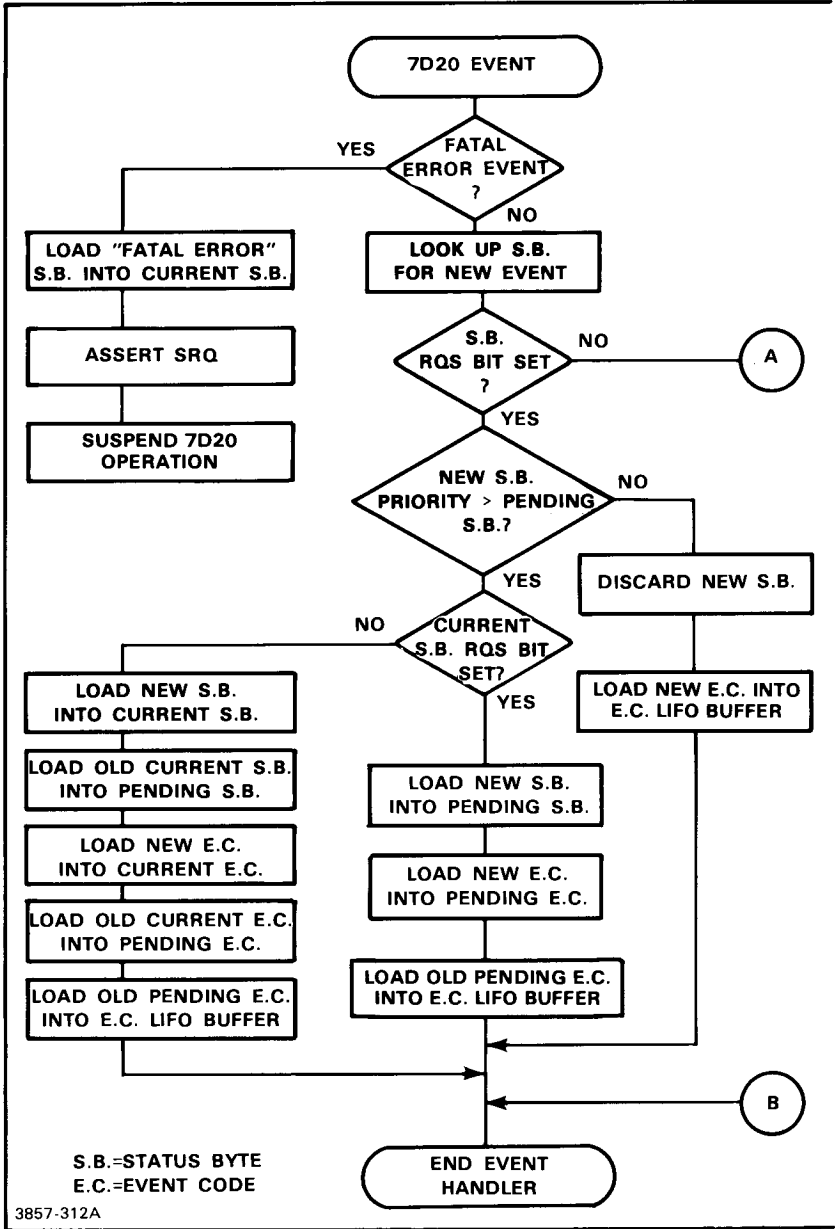
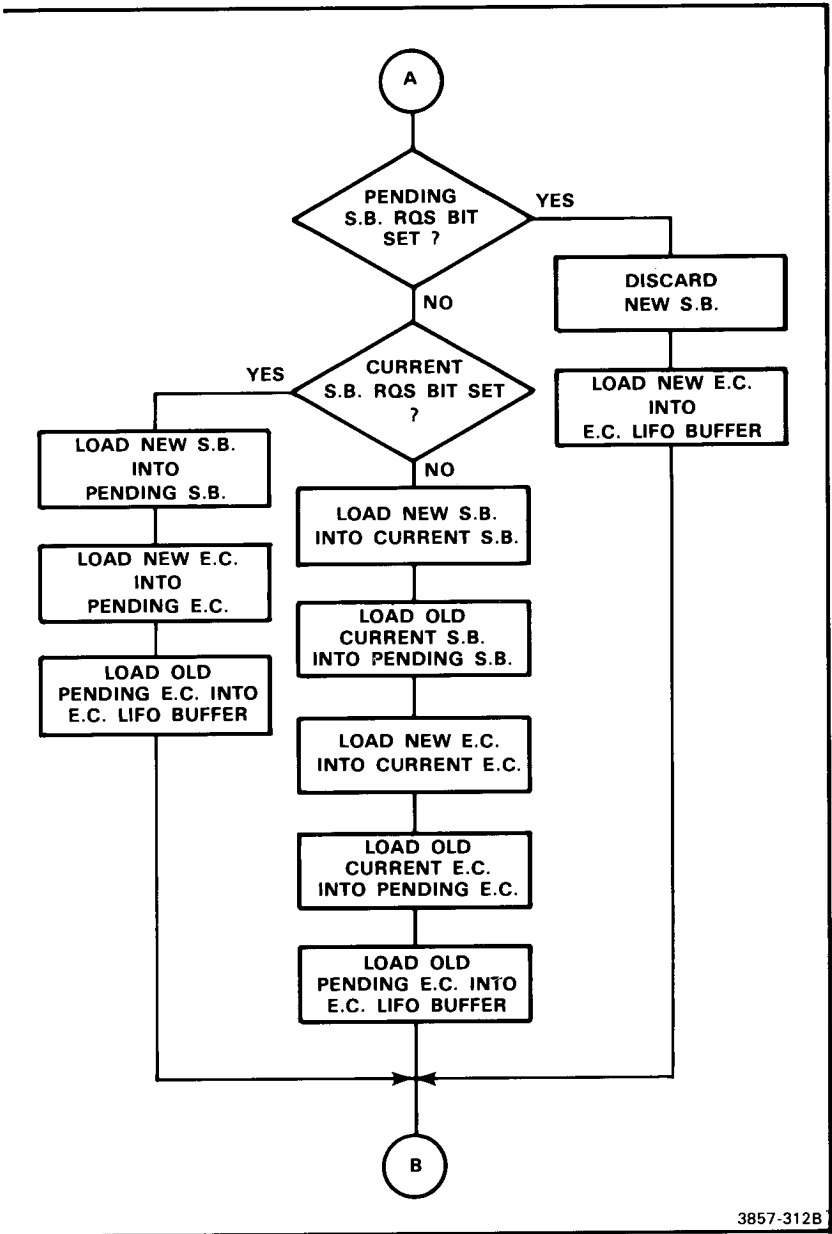


Figure 4-15. Event handler flow chart.



3857-312B

Figure 4-15 (cont). Event handler flow chart.



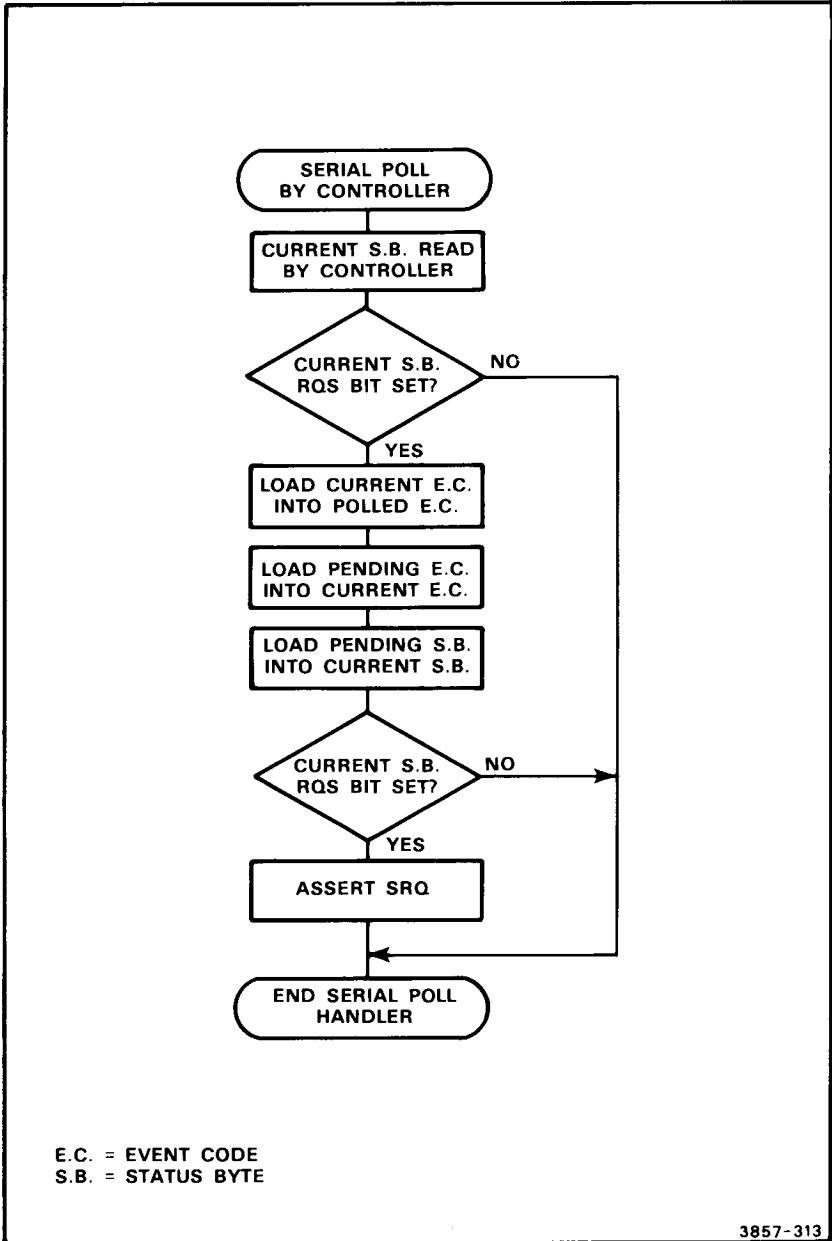


Figure 4-16. Serial poll handler flow chart.

Event codes follow very nearly the same path as when RQS:ON, except that the prioritizing process does not occur. That is, as event codes occur, they are loaded into the LIFO buffer and are discarded by overflow when the buffer is full.

### Service Request Masks

Upon occasion, the user may set up operations that cause service requests to be generated regularly, to the point where they interfere with efficient system functioning. The 7D20 allows masking all or part of the service request functions to prevent service request interference with legitimate programs or operations. Table 4-5 lists the commands that enable and disable the service request functions.

**TABLE 4-5**  
**Service Request Masks**

Commands	Description
<b>RQS ON OFF</b>	enables or disables all service request functions except power on and fatal error.
<b>CER ON OFF</b>	enables or disables command-error service request.
<b>EXR ON OFF</b>	enables or disables execution-error service request.
<b>INR ON OFF</b>	enables or disables internal-error service request.
<b>EXW ON OFF</b>	enables or disables execution-warning service request.
<b>OPC ON OFF</b>	enables or disables operation-complete service request.
<b>USER ON OFF</b>	enables or disables front-panel generated RQS key service request.
<b>PID ON OFF</b>	enables or disables probe-identify service request.
Queries	Description
<b>RQS?</b>	responds with the RQS service request status: <b>RQS&lt;onoff&gt;</b>
<b>CER?</b>	responds with the CER service request status: <b>CER&lt;onoff&gt;</b>
<b>EXR?</b>	responds with the EXR service request status: <b>EXR&lt;onoff&gt;</b>
<b>INR?</b>	responds with the INR service request status: <b>INR&lt;onoff&gt;</b>
<b>EXW?</b>	responds with the EXW service request status: <b>EXW&lt;onoff&gt;</b>
<b>OPC?</b>	responds with the OPC service request status: <b>OPC&lt;onoff&gt;</b>
<b>USER?</b>	responds with the USER service request status: <b>USER&lt;onoff&gt;</b>

TABLE CONTINUED

**TABLE 4-5 (CONT)**  
**Service Request Masks**

Queries	Description
<b>PID?</b>	responds with the PID service request status: <b>PID&lt;onoff&gt;</b>
<b>SRQ?</b>	responds with all service request settings: <b>RQS&lt;onoff&gt;;CER&lt;onoff&gt;;EXR&lt;onoff&gt;;INR&lt;onoff&gt;; EXW&lt;onoff&gt;;OPC&lt;onoff&gt;;USER&lt;onoff&gt;;PID&lt;onoff&gt; &lt;onoff&gt;::=ON OFF</b>

## EVENT QUERIES

Following the reception of a status byte, the controller should send an event query to specify the event more precisely. This is done by use of the query forms:

**EVQTY?**  
**ALLEVENT?**  
**EVENT?**

If an SRQ is pending and an event query is sent before a serial poll, the 7D20 will return an SRQ PENDING event code to remind the controller that a service request is asserted and that a serial poll should be initiated.

The query "**EVQTY?**" interrogates the 7D20 to find how many event codes are pending. It should be noted that the 7D20 stores events as each occurs, even if service requests are disabled. The query "**EVQTY?**" should be the first interrogation, to determine the event count. The response is:

**EVQTY <NR1>**

if events are pending, and:

**EVQTY 0**

if none are pending.

The "**ALLEVENT?**" query causes the 7D20 to respond by reporting all pending event codes as follows:

**EVENT <NR1>,<NR1>...**

or if no event codes are pending:

**EVENT 0**

All event codes are cleared from the buffer when this query is made. Figure 4-17 is a flow chart of the "**ALLEVENT?**" sequence.

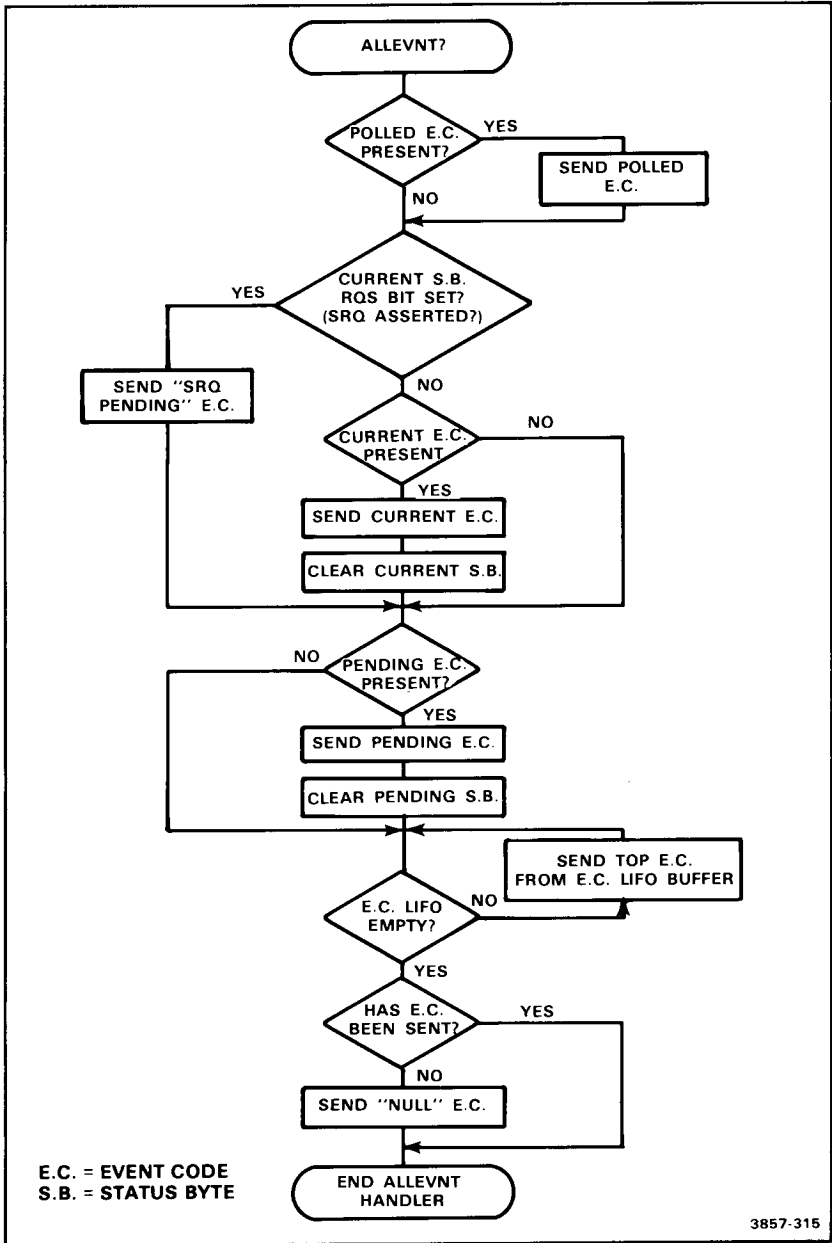


Figure 4-17. All-event query flow chart.

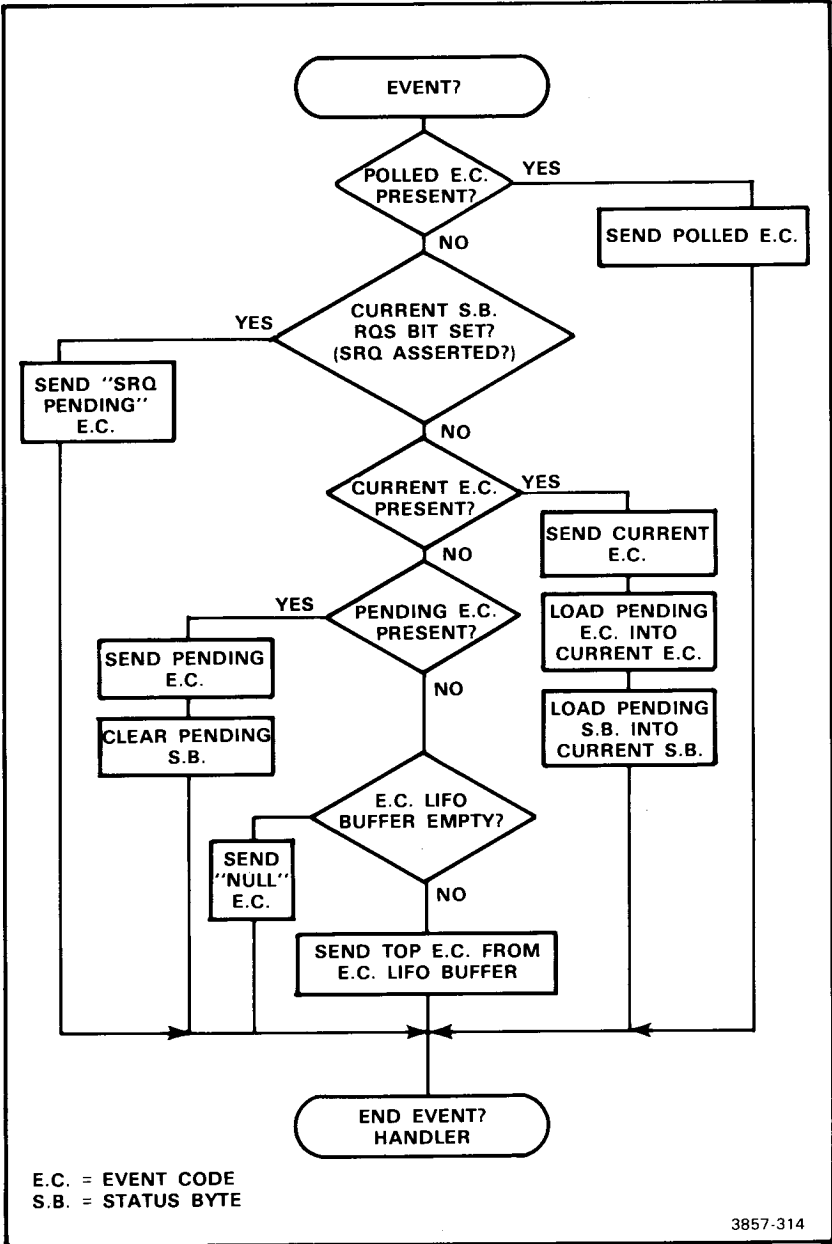


Figure 4-18. Event query flow chart.

The query "EVENT?" causes the 7D20 to report a single event code.

**EVENT <NR1>**

or:

**EVENT 0**

if none are pending. Figure 4-18 is a flow chart of the event query (EVENT?) sequence.

### **Event Classification**

Events that are reported by the 7D20 fall into the following classes:

- Command Error:** A command error results when a message cannot be parsed or lexically analyzed.
- Execution Error:** An execution error exists when a message is parsed but cannot be executed.
- Internal Error:** This error results from a malfunction or fault condition within the 7D20 (self test error).
- System Event:** This class includes all events that are not errors or warnings.
- Execution Warning:** The command received is being executed, but a potential problem may exist.

**Event Codes**

Table 4-6 lists the event codes from the 7D20, divided into the classes listed just above.

**TABLE 4-6**  
**Event Codes**

Code & Mask	Description
Command Errors (Mask:CER) 108 109 151 152 153 154 155 156 157 158 159 160 161 162	Checksum error in curve transfers Illegal byte count value on binary transfers Symbol or number too long Invalid input or control character input Invalid special character after escape Invalid number input Invalid string input Symbol not found Syntax error Invalid EOI Invalid delimiter Expression too complex Excessive binary curve points Excessive ASCII curve points
Execution Errors (Mask:EXR) 203 250 251 252 253 254 255 256 257 258 259	I/O buffers full, output dumped Not in HOLD mode Illegal waveform number Illegal settings memory Illegal cursor number Settings recall error Display reference error Turning off cursor waveform display error Illegal data memory Roll mode, AVE, ENV, negative trigger position error Waveform preamble illegal NR.PT
Internal Errors (Mask:INR) 331 332 393 394	Selftest failure module #1 <sup>a</sup> Selftest failure module #2 <sup>a</sup> Selftest failure module #63 <sup>a</sup> Selftest completed and failed

<sup>a</sup>The selftest base value is 330. The module number of the failed circuit is added to the base value (e.g., module #41 would have an event code of 330 + 41 = 371). Possible event codes are 331 through 393.

**TABLE 4-6 (CONT)**  
 **Event Codes**

Code & Mask	Description
<b>System Events</b>	
401 (Mask:NONE)	7D20 Power On
403 (Mask:USER)	User Request (RQS key)
450 (Mask:OPC)	Hold after AVEN, ENVN, or HOLD NEXT
451 (Mask:USER)	RQS 1
452 (Mask:USER)	RQS 2
453 (Mask:USER)	RQS 3
454 (Mask:USER)	RQS 4
455 (Mask:USER)	RQS 5
456 (Mask:USER)	RQS 6
457 (Mask:PID)	CH 1 probe identify
458 (Mask:PID)	CH 2 probe identify
459 (Mask:NONE)	SRQ pending
460 (Mask:OPC)	Selftest operation complete, passed
<b>Execution Warning (Mask:EXW)</b>	
550	Volts/Div out of range
551	Vertical position out of range
552	Trigger level out of range
553	Trigger position out of range
554	Horizontal Time/Div out of range
555	VXPD command out of range
556	CSW position out of range
557	Set N value out of range
558	Cursor position out of range
559	ASCII point out of range
560	WFMPRE X increment out of range
561	WFMPRE point offset out of range
562	WFMPRE Y multiplier out of range
563	WFMPRE Y zero out of range

**USER-REQUEST FUNCTIONS**

Ordinarily, communications over the GPIB occur from user to controller, then over the GPIB to devices like the 7D20. Likewise, a talker-listener like the 7D20 can communicate implicitly with the user by way of the GPIB and through the controller. Primitive or simple controllers may lack the means to allow the user to communicate changes in instructions, save by altering the software or firmware. Through the use of the front-panel keypad (MEMORY DISPLAY section) and the RQS # key, the user can communicate up to six different messages to the controller. Without using the RQS # key, two more user request functions are available



through use of the P6053B probe IDENT button. Thus, the probe can remotely initiate 7D20 operations via the controller. In addition, the feature permits probe identification from the probe tip to verify that the proper probe has been connected to the device under test.

Use of the feature is as follows:

1. The user presses the function *f* key to shift the front-panel keys to the orange-colored functions, then presses RQS #. The prompt field will respond by displaying "RQS", followed by a blinking "#".
2. The user should now press the appropriate key of the keypad, 1 through 6. This generates an SRQ having a unique event code. The prompt field displays "RQS", followed by the selected number. In turn, the unique event code generated by this action may be used to initiate a preplanned activity on the part of the controller. Note that the user can also use only the RQS key, which calls forth event code 403. If the GPIB mode is OFFLINE, RQS OFF, or USER OFF, the prompt field will display "RQS OFF" and no action will occur.
3. If the Identify button on either probe is pressed, an SRQ is generated and event code 457 or 458 is produced, depending on the probe selected. If the GPIB mode is OFFLINE, RQS OFF, or PID OFF, the prompt field will display "RQS OFF" and no action will occur.

The user-request functions allow numerous applications that would otherwise not be available. Along with the text feature, this allows the user to produce special menus, for example. It is possible to initiate any activity that is possible with the controller and accompanying system that can be encoded. For example, a menu could be created and transferred to the 7D20 with a TEXT command by the user through the front-panel keypad. The keypad is then used to make the menu selections, as prompted by the menu. An example follows:

#### WFM PARAMETERS

- 1 MAX
- 2 MIN
- 3 P-P
- 4 MEAN
- 5 RMS
- 6 PULSE PARAM MENU

PRESS '*f*' RQS # TO SELECT

### COMMAND LIST

Along with the functional classes that come naturally with an instrument such as the 7D20 (i.e., vertical, horizontal, display, etc.), the 7D20 command set divides into three classes: Menu-related, Front-panel related, and Bus-unique. A very few of the commands relate to more than one class, and are so noted in the following descriptions.

The 7D20 command set is listed in Table 4-7. Each major segment of the command list is headed by a graphic representation of the related area for reference by the reader. For example, near the beginning of the command list, the commands and queries for Channel 1 are listed. At the head of the Channel 1 listing is an illustration of the CH 1 area of the front panel of the 7D20, to which those commands are closely related. Likewise, the Stored Settings portion of the Command List is preceded by an illustration of the Master Menu, which relates directly to the Stored Settings commands. Following the Stored Settings commands is the Device Trigger commands list, which is preceded by the statement that the following set of commands is bus-unique and has no direct counterpart in either the front panel or any of the menus.

Note also that, following most of the major groups in this table, there is a group of explanations for the bracketed items. Following the Time/div group, for example, is:

<time/div>::=<NR3>, which means:  
 Time per division is defined as a  
 number in exponential notation format.

TABLE 4-7  
 7D20 Command Set

CHANNEL 1 GROUP

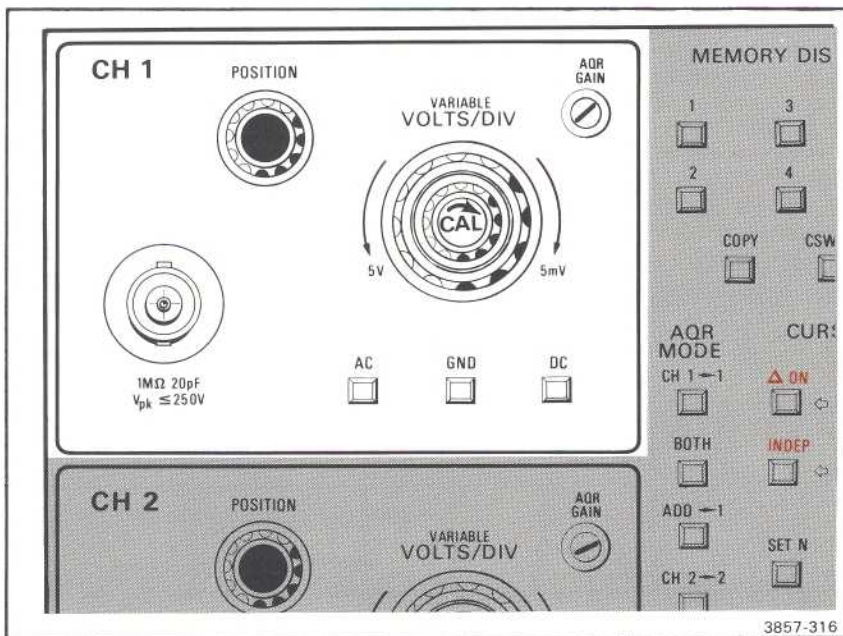


TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description	
<b>CH1 VOLTS:</b> <volts/div>	Sets the channel 1 volts/div to the valid setting nearest <volts/div>.	
	<volts/div>	volts/div
	<3.5E-3	5 mV <sup>a</sup>
	3.5E-3 to <7.5E-3	5 mV
	7.5E-3 to <15E-3	10 mV
	15E-3 to <35E-3	20 mV
	35E-3 to <75E-3	50 mV
	75E-3 to <150E-3	100 mV
	150E-3 to <350E-3	200 mV
	350E-3 to <750E-3	500 mV
	750E-3 to <1.5	1 V
	1.5 to <3.5	2 V
	3.5 to <7.5	5 V
	>7.5	5 V <sup>a</sup>
<b>CH1? VOLTS</b>	Responds with the channel 1 volts/div setting:  <b>CH1 VOLTS:&lt;volts/div&gt;</b>	
<b>CH1 POSITION:</b> <position>	Sets the channel 1 position to the valid setting nearest <position>.	
	<position>	position (div)
	<-10.25	-10.24 <sup>a</sup>
	-10.24 to +10.22	<position>
	>+10.23	+10.22 <sup>a</sup>
<b>CH1? POSITION</b>	Responds with the channel 1 position setting:  <b>CH1 POSITION:&lt;position&gt;</b>	
<b>CH1 COUPLING:</b> <b>AC   GND   DC</b>	Sets the channel 1 coupling to either AC, GND or DC.	
<b>CH1? COUPLING</b>	Responds with channel 1's coupling setting:  <b>CH1 COUPLING:&lt;coupling&gt;</b>	

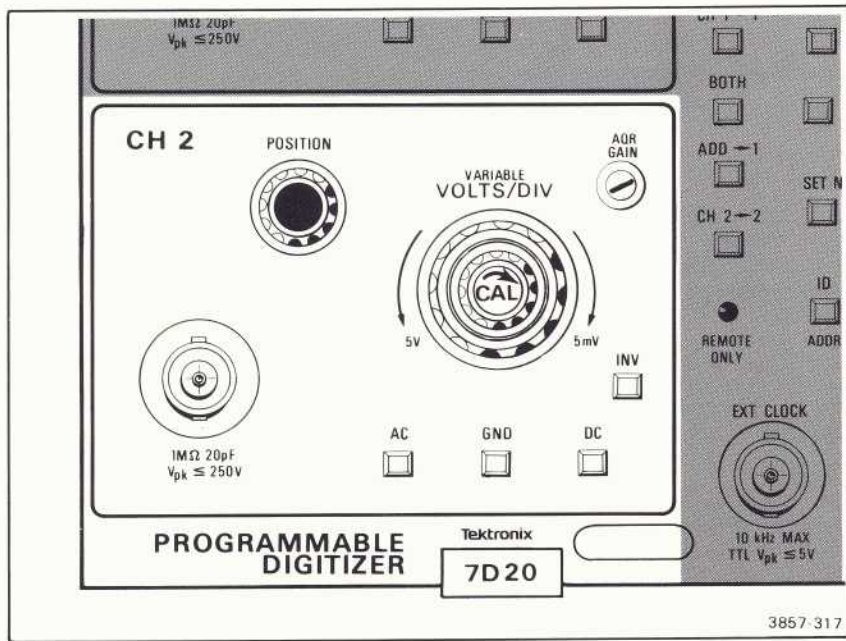
<sup>a</sup>execution error warning

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>CH1 VARIABLE:</b> <b>ON   OFF</b>	Enables/Disables CH 1 variable gain control.
<b>CH1? VARIABLE</b>	Responds with the channel 1 variable gain control status:  <b>CH1 VARIABLE:&lt;onoff&gt;</b>
<b>CH1 PROBE:</b> <prbcode>	Command is ignored by the 7D20.
<b>CH1? PROBE</b>	Responds with probe attenuation encoding for channel 1:  <b>CH1 PROBE:&lt;prbcode&gt;</b>
<b>CH1?</b>	Responds with all channel 1 settings:  <b>CH1 VOLTS:&lt;volts/div&gt;, POSITION:&lt;position&gt;, COUPLING:&lt;coupling&gt; ,VARIABLE:&lt;onoff&gt; , PROBE:&lt;prbcode&gt;</b>  <volts/div> ::= <NR3> <position> ::= <NR2> <coupling> ::= AC   GND   DC <onoff> ::= ON   OFF <prbcode> ::= 1   10   100

TABLE 4-7 (CONT)  
 7D20 Command Set

CHANNEL 2 GROUP



Command/Query	Description	
<b>CH2 VOLTS:</b> <volts/div>	Sets the channel 2 volts/div to the valid setting nearest <volts/div>.	
	<volts/div>	volts/div
	<3.5E-3	5 mV <sup>a</sup>
	3.5E-3 to <7.5E-3	5 mV
	7.5E-3 to <15E-3	10 mV
	15E-3 to <35E-3	30 mV
	35E-3 to <75E-3	50 mV
	75E-3 to <150E-3	100 mV
	150E-3 to <350E-3	200 mV
	350E-3 to <750E-3	500 mV
	750E-3 to <1.5	1 V
	1.5 to <3.5	2 V
	3.5 to <7.5	5 V
	≥7.5	5 V <sup>a</sup>

<sup>a</sup> execution error warning



TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description	
<b>CH2? VOLTS</b>	Responds with the channel 2 volts/div setting:  <b>CH2 VOLTS:&lt;volts/div&gt;</b>	
<b>CH2 POSITION:</b> <position>	Sets the channel 2 vertical position to the valid setting nearest <position>.	
	<position>	position (div)
	<-10.25 -10.24 to +10.22 >+10.23	-10.24 <sup>a</sup> <position> +10.22 <sup>a</sup>
<b>CH2? POSITION</b>	Responds with the channel 2 position setting:  <b>CH2 POSITION:&lt;position&gt;</b>	
<b>CH2 COUPLING:</b> <b>AC   GND   DC</b>	Sets the channel 2 coupling to either AC, GND or DC.	
<b>CH2? COUPLING:</b>	Responds with channel 2 coupling setting:  <b>CH2 COUPLING:&lt;coupling&gt;</b>	
<b>CH2 VARIABLE:</b> <b>ON   OFF</b>	Enables/Disables CH 2 variable gain control.	
<b>CH2? VARIABLE</b>	Responds with the channel 2 variable gain control status:  <b>CH2 VARIABLE:&lt;onoff&gt;</b>	
<b>CH2 INVERT:</b> <b>ON   OFF</b>	Turns inversion of the channel 2 signal either ON or OFF	
<b>CH2? INVERT</b>	Responds with the channel 2 inversion setting:  <b>CH2 INVERT:&lt;onoff&gt;</b>	

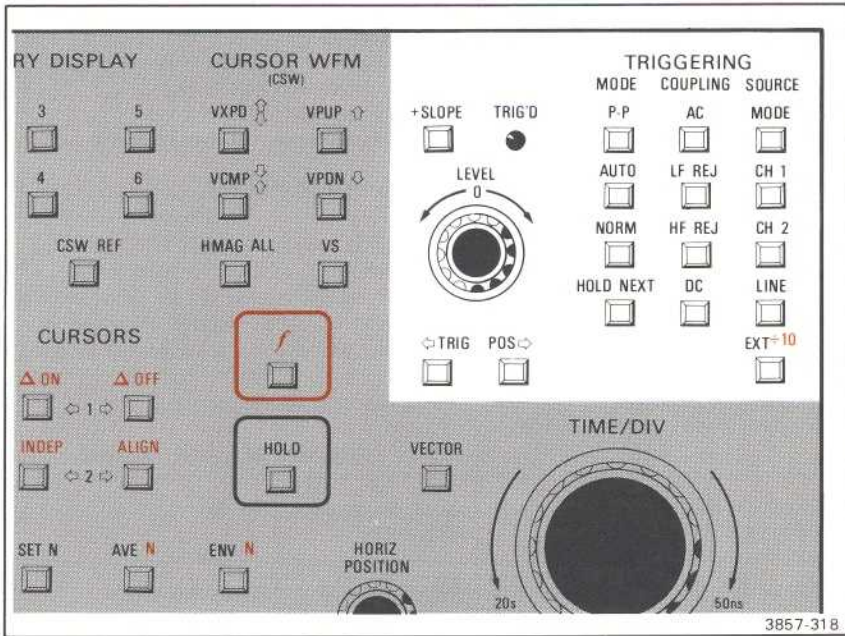
<sup>a</sup>execution error warning

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>CH2? PROBE</b>	Responds with probe encoding for channel 2:  <b>CH2 PROBE:&lt;prbcode&gt;</b>
<b>CH2?</b>	Responds with all channel 2 settings:  <b>CH2 VOLTS:&lt;volts/div&gt;,POSITION &lt;position&gt;, COUPLING:&lt;coupling&gt;,VARIABLE:&lt;onoff&gt;, INVERT:&lt;onoff&gt;:PROBE:&lt;prbcode&gt;</b>  <volts/div> ::= <NR3> <position> ::= <NR2> <coupling> ::= AC   GND   DC <onoff> ::= ON   OFF <prbcode> ::= 1   10   100

TABLE 4-7 (CONT)  
 7D20 Command Set

TRIGGERING GROUP



Command/Query	Description
<b>TRIGGER MODE:</b> P-P   AUTO   NORMAL	Sets trigger mode to either peak to peak, AUTO, or NORMAL.
<b>TRIGGER? MODE</b>	Responds with the triggering mode setting:  <b>TRIGGER MODE:&lt;mode&gt;</b>
<b>TRIGGER HOLDNEXT:</b> ON   OFF	Turns HOLDNEXT either ON or OFF. Retains the same trigger level as previously set by P-P, AUTO, or NORM. Refer to HOLDNEXT in Front Panel controls section for more details. Upon entering HOLD, an operation complete SRQ is generated if OPC:ON.
<b>TRIGGER? HOLDNEXT</b>	Responds with the hold next setting:  <b>TRIGGER HOLDNEXT:&lt;onoff&gt;</b>



**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description								
<b>TRIGGER COUPLING:</b> <b>AC</b>   <b>ACLFREJ</b>   <b>ACHFREJ</b>   <b>DCHFREJ</b>   <b>DC</b>	Sets trigger coupling to either AC, AC low-frequency reject, AC high-frequency reject DC high-frequency reject, or DC.								
<b>TRIGGER? COUPLING</b>	Responds with the triggering coupling:  <b>TRIGGER COUPLING:</b> <coupling>								
<b>TRIGGER SOURCE:</b> <b>MODE</b>   <b>CH1</b>   <b>CH2</b>   <b>LINE</b>   <b>EXT</b>   <b>EXT/10</b>	Sets trigger source to either MODE, CH1, CH2, LINE, EXT, or EXT/10. If MODE is selected, trigger source is CH1 when acquire mode is CH1, ADD, or BOTH, and channel two when Acquire mode is channel two.								
<b>TRIGGER? SOURCE</b>	Responds with the triggering source:  <b>TRIGGER SOURCE:</b> <source>								
<b>TRIGGER SLOPE:</b> <b>PLUS</b>   <b>MINUS</b>	Sets trigger slope to either PLUS or or MINUS.								
<b>TRIGGER? SLOPE</b>	Responds with the triggering slope:  <b>TRIGGER SLOPE:</b> <slope>								
<b>TRIGGER LEVEL:</b> <level>	Sets trigger level to the valid setting nearest <level>.								
	<table border="1"> <thead> <tr> <th>&lt;level&gt;</th> <th>trigger level</th> </tr> </thead> <tbody> <tr> <td>&lt;-6.425</td> <td>-6.4<sup>a</sup></td> </tr> <tr> <td>-6.4 to +6.35</td> <td>&lt;level&gt;</td> </tr> <tr> <td>&gt;+6.375</td> <td>+6.35<sup>a</sup></td> </tr> </tbody> </table>	<level>	trigger level	<-6.425	-6.4 <sup>a</sup>	-6.4 to +6.35	<level>	>+6.375	+6.35 <sup>a</sup>
<level>	trigger level								
<-6.425	-6.4 <sup>a</sup>								
-6.4 to +6.35	<level>								
>+6.375	+6.35 <sup>a</sup>								
	In p-p, +6.35 and -6.4 are scaled to correlate the maximum and minimum values of the waveform.								
<b>TRIGGER? LEVEL</b>	Responds with the triggering level setting:  <b>TRIGGER LEVEL:</b> <level>								

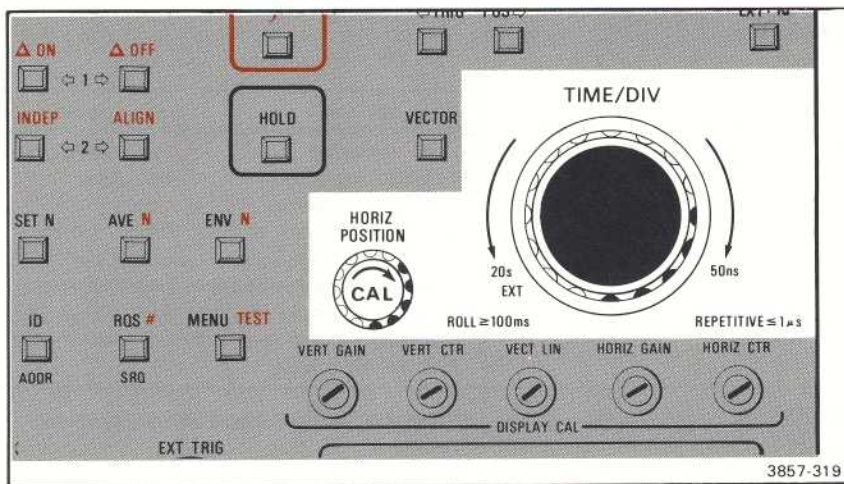
<sup>a</sup> execution error warning

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description		
<b>TRIGGER POSITION:</b> <division>	Sets position for displaying trigger point to the valid setting nearest <division>.		
	<table border="1"> <tr> <td data-bbox="485 362 740 402">&lt;division&gt;</td> <td data-bbox="740 362 962 402">trigger point</td> </tr> </table>	<division>	trigger point
	<division>	trigger point	
	<table border="1"> <tr> <td data-bbox="485 410 740 495">                     &lt;-1500.5 -1500.5 to &lt;10.5 10.5 or greater                 </td> <td data-bbox="740 410 962 495">                     div -1500<sup>a</sup> &lt;division&gt; div 10<sup>a</sup> </td> </tr> </table>	<-1500.5 -1500.5 to <10.5 10.5 or greater	div -1500 <sup>a</sup> <division> div 10 <sup>a</sup>
<-1500.5 -1500.5 to <10.5 10.5 or greater	div -1500 <sup>a</sup> <division> div 10 <sup>a</sup>		
Restriction: Maximum position is 10 <sup>th</sup> graticule division; minimum is 1500 divisions to left of graticule line zero. In ROLL or EXT digitizing modes, the minimum setting with ENV or AVE is 0.			
<b>TRIGGER? POSITION</b>	Responds with the trigger display position setting:  <b>TRIGGER POSITION:</b> <position>		
<b>TRIGGER?</b>	Responds with all triggering settings:  <b>TRIGGER MODE:</b> <mode>, <b>HOLDNEXT:</b> <onoff>, <b>COUPLING:</b> <coupling>, <b>SOURCE:</b> <source>, <b>SLOPE:</b> <slope>, <b>LEVEL:</b> <level>, <b>POSITION:</b> <position> <mode> ::= P-P   AUTO   NORM <coupling> ::= AC   ACLFREJ   ACHFREJ   DCHFREJ   DC <source> ::= MODE   CH1   CH2   LINE   EXT   EXT/10 <slope> ::= PLUS   MINUS <level> ::= <NR3> <position> ::= <NR1>		

TABLE 4-7 (CONT)  
7D20 Command Set

TIME/DIV GROUP



Command/Query	Display	
<b>HORIZONTAL TIME:</b> <time/div>	Sets time/div to the valid setting nearest <time/div>.	
	<time/div>	time/div
	<35E-9	50 nS <sup>a</sup>
	35E-9 to <75E-9	50 nS
	75E-9 to <150E-9	100 nS
	150E-9 to <350E-9	200 nS
	350E-9 to <750E-9	500 nS
	750E-9 to <1.5E-6	1 μS
	1.5E-6 to <3.5E-6	2 μS
	3.5E-6 to <7.5E-6	5 μS
	7.5E-6 to <15E-6	10 μS
	15E-6 to <35E-6	20 μS
	35E-6 to <75E-6	50 μS
	75E-6 to <150E-6	100 μS
	150E-6 to <350E-6	200 μS
	350E-6 to <750E-6	500 μS
	750E-6 to <1.5E-3	1 mS
	1.5E-3 to <3.5E-3	2 mS
	3.5E-3 to <7.5E-3	5 mS

<sup>a</sup> execution error warning

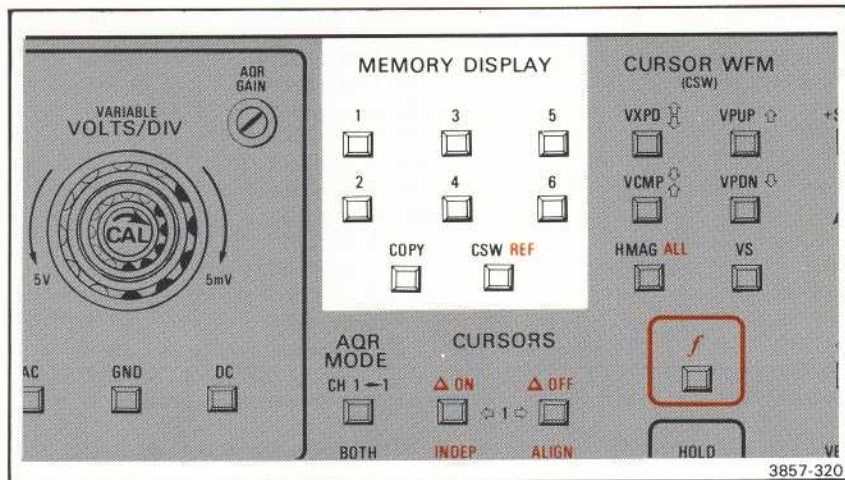
TABLE 4-7 (CONT)  
 7D20 Command Set

Command/Query	Description	
	<time/div>	time/div
	7.5E-3 to <15E-3	10 mS
	15E-3 to <35E-3	20 mS
	35E-3 to <75E-3	50 mS
	75E-3 to <150E-3	100 mS
	150E-3 to <350E-3	200 mS
	350E-3 to <750E-3	500 mS
	750E-3 to <1.5	1 S
	1.5 to <3.5	2 S
	3.5 to <7.5	5 S
	7.5 to <15	10 S
	15 to <35	20 S
	35 or greater	20 S <sup>a</sup>
<b>HORIZONTAL? TIME</b>	Responds to the time/div setting:  <b>HORIZONTAL TIME:&lt;time/div&gt;</b>	
<b>HORIZONTAL POSITION: ON   OFF</b>	Turns enable for horizontal position knob ON or OFF. Sets horizontal position to current knob setting or calibrated setting.	
<b>HORIZONTAL? POSITION</b>	Responds with the horizontal position knob enable state:  <b>HORIZONTAL POSITION:&lt;position&gt;</b>	
<b>HORIZONTAL CLOCK: INTERNAL   EXTP   EXTN</b>	Selects internal clock mode with time/div determined by <time/div> (INT) or external clock mode with positive (EXTP) or negative (EXTN) edge sensitivity, respectively.	
<b>HORIZONTAL? CLOCK</b>	Responds with horizontal clock source:  <b>HORIZONTAL CLOCK:&lt;hclock&gt;</b>	
<b>HORIZONTAL?</b>	Responds with all the horizontal setting: <b>HORIZONTAL TIME:&lt;time/div&gt;</b> , <b>POSITION:&lt;position&gt;</b> , <b>CLOCK:&lt;hclock&gt;</b> <time/div> ::= <NR3> <position> ::= ON   OFF <hclock> ::= INTERNAL   EXTP   EXTN	

<sup>a</sup> execution error warning

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

**DISPLAY GROUP**



Command/Query	Description
<b>DISPLAY 1:</b> <b>ON   OFF</b>	Turns the waveform 1 display either ON or OFF.
<b>DISPLAY? 1</b>	Responds with the display status of waveform 1:  <b>DISPLAY 1:&lt;onoff&gt;</b>
<b>DISPLAY 2:</b> <b>ON   OFF</b>	Turns the waveform 2 display either ON or OFF.
<b>DISPLAY? 2</b>	Responds with the display status of waveform 2:  <b>DISPLAY 2:&lt;onoff&gt;</b>
<b>DISPLAY 3:</b> <b>ON   OFF</b>	Turns the waveform 3 display either ON or OFF.
<b>DISPLAY? 3</b>	Responds with the display status of waveform 3:  <b>DISPLAY 3:&lt;onoff&gt;</b>



TABLE 4-7 (CONT)  
 7D20 Command Set

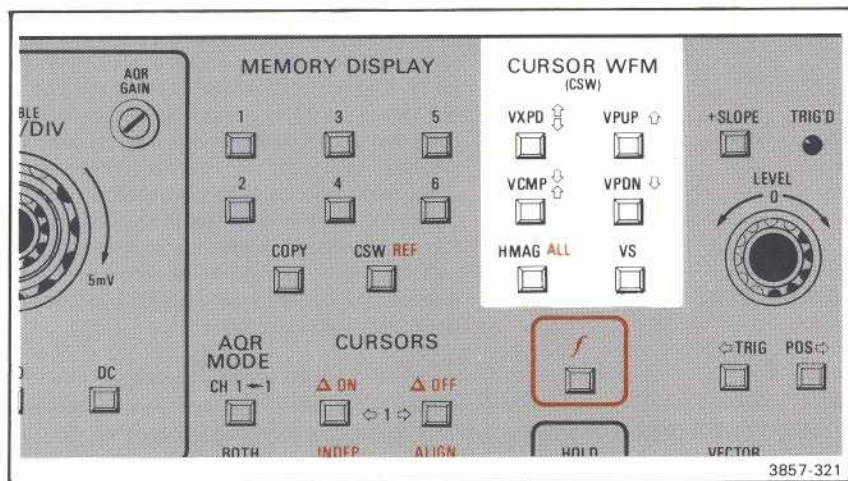
Command/Query	Description
<b>DISPLAY 4:</b> <b>ON   OFF</b>	Turns the waveform 4 display either ON or OFF.
<b>DISPLAY? 4</b>	Responds with the display status of waveform 4:  <b>DISPLAY 4:&lt;onoff&gt;</b>
<b>DISPLAY 5:</b> <b>ON   OFF</b>	Turns the waveform 5 display either ON or OFF.
<b>DISPLAY? 5</b>	Responds with the display status of waveform 5:  <b>DISPLAY 5:&lt;onoff&gt;</b>
<b>DISPLAY 6:</b> <b>ON   OFF</b>	Turns the waveform 6 display either ON or OFF.
<b>DISPLAY? 6</b>	Responds with the display status of waveform 6:  <b>DISPLAY 6:&lt;onoff&gt;</b>
<b>DISPLAY CSW:</b> <wfm #>	Selects waveform <wfm #> as the cursor waveform.
<b>DISPLAY? CSW</b>	Responds with the cursor waveform number:  <b>DISPLAY CSW:&lt;wfm #&gt;</b>
<b>DISPLAY VECTOR:</b> <b>ON   OFF</b>	Turns vector display either ON or OFF.
<b>DISPLAY? VECTOR</b>	Responds with the vector status:  <b>DISPLAY VECTOR:&lt;onoff&gt;</b>
<b>DISPLAY REFERENCE:</b> <b>ON   OFF</b>	Turns the cursor waveform reference display ON or OFF.  Restriction: Reference waveform can only be turned on if the cursor waveform is in HMAG or VS modes; otherwise, an SRQ is generated.

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description
<b>DISPLAY?</b> REFERENCE	Responds with the reference cursor waveform status:  <b>DISPLAY REFERENCE:</b> <onoff>
<b>DISPLAY RDOUT:</b> ON   OFF	Turns display text lines 1, 2, 15, and 16 on or off. DISP RDOUT may be selected by means of the Utilities menu.
<b>DISPLAY?</b> RDOUT	Responds with readout display status:  <b>DISPLAY RDOUT:</b> <onoff>
<b>DISPLAY?</b>	Responds with the waveform display information:  <b>DISPLAY CSW:</b> <csw>,1:<onoff>,2:<onoff>, 3:<onoff>,4:<onoff>,5:<onoff>,6:<onoff>, <b>VECTOR:</b> <onoff>, <b>REFERENCE:</b> <onoff>, <b>RDOUT:</b> <onoff>
<b>COPY 1:</b> <wfm #>	Copies waveform 1 to waveform <wfm #>.
<b>COPY 2:</b> <wfm #>	Copies waveform 2 to waveform <wfm #>.
<b>COPY 3:</b> <wfm #>	Copies waveform 3 to waveform <wfm #>.
<b>COPY 4:</b> <wfm #>	Copies waveform 4 to waveform <wfm #>.
<b>COPY 5:</b> <wfm #>	Copies waveform 5 to waveform <wfm #>.
<b>COPY 6:</b> <wfm #>	Copies waveform 6 to waveform <wfm #>.  <onoff> ::= ON   OFF <wfm #> ::= 1   2   3   4   5   6

TABLE 4-7 (CONT)  
 7D20 Command Set

CURSOR WAVEFORM GROUP



Command/Query	Description																
<b>CSW VOLTS:</b> <volts/div>	This command is ignored by the 7D20.																
<b>CSW? VOLTS</b>	Responds with the cursor waveform's display volts/div:  <b>CSW VOLTS:&lt;volts/div&gt;</b>																
<b>CSW VXP:</b> <setting>	Vertically expands or compresses the cursor waveform by selecting the display volts/div for the cursor waveform.																
	<table border="1"> <thead> <tr> <th>&lt;setting&gt;</th> <th>display volts/div</th> </tr> </thead> <tbody> <tr> <td>&lt;-2.5</td> <td>2 steps lower than acquired<sup>a</sup></td> </tr> <tr> <td>-2.5 to &lt;1.5</td> <td>2 steps lower than acquired</td> </tr> <tr> <td>-1.5 to &lt;0.5</td> <td>1 step lower than acquired</td> </tr> <tr> <td>-0.5 to &lt;0.5</td> <td>same volts/div as acquired</td> </tr> <tr> <td>0.5 to &lt;1.5</td> <td>1 step higher than acquired</td> </tr> <tr> <td>1.5 to &lt;2.5</td> <td>2 steps higher than acquired</td> </tr> <tr> <td>2.5 or greater</td> <td>2 steps higher than acquired<sup>a</sup></td> </tr> </tbody> </table>	<setting>	display volts/div	<-2.5	2 steps lower than acquired <sup>a</sup>	-2.5 to <1.5	2 steps lower than acquired	-1.5 to <0.5	1 step lower than acquired	-0.5 to <0.5	same volts/div as acquired	0.5 to <1.5	1 step higher than acquired	1.5 to <2.5	2 steps higher than acquired	2.5 or greater	2 steps higher than acquired <sup>a</sup>
<setting>	display volts/div																
<-2.5	2 steps lower than acquired <sup>a</sup>																
-2.5 to <1.5	2 steps lower than acquired																
-1.5 to <0.5	1 step lower than acquired																
-0.5 to <0.5	same volts/div as acquired																
0.5 to <1.5	1 step higher than acquired																
1.5 to <2.5	2 steps higher than acquired																
2.5 or greater	2 steps higher than acquired <sup>a</sup>																

<sup>a</sup> execution error warning



TABLE 4-7 (CONT)  
 7D20 Command Set

Command/Query	Description								
	<p>Restriction: Waveform memory displays may be compressed or expanded no more than twice the acquired scale factor (in a 1-2-5 sequence) without incurring error. VXPd does not affect active waveforms except in HMAG, or Y vs X modes when the REFERENCE waveform is displayed. In these cases, an SRQ is generated. See Section 2, Cursor Waveform Keys, for more details.</p>								
<p><b>CSW? VXPd</b></p>	<p>Responds with the cursor waveform's displayed volts/div expansion factor:</p> <p style="text-align: center;"><b>CSW VXPd:&lt;setting&gt;</b></p>								
<p><b>CSW POSITION:</b> &lt;position&gt;</p>	<p>Sets cursor waveform's displayed Vertical position.</p> <table border="1" data-bbox="425 769 917 899"> <thead> <tr> <th data-bbox="425 769 681 808">&lt;position&gt;</th> <th data-bbox="681 769 917 808">position offset</th> </tr> </thead> <tbody> <tr> <td data-bbox="425 808 681 841" style="text-align: center;">&lt;-5.14</td> <td data-bbox="681 808 917 841" style="text-align: center;">-5.12<sup>a</sup></td> </tr> <tr> <td data-bbox="425 841 681 873" style="text-align: center;">-5.12 to 5.08</td> <td data-bbox="681 841 917 873" style="text-align: center;">&lt;position&gt;</td> </tr> <tr> <td data-bbox="425 873 681 899" style="text-align: center;">+5.10 or greater</td> <td data-bbox="681 873 917 899" style="text-align: center;">+5.08<sup>a</sup></td> </tr> </tbody> </table> <p>The position offset is added to the original acquired position value. Vertical expansion will also affect the actual displayed position. Position does not affect active waveforms except in HMAG, or Y vs X modes when the REF waveform is displayed. See Section 2, Cursor Waveform Keys, for more details.</p>	<position>	position offset	<-5.14	-5.12 <sup>a</sup>	-5.12 to 5.08	<position>	+5.10 or greater	+5.08 <sup>a</sup>
<position>	position offset								
<-5.14	-5.12 <sup>a</sup>								
-5.12 to 5.08	<position>								
+5.10 or greater	+5.08 <sup>a</sup>								
<p><b>CSW? POSITION</b></p>	<p>Responds with the cursor waveform's display position:</p> <p style="text-align: center;"><b>CSW POSITION:&lt;position&gt;</b></p>								
<p><b>CSW HMAG:</b> ON   OFF   ALLON   ALLOFF</p>	<p>ON and OFF turns horizontal magnification on or off, respectively, for the CSW waveform. ALLON and ALLOFF affects all displayed waveforms.</p>								

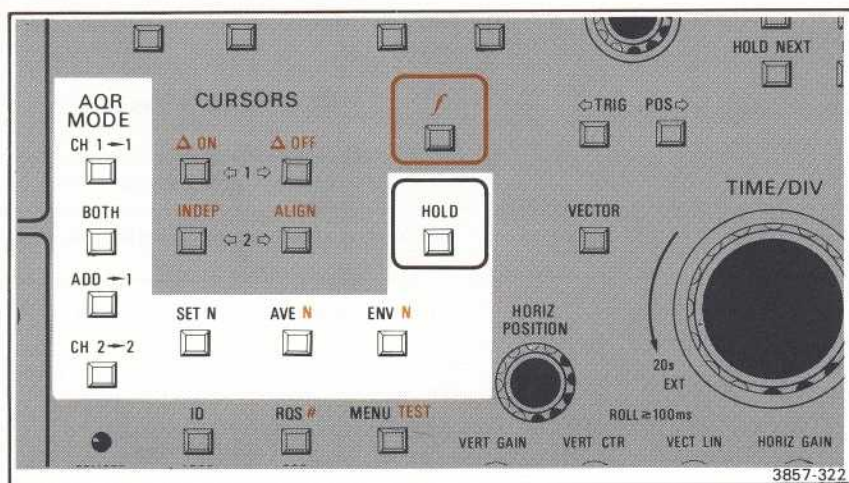
<sup>a</sup>execution error warning

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>CSW? HMAG</b>	<p>Responds with the cursor waveform's hmag setting:</p> <p style="text-align: center;"><b>CSW HMAG:&lt;hmag&gt;</b></p> <p>An OFF response indicates the cursor waveform is not magnified, but other waveforms may be.</p>
<b>CSW VS: &lt;wfm #&gt;  </b>	Displays the cursor waveform versus waveform <wfm #>.
<b>Ø</b>	Turns cursor waveform VS display OFF.
<b>CSW? VS</b>	<p>Responds with the cursor waveform's versus status:</p> <p style="text-align: center;"><b>CSW VS:&lt;versus&gt;</b></p> <p>To interrogate the VS status of waveforms other than the cursor waveform, the cursor waveform must be reselected.</p>
<b>CSW?</b>	<p>Responds with the display information for cursor waveform:</p> <p style="text-align: center;"><b>CSW VOLTS:&lt;volts/div&gt;,VXPD:&lt;setting&gt;, POSITION:&lt;position&gt;, HMAG&lt;hmag&gt;,VS:&lt;versus&gt;</b></p> <p>&lt;volts/div&gt; ::= &lt;NR3&gt;            &lt;setting&gt; ::= -2   -1   0   1   2            &lt;position&gt; ::= &lt;NR2&gt;            &lt;hmag&gt; ::= ON   OFF   ALLON   ALLOFF            &lt;versus&gt; ::= &lt;wfm #&gt;   0            &lt;wfm #&gt; ::= 1   2   3   4   5   6</p>

TABLE 4-7 (CONT)  
7D20 Command Set

ACQUISITION GROUP



Command/Query	Description
<b>AQR MODE:</b> CH1   BOTH   ADD   CH2	Sets acquire mode to either CH1, BOTH, ADD, or CH2. Refer to front-panel section for more details on AQR mode.
<b>AQR? MODE</b>	Responds with the acquire mode setting: <b>AQR MODE:&lt;mode&gt;</b>
<b>AQR HOLD:</b> ON   OFF	Turns acquire hold either ON or OFF.
<b>AQR? HOLD</b>	Responds with the acquire hold status: <b>AQR HOLD:&lt;hold&gt;</b>

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description	
<b>AQR SET:</b> <n>	Sets N to <n>. N is the number of averages or envelopes, as appropriate.	
	<n>	N
	<6	8 <sup>a</sup>
	6 to <12 12 to <24 24 to <48 48 to <96 96 to <192 192 to <384 384 or greater	8 16 32 64 128 256 256 <sup>a</sup>
<b>AQR? SET</b>	responds with <n>:  <div style="text-align: center;"><b>AQR SET:&lt;n&gt;</b></div>	
<b>AQR TYPE:</b> <b>NORMAL</b>	Acquires waveform(s) normally. Terminates HOLD condition.	
<b>AVE</b>	Averages waveform(s) continuously. Terminates HOLD. Averages Wfm 1 or Wfm 2 as determined by AQR Mode control. A GPIB AVE command received during AVE is ignored. AVE terminates ENV, ENVN, SETN, and AVEN. <sup>a</sup>	
<b>AVEN</b>	Averages waveform(s) N times. If AVEN is received during AVE then AVE is terminated and AVEN executes. Upon completion of the AVEN operation, an operation complete SRQ is generated when the 7D20 enters HOLD. <sup>a</sup>	
<b>ENV</b>	Envelopes waveform(s) continuously. Terminates HOLD. Wfm 1 or Wfm 2, as determined by AQR MODE, acquire envelope processed data.  GPIB ENV commands received during execution are ignored.  ENV terminates AVE, AVEN, SETN, and ENVN. <sup>a</sup>	

<sup>a</sup> See Section 2 for more information

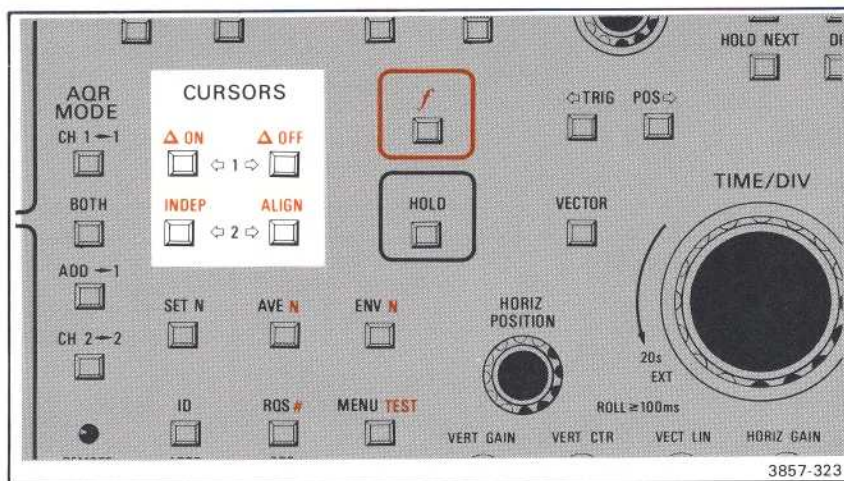
TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>ENVN</b>	<p>Envelopes waveform(s) N times.</p> <p>Upon completion of the ENVN operation, an operation complete SRQ is generated when the 7D20 enters HOLD.<sup>a</sup></p>
<b>AQR? TYPE</b>	<p>responds with the acquire type status:</p> <p style="text-align: center;"><b>AQR TYPE:&lt;type&gt;</b></p>
<b>AQR?</b>	<p>returns the current acquisition status:</p> <p><b>AQR MODE:&lt;mode&gt;,TYPE:&lt;type&gt;, HOLD:&lt;hold&gt;,SET:&lt;n&gt;,</b></p> <p>&lt;mode&gt; ::= CH1   BOTH   ADD   CH2            &lt;hold&gt; ::= ON   OFF            &lt;n&gt; ::= 8   16   32   64   128   256            &lt;type&gt; ::= NORMAL   AVE   AVEN   ENV   ENVN</p>



TABLE 4-7 (CONT)  
7D20 Command Set

CURSOR GROUP



Command/Query	Description														
<b>CURSOR MODE:</b> <b>INDEP   ALIGN</b>	Sets cursor mode to INDEP or ALIGN.														
<b>CURSOR? MODE</b>	Responds with the cursor mode: <b>CURSOR MODE:&lt;mode&gt;</b>														
<b>CURSOR DELTA:</b> <b>ON   OFF</b>	Turns cursor 2 either ON or OFF.														
<b>CURSOR? DELTA</b>	Responds with the cursor delta mode: <b>CURSOR DELTA:&lt;onoff&gt;</b>														
<b>CURSOR 1:</b> <point #>	Sets cursor 1 to the valid position nearest <point#>.														
	<table border="1"> <thead> <tr> <th>&lt;point #&gt;</th> <th>point #</th> </tr> </thead> <tbody> <tr> <td>&lt;-0.5</td> <td>0<sup>a</sup></td> </tr> <tr> <td>-0.5 to &lt;819.5</td> <td>&lt;point #&gt;</td> </tr> <tr> <td>&gt;819.5</td> <td>819<sup>a</sup></td> </tr> <tr> <td>&lt;-0.5</td> <td>0<sup>a</sup></td> </tr> <tr> <td>-0.5 to &lt;1023.5</td> <td>&lt;point #&gt;</td> </tr> <tr> <td>≥1023.5</td> <td>1023<sup>a</sup></td> </tr> </tbody> </table>	<point #>	point #	<-0.5	0 <sup>a</sup>	-0.5 to <819.5	<point #>	>819.5	819 <sup>a</sup>	<-0.5	0 <sup>a</sup>	-0.5 to <1023.5	<point #>	≥1023.5	1023 <sup>a</sup>
<point #>	point #														
<-0.5	0 <sup>a</sup>														
-0.5 to <819.5	<point #>														
>819.5	819 <sup>a</sup>														
<-0.5	0 <sup>a</sup>														
-0.5 to <1023.5	<point #>														
≥1023.5	1023 <sup>a</sup>														

<sup>a</sup>execution error warning

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description	
<b>CURSOR? 1</b>	Responds with cursor 1's point #:  <b>CURSOR 1:&lt;point #&gt;</b>	
<b>CURSOR 2:</b> <point #>	Sets cursor 2 to the valid position nearest <point #>.	
	<point #>	point #
p/w = 820:	<-0.5 -0.5 to <819.5 >819.5	0 <sup>a</sup> <point #> 819 <sup>a</sup>
p/w = 1024:	<-0.5 0.5 to <1023.5 ≥1023.5	0 <sup>a</sup> <point #> 1023 <sup>a</sup>
	Restriction: Cursor 2 may not be set to a position less than Cursor 1.	
<b>CURSOR? 2</b>	Responds with cursor 2's status:  <b>CURSOR 2:&lt;point #&gt;</b>	
<b>CURSOR?</b>	Responds with the cursor status:  <b>CURSOR MODE:&lt;mode&gt;,DELTA:&lt;onoff&gt;,1: &lt;point #&gt;,2:&lt;point #&gt;</b>  <mode> ::= INDEP   ALIGN <point #> ::= <NR1> <onoff> ::= ON   OFF	
<b>COORD? VCRD</b>	Responds with the displayed vertical cursor coordinate ( $\Delta V$ value if cursor delta mode is on):  <b>COORD VCRD:&lt;vert coord&gt;</b>	
<b>COORD? HCRD</b>	Responds with the displayed horizontal cursor coordinate ( $\Delta t$ value if cursor delta mode is on):  <b>COORD HCRD:&lt;horz coord&gt;</b>	

<sup>a</sup> execution error warning

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>COORD?</b>	<p>Responds with the displayed cursor coordinates:</p> <p><b>COORD VCRD:</b>&lt;vert coord&gt;,HCRD:   &lt;horz coord&gt;</p> <p>&lt;vert coord&gt; ::= &lt;NR3&gt; &lt;horz coord&gt; ::= &lt;NR3&gt;</p>

STORED SETTINGS GROUP



Command/Query	Description
<b>STORE</b> <set #>	Stores front panel settings in memory location <set #>.
<b>RECALL</b> <set #>	<p>Recalls front panel settings from memory location &lt;set #&gt;.</p> <p>&lt;set #&gt; ::= 1   2   3   4   5   6</p>



TABLE 4-7 (CONT)  
7D20 Command Set

DEVICE TRIGGER

NOTE

*Device Trigger is Bus-unique; that is, no front-panel equivalent exists in the 7D20.*

Command/Query	Description
<p>DT OFF              HOLD   HOLDNEXT              AVE   AVEN              ENV   ENVN              NORMAL  </p>	<p>Disables 7D20 response to GET.</p> <p>Executes deferred command on GET interface message. Only one deferred command may be triggered from each GET. Clears previous pending deferred commands without executing them.</p>
<p>DT?</p>	<p>Responds with deferred command to be executed on next GET interface message.</p> <p style="text-align: center;"><b>DT &lt;deferred command&gt;</b></p> <p>&lt;deferred command&gt; ::= OFF   HOLD   HOLDNEXT   AVE   AVEN   ENV   ENVN   NORMAL</p>

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

**INITIALIZATION GROUP**

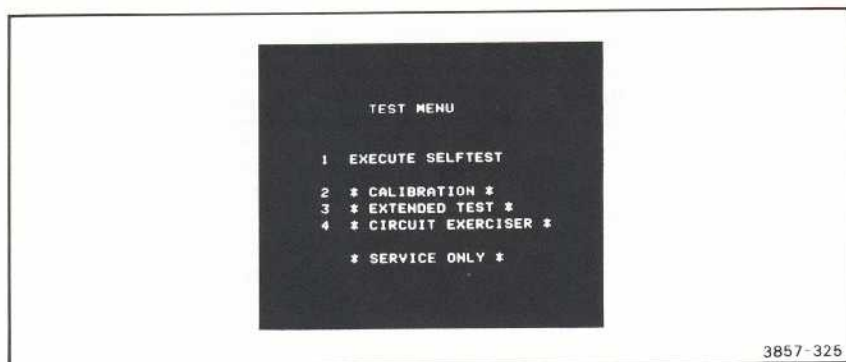
**NOTE**

*The Initialization Group is Bus-unique.*

Command/Query	Description
<b>INIT</b>	Full initialization equivalent to an INIT PANEL, WAVEFORM, GPIB command.
<b>INIT PANEL</b>	Initializes front panel settings as described in front-panel section.
<b>INIT WAVEFRM</b>	Initializes waveform memories 1-6 by setting the vertical data and vertical zero reference (VZR) values to zero and display attributes (VPUP, VPDN, VXPDP, VCMP, HMAG, VS, and REF) to the off condition.
<b>INIT GPIB</b>	<p>Initializes the status of the following functions, which affect GPIB operation:</p> <p>Sets:</p> <p>DT           OFF  RQS          ON  CER          ON  EXR          OFF  INR          ON  SYS          ON  EXW          ON  OPC          ON  USER         ON  PID          OFF  DEBUG       OFF  LONGFORM    ON  RECORDING   OFF  DATA MEMORY:1  DATA ENCODING: BINARY  DATA INTERPOLATE: OFF</p> <p>Clears:</p> <p>Event Buffer</p>

TABLE 4-7 (CONT)  
7D20 Command Set

## SELFTEST



The 7D20 performs a selftest routine that tests 63 circuit modules. If a fault condition is detected, an SRQ is generated with an Internal Error status byte. An event query may be used to determine the specific module at fault. When the selftest is complete, an SRQ and an Operation Complete status byte are generated. Upon completion or exit from the selftest routine, the front panel settings are restored to the status prior to selftest and the waveform memory is initialized.

Command/Query	Description
<b>TEST START</b>	Initiates selftest routine.
<b>CONTINUE</b>	Continues selftest routine having halted on fault condition.
<b>EXIT</b>	Exits selftest routine that has halted on fault condition. Will not terminate an executing selftest.
<b>TEST</b>	Same as TEST START.

TABLE 4-7 (CONT)  
7D20 Command Set

CALIBRATION GROUP



Command/Query	Description
<b>CAL DISPLAY</b>	Turns front-panel display calibration pattern on.
<b>DDAC</b>	Initiates vertical display DAC calibration routine.
<b>POSOFF</b>	Initiates vertical position offset calibration routine.
<b>POSGAIN</b>	Initiates vertical position gain calibration routine.
<b>RAMP</b>	Initiates fast ramp calibration routine.
<b>OFF</b>	Turns off selected calibration routine.
<b>CAL?</b>	Responds with calibration selection status:  <b>CAL DISPLAY</b>   <b>DDAC</b>   <b>VERTICAL</b>   <b>POSOFF</b>   <b>POSGAIN</b>   <b>RAMP</b>   <b>OFF</b>
<b>RAMP?</b>	Responds with fast ramp calibration routine offset gain and count values:  <b>RAMP GAIN:</b> <gain>, <b>COUNT:</b> <count>  <gain> ::= <NR2> <count> ::= <NR1>

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

**SERVICE REQUEST GROUP**

**NOTE**

*These Commands and Queries are all bus-unique; that is, no front-panel equivalents exist.*

Command/Query	Description
<b>RQS ON   OFF</b>	Enables/disables all service request functions except power-on.
<b>RQS?</b>	Responds with the RQS service request status:  <b>RQS&lt;onoff&gt;</b>
<b>CER ON   OFF</b>	Enables/disables command-error service request.
<b>CER?</b>	Responds with the CER service request status:  <b>CER&lt;onoff&gt;</b>
<b>EXR ON   OFF</b>	Enables/disables execution-error service request.
<b>EXR?</b>	Responds with the EXR service request status:  <b>EXR&lt;onoff&gt;</b>
<b>INR ON   OFF</b>	Enables/disables internal-error service request.
<b>INR?</b>	Responds with the INR service request status:  <b>INR&lt;onoff&gt;</b>
<b>EXW ON   OFF</b>	Enables/disables execution-warning service request.
<b>EXW?</b>	Responds with the EXW service request status:  <b>EXW&lt;onoff&gt;</b>

TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>OPC ON   OFF</b>	Enables/disables operation-complete service request.
<b>OPC?</b>	Responds with the OPC service request status:  <b>OPC&lt;onoff&gt;</b>
<b>USER ON   OFF</b>	Enables/disables front panel generated RQS Key service request.
<b>USER?</b>	Responds with the USER service request status:  <b>USER&lt;onoff&gt;</b>
<b>PID ON   OFF</b>	Enables/disables probe-ID service request.
<b>PID?</b>	Responds with the PID service request status:  <b>PID&lt;onoff&gt;</b>
<b>SRQ?</b>	Responds with all service request settings:  <b>RQS&lt;onoff&gt;;CER&lt;onoff&gt;;EXR&lt;onoff&gt;; INR&lt;onoff&gt;;EXW&lt;onoff&gt;;OPC&lt;onoff&gt;; USER&lt;onoff&gt;;PID&lt;onoff&gt;</b>  <onoff> ::= ON   OFF

**TABLE 4-7 (CONT)**  
**7D20 Command Set**

**WAVEFORM PREAMBLE GROUP**

**NOTE**

*The Waveform Preamble Group is Bus-unique.*

Command/Query	Description
<b>NOTE</b>	
<p><i>WFMPRE commands that are not used by the 7D20 will be accepted as part of the waveform preamble or individually, but will be ignored. No error or warning will be issued. This permits preamble outputs to be input without generating unnecessary service requests.</i></p>	
<b>WFMPRE WFID:</b>	Command is not used by 7D20. Destination waveform is set with the DATA command.
<b>WFMPRE? WFID</b>	<p>Responds with the current source waveform's number and indicates if interpolation will be applied to curve data:</p> <p style="text-align: center;"><b>WFMPRE WFID:&lt;wfmid&gt;</b></p>
<b>WFMPRE ENCDG:</b> ASCII   BINARY	Command is not used by 7D20. Data encoding is set with the DATA command.
<b>WFMPRE? ENCDG</b>	<p>Responds with the current curve encoding:</p> <p style="text-align: center;"><b>WFMPRE ENCDG:&lt;encdg&gt;</b></p>
<b>WFMPRE NR.PT:</b> 820   1024	Sets number of points to input to 820   1024.
<b>WFMPRE? NR.PT</b>	<p>Responds with the current source waveform's points/waveform:</p> <p style="text-align: center;"><b>WFMPRE NR.PT:&lt;p/w&gt;</b></p>
<b>WFMPRE PT.FMT: Y</b>	Command is not used by 7D20. Point format is fixed as Y format.
<b>WFMPRE? PT.FMT</b>	<p>Responds with a point format of Y:</p> <p style="text-align: center;"><b>WFMPRE PT.FMT:Y</b></p>



TABLE 4-7 (CONT)  
7D20 Command Set

Command/Query	Description
<b>WFMPRE XINCR:</b> <x increment>	Sets time between points to <x increment>.  <x increment> ::= time/div / 80 points/division for 820 point waveforms.  <x increment> ::= time/div / 100 points/division for 1024 point waveforms
<b>WFMPRE? XINCR</b>	Responds with the time between points of the current source waveform:  <b>WFMPRE XINCR:&lt;x increment&gt;</b>
<b>WFMPRE PT.OFF:</b> <point #>	Selects the waveform point at which the cursor reads time zero. Range is -150,000 ≤point # ≤1023, where point 0 is the left-most displayed point on the waveform. An execution error warning is given if an out of range condition occurs.
<b>WFMPRE? PT.OFF</b>	Responds with the trigger point #:  <b>WFMPRE PT.OFF:&lt;point #&gt;</b>
<b>WFMPRE XZERO:</b> 0	Command is not used by 7D20. XZERO is fixed as 0.
<b>WFMPRE? XZERO</b>	Responds with an XZERO of 0:  <b>WFMPRE XZERO:0</b>
<b>WFMPRE XUNIT:</b> S	Sets displayed horizontal scale factor unit for waveforms input from the curve command.
<b>WFMPRE? XUNIT:</b> <xunit>	Responds with XUNIT,  <b>WFMPRE XUNIT:&lt;xunit&gt;</b>
<b>WFMPRE YMULT:</b> <y multiplier>	Sets vertical scale (volts/div) to <y multiplier>.
<b>WFMPRE? YMULT</b>	Responds with the current source waveforms vertical scale factor:  <b>WFMPRE YMULT:&lt;y multiplier&gt;</b>



**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description
<b>WFMPRE YZERO:</b> <y zero>	Sets Y zero to <y zero>.  <y zero> ::= -(volts/div × VZR (vertical zero reference)) Where VZR is prior to vertical expansion, or positioning.
<b>WFMPRE?</b> YZERO	Responds with the current source waveform vertical zero:  <b>WFMPRE YZERO:</b> <y zero>
<b>WFMPRE YUNIT:</b> V	Sets displayed vertical scale factor unit for waveforms input from the CURVE command.
<b>WFMPRE YUNIT:</b> <y unit>	Responds with YUNIT,  <b>WFMPRE YUNIT:</b> <y unit>
<b>WFMPRE BYT/NR:</b> 1	Command is not used by 7D20. Binary data field width is fixed as one byte.
<b>WFMPRE?</b> BYT/NR	Responds with a field width of 1 byte:  <b>WFMPRE BYT/NR:</b> 1
<b>WFMPRE BN.FMT:</b> LF	Command is not used by 7D20. Binary number format is fixed as binary-fraction.
<b>WFMPRE?</b> BN.FMT	Responds with a number format of binary fraction:  <b>WFMPRE BN.FMT:</b> LF
<b>WFMPRE BIT/NR:</b> 8	Command is not used by 7D20. Binary data precision is fixed as eight bits.
<b>WFMPRE?</b> BIT/NR	Responds with a precision of 8 binary bits:  <b>WFMPRE BIT/NR:</b> 8

TABLE 4-7 (CONT)  
 7D20 Command Set

Command/Query	Description
<b>WFMPRE? CRVCHK</b>	<p>Responds with binary curve that includes a checksum:</p> <p style="text-align: center;"><b>WFMPRE CRVCHK: CHKSM0</b></p>
<b>WFMPRE?</b>	<p>Responds with the current source waveform's preamble:</p> <p>If ASCII:  <b>WFMPRE WFID:&lt;wfmid&gt;,ENCDG:ASCII, NR.PT:&lt; p/w&gt;,PT.FMT:Y,XINCR:&lt;x increment&gt;,PT.OFF:&lt;point #&gt;,XZERO:0, XUNIT:&lt;xunit&gt;YMULT:&lt;y multiplier&gt;, YZERO:&lt;y zero&gt;,YUNIT:&lt;y unit&gt;</b></p> <p>If binary:  <b>WFMPRE WFID:&lt;wfmid&gt;,ENCDG:BINARY, NR.PT:&lt;p/w&gt;,PT.FMT:Y,XINCR:&lt;x increment&gt;,PT.OFF:&lt;point #&gt;,XZERO:0,XUNIT:&lt;xunit&gt;YMULT:&lt;y multiplier&gt;,YZERO:&lt;y zero&gt;,Y UNIT:&lt;y unit&gt;, BYTNR:1,BN.FMT:LF, BITNR:8,CRVCHK:CHKSM0,</b></p> <p>&lt;wfm #&gt; ::= 1   2   3   4   5   6          &lt;encdg&gt; ::= ASCII   BINARY          &lt;x increment&gt; ::= &lt;NR3&gt; = time/div / points/div          &lt;point #&gt; ::= &lt;NR1&gt;          &lt;x multiplier&gt; ::= &lt;NR3&gt; = vertical scale factor          &lt;y zero&gt; ::= &lt;NR3&gt; = -(VZR * vertical scale factor)          &lt;wfmid&gt; ::= W &lt;wfm #&gt;   W &lt;wfm #&gt;, where I indicates interpolated data.          &lt;x unit&gt; ::= Capability to be set to any one of the the 7D20 single character set, except for (,), (;), or ESC (escape).          &lt;y unit&gt; ::= Same as above. If x unit or y unit is set to "&amp;", the 7D20 will interpret this as a blank unit or space.</p>

TABLE 4-7 (CONT)  
7D20 Command Set

WAVEFORM CURVE

NOTE

*Waveform Curve is Bus-unique*

Command/Query	Description
<p><b>CURVE</b> &lt;ASCII curve&gt;</p> <hr/> <p>&lt;binary block curve&gt;</p>	<p>Loads destination waveform with &lt;ASCII curve&gt;.</p> <hr/> <p>Loads destination waveform with &lt;binary block curve&gt;.</p> <p>&lt;ASCII curve&gt; ::= &lt;ASCII point&gt;,&lt;ASCII point&gt;}...</p> <p>&lt;binary block curve&gt; ::= %&lt;binary count&gt;&lt;binary point&gt;...&lt;checksum&gt;</p>
<p><b>CURVE?</b></p>	<p>Responds with the current source waveform's curve:</p> <p>If ASCII: <b>CURVE</b> &lt;ASCII point&gt;,&lt;ASCII point&gt;}...</p> <p>If binary: <b>CURVE</b> %&lt;binary count&gt;&lt;binary point&gt;...&lt;checksum&gt;</p> <p>&lt;ASCII point&gt; ::= &lt;NR2&gt;                  &lt;binary count&gt; ::= two bytes representing number of data points +1                  &lt;binary point&gt; ::= 8-bit byte (00   01   02  ...  FE   FF)                  &lt;checksum&gt; ::= 2's complement of the modulo 256 sum of the preceding binary data bytes and the binary count but not the "%" preceding the binary byte count.</p>

TABLE 4-7 (CONT)  
 7D20 Command Set

WAVEFORM PREAMBLE AND CURVE

NOTE

*Waveform Preamble and Curve is Bus-Unique*

Command/Query	Description
<b>WAVFRM?</b>	Responds with <preamble><curve> <preamble> ::= response to <b>WFMPRE?</b> <curve> ::= response to <b>CURVE?</b>
<b>DATA ENCDG:</b> <b>ASCII   BINARY</b>	Sets encoding for transmission of waveform data from 7D20 as binary or ASCII.
<b>DATA? ENCDG</b>	Responds with curve data encoding for waveforms transmitted from 7D20:  <b>DATA ENCDG:&lt;encdg&gt;</b>
<b>DATA INTERPOLATE:</b> <b>ON   OFF</b>	Sets 820 to 1024-point interpolation mode for all Extended Real-time Waveforms transmitted from 7D20.
<b>DATA? INTERPOLATE</b>	Responds with status of interpolation mode:  <b>DATA INTERPOLATE:&lt;onoff&gt;</b>
<b>DATA MEMORY: &lt;wfm #&gt;</b>	Sets memory destination and source for all waveform data transmission to and from 7D20.
<b>DATA? MEMORY</b>	Responds with source/destination for waveform data transmission:  <b>DATA MEMORY:&lt;wfm #&gt;</b>
<b>DATA?</b>	Responds with data command status: <b>DATA ENCDG:&lt;encdg&gt;;DATA INTERPOLATE:ON   OFF DATA MEMORY:&lt;wfm #&gt;</b> <encdg> ::= ASCII   BINARY <wfm #> ::= 1   2   3   4   5   6

TABLE 4-7 (CONT)  
7D20 Command Set

READOUT/TEXT

NOTE

*Readout/Text is Bus-Unique.*

Command/Query	Description
<b>TEXT</b> "text"	Displays text in center 12 lines of CRT. See discussion on TEXT command earlier in this section.
<b>TEXT?</b>	<p>Responds with the center 12 lines of display text:</p> <pre>TEXT "&lt;line 3&gt;&lt;CR&gt;&lt;line 4&gt; &lt;CR&gt;&lt;line 5&gt;&lt;CR&gt;... &lt;line 12&gt;&lt;CR&gt;&lt;line 13&gt; &lt;CR&gt;&lt;line 14&gt;&lt;CR&gt;"</pre> <p>&lt;line 3&gt; ::= third line of display                      &lt;line 4&gt; ::= fourth line of display                      .                      .                      .                      &lt;line 13&gt; ::= thirteenth line of display                      &lt;line 14&gt; ::= fourteenth line of display</p>

NOTE

*A double quote (") within a line must be sent as two double quotes (" ").*

<b>RDOUT?</b>	<p>Responds with the four lines of readout:</p> <pre>RDOUT "&lt;line 1&gt;&lt;CR&gt;&lt;line 2&gt; &lt;CR&gt;&lt;line 15&gt;&lt;CR&gt; &lt;line 16&gt;&lt;CR&gt;"</pre> <p>&lt;CR&gt; ::= carriage return</p>
---------------	---

TABLE 4-7 (CONT)  
7D20 Command Set

## PROGRAMMING AIDS

## NOTE

*The following commands and queries are bus-unique.*

Command/Query	Description
<b>ID?</b>	<p>Responds with the 7D20's ID:</p> <p><b>ID TEK/7D20,V81.1,&lt;rom&gt;.&lt;patch&gt;</b></p> <p>&lt;rom&gt; ::= 2 character ROM version &lt;patch&gt; ::= 2 digit PATCH revision</p>
<b>SET?</b>	<p>Responds with the front panel settings:</p> <p><b>&lt;ch1&gt;;&lt;ch2&gt;;&lt;horz&gt;;&lt;aqr&gt;;&lt;csw&gt;;&lt;disp&gt;;&lt;trig&gt;;&lt;curs&gt;</b></p> <p>&lt;ch1&gt; ::= response to CH1? &lt;ch2&gt; ::= response to CH2? &lt;trig&gt; ::= response to TRIG? &lt;horz&gt; ::= response to HORIZ? &lt;disp&gt; ::= response to DISP? &lt;csw&gt; ::= response to CSW? &lt;aqr&gt; ::= response to AQR? &lt;curs&gt; ::= response to CURS?</p>
<b>HELP?</b>	<p>Responds with a list of all valid command headers:</p> <p><b>CH1, CH2, TRIGGER, HORIZONTAL, DISPLAY,COPY, CSW, AQR, CURSOR, STORE, RECALL,DT, INIT, TEST, CAL, RQS, CER, EXR, INR, EXW, OPC, USER, PID, SRQ, WFMPRE, CURVE, DATA, WAVFRM, TEXT, DEBUG, RECORDING, LONGFORM</b></p>
<b>DEBUG ON   OFF</b>	<p>Turns on debug option. Commands are displayed in the order of occurrence on the screen area normally used by text commands and menus. A command or query that follows an EOI will clear the text area before it is displayed. If an error occurs, the erroneous item is displayed, followed by the 16-bit error code describing the type of error. Control characters are displayed as "@" . Lower case characters are display-</p>



**TABLE 4-7 (CONT)**  
**7D20 Command Set**

Command/Query	Description
<b>DEBUG ON   OFF</b> (cont)	ed as upper-case characters. An EOI is displayed as an exclamation mark (!). ASCII waveform transmissions are displayed, binary waveform transmissions are not.
<b>DEBUG?</b>	Responds with the debug status: <b>DEBUG&lt;onoff&gt;</b>
<b>LONGFORM ON OFF</b>	Affects results when querying response to either receive longform or shortform of Command or Query.
<b>LONGFORM?</b>	Responds with the LONGFORM status: <b>LONGFORM ON or LO OF</b>
<b>RECORDING ON OFF</b>	By using a controller and the RECORDING command, it is possible to get longer record lengths. The RECORDING mode works only from 100 mS/div to 20 S/div, including EXT clock mode. In the other time/div regions, RECORDING will have no effect. RECORDING may be used with any acquire mode, and with the normal, AVE and ENV, acquire types. RECORDING allows the user to read out points using a CURVE? command, 1k at a time. The waveforms appear to roll across the screen, just as in non-RECORDING mode, but every time 1024 new points have been acquired they are copied into a RAM buffer accessible via the CURVE? command. There is a separate buffer for each channel. The user knows it is time to do the next CURVE? command, when an OPC service request (66 or 82) is produced by the 7D20. If the user receives the service request before the 1024 points have been read out from the CURVE?, then an overrun condition has occurred, and either the controller needs to take less time to read in the curve, or you should use a slower time/div.
<b>RECORDING?</b>	Responds with RECORDING status: <b>RECORDING ON or RECORDING OFF</b>

## SAMPLE PROGRAMS

### INTRODUCTION

The following pages contain sample programs that may be used in the design of programs that fit task-related applications and specific requirements where a 7D20 is used. The programs are set up for both a Tektronix 4050-series and 4041 controller. Also, the programs are divided into two classes, operations and applications. Each of the programs consists of four parts: Introduction, Remarks, Program, and Comments, where each applies.

### SAMPLE PROGRAM LIST

#### 4050-Series Operating Programs

1. Text Generation: 4050-Series to 7D20.
2. Transfer text to Controller.
3. Transfer Waveform Data to Controller—ASCII format.
4. Transfer Waveform Data to Controller—binary format.
5. Transfer Waveform Data to 7D20—ASCII format.
6. Transfer Waveform Variables and Arrays to 7D20.
7. Query routine.
8. Implement and store 7D20 settings.
9. Query functions.
10. Print Poll statement.

#### 4050-Series Application Programs

1. Event Capture.
2. Store 7D20 settings on Magtape.
3. SRQ Decoding Routine.

#### 4041 Operating Programs

1. ASCII String Array Waveform Transfer to controller.
2. ASCII String Array Waveform Transfer to 7D20.
3. Binary Numeric Array Waveform Transfer to controller.
4. Binary Numeric Array Waveform Transfer to 7D20.
5. Output 7D20 Front-panel Settings—String Array.
6. Input 7D20 Front-panel Settings—String Array.
7. Setting the 7D20 to Remote.
8. Setting the 7D20 to Local.
9. Text Transfer: 7D20 to Tape.
10. Text Transfer: Tape to 7D20.
11. Text Generation: controller to 7D20.

#### 4041 Application Programs

1. SRQ Handler
2. Store and Recall Front-panel Settings.
3. Event Capture.



## 4050-Series Operating Programs

**1. Text Generation: 4050-Series to 7D20.** This program displays text information on crt displays of 7D20 signals. By means of this program, dates, sample numbers, operator names, etc., may be entered. The 4050 constructs an ASCII string that consists of individual lines of text that are to appear on the oscilloscope crt. The lines of text are separated by a carriage return character.

### REMARKS:

```

100 REM * TRANSFER TEXT FROM 4050 TO 7D20 CRT
110 REM * A= 7D20 GPIB DEVICE ADDRESS
120 REM *
130 REM * LIMIT IS 36 CHARACTERS PER LINE OF TEXT
140 REM * LIMIT OF APPROXIMATELY 100 CHARACTERS TOTAL
150 REM *
160 REM * BUILD TEXT IN STRING T#
    
```

### PROGRAM:

```

170 A=20
180 ON SRQ THEN 500
190 DIM T$(12*36)
200 PRINT "INPUT TEXT AS IT IS TO BE DISPLAYED ON 7D20"
210 PRINT "<SPACE><RETURN>FOR A BLANK LINE"
220 PRINT "<RETURN> ONLY TO END INPUT"
230 L=0
240 T=""
250 T%=CHR(34)
260 Q%=T%
270 C%=CHR(13)
280 FOR I=1 TO 12
290 PRINT "-->";
300 INPUT L$
310 IF LEN(L$)<1 THEN 460
320 IF LEN(L$)+LEN(C$)>36 THEN 390
330 L=LEN(L$)
340 T=T+L
350 IFT>100 THEN 410
360 T%=T%&L$
370 T%=T%&C%
380 GO TO 450
390 PRINT "TOO MANY CHARACTERS, RE-ENTER LINE"
400 GO TO 290
410 PRINT "TEXT EXCEEDS 100 CHARACTERS BY",T-100
420 PRINT "RE-ENTER LINE TO CONFORM TO LIMITS"
430 T=T-L
440 GO TO 290
450 NEXT I
460 S=0
470 T%=T%&Q%
480 PRINT @A:"TEXT",T%
490 END
500 POLL D,S;A
510 RETURN
    
```

**COMMENTS:**

```

170 SET 7D20 GPIB DEVICE ADDRESS
180 ENABLE SRQ HANDLER AT LINE 470
190 DIMENSION TEXT STRING T$
200-220 PRINT INSTRUCTION SET
230-270 INITIALIZE VARIABLES
280 LOOP COUNTER FOR TWELVE LINES OF TEXT
290 PROMPT THE USER TO ENTER ONE LINE OF TEXT
300 INPUT ONE LINE OF TEXT
310 CHECK FOR LAST LINE OF TEXT
320 CHECK FOR LINE THAT EXCEEDS LEGAL LENGTH
330-350 CHECK FOR TOTAL TEXT LENGTH LESS THAN 1000 CHAR
360 ADD CURRENT LINE TO PREVIOUS TEXT IN T$
370 ADD CARRIAGE RETURN TO END OF LINE
380 JUMP TO LINE 450
390 PRINT ERROR MESSAGE
400 JUMP TO LINE 290
410-420 PRINT ERROR MESSAGE
440 JUMP TO LINE 290
450 END OF LOOP
460 CLEAR STATUS FLAG
470 ADD QUOTATION MARK TO END OF TEXT STRING
480 PREPARE THE 7D20 TO ACCEPT TEXT INFORMATION AND
    SEND THE TEXT STRING
490 STOP
500 SERIAL POLL THE 7D20: S=STATUS BYTE
510 RETURN FROM SRQ HANDLING ROUTINE

```

**2. Transfer text to 4050-series.** This program causes the controller to read the four lines of readout data and any text that appears on the 7D20 display. The four lines of readout are each assigned to an ASCII string for proper placement when printed on the controller graphics display.

**REMARKS:**

```

100 REM * TRANSFER TEXT FROM 7D20 DISPLAY TO 4050
110 REM *
120 REM * 7D20 ADDRESS = "A"
    
```

**PROGRAM:**

```

130 A=20
140 ON SRQ THEN 300
145 WBYTE@20:
150 DIM T$(200)
160 PAGE
170 PRINT @A: "RDOUT?"
180 INPUT @A: A$, B$, C$, D$
190 PRINT A$
200 PRINT B$
205 FOR I=1 TO 12
210 PRINT @A: "TEXT?"
220 INPUT @A: T$
230 PRINT T$
250 DELETE T$
260 NEXT I
270 PRINT C$
280 PRINT D$
290 END
300 POLL D, S;A
310 RETURN
    
```

**COMMENTS:**

```

130 SET 7D20 GPIB DEVICE ADDRESS
140 ENABLE SRQ HANDLER AT LINE 270
150 DIMENSION THE TEXT ARRAY
160 PAGE THE GRAPHICS DISPLAY
170 REQUEST THE 7D20'S READOUT DATA
180 INPUT EACH LINE OF READOUT INTO A SEPARATE STRING
190 PRINT THE FIRST LINE OF READOUT
200 PRINT THE SECOND LINE OF READOUT
205-260 INPUT AND PRINT THE 12 LINES OF TEXT
270 PRINT THE THIRD LINE OF READOUT
280 PRINT THE FOURTH LINE OF READOUT
290 STOP
300 SERIAL POLL THE 7D20: S=STATUS BYTE
310 RETURN FROM SRQ HANDLING ROUTINE
    
```

**3. Transfer Waveform Data to Controller—ASCII format.** Using the ASCII format, transferring data from the 7D20 to the controller is relatively simple. Once the string arrays are dimensioned, all that is left is to input the waveform preamble and curve information.

**REMARKS:**

```

100 REM * TRANSFER WAVEFORM FROM MEMORY 1 TO 4050 ASCII STRINGS
110 REM * A=7D20 DEVICE ADDRESS
120 REM *
130 REM * INPUT WAVEFORM PREAMBLE INTO ASCII STRING P#
140 REM * INPUT WAVEFORM CURVE INTO ASCII STRING W#

```

**PROGRAM:**

```

150 A=20
160 ON SRQ THEN 260
170 DIM P*(200),W*(7000)
180 S=0
190 PRINT @A:"DATA ENCDG: ASC"
200 PRINT @A:"DATA MEM: 1"
210 PRINT @A:"WFMPRE?"
220 INPUT @A:P#
230 PRINT @A:"CURVE?"
240 INPUT @A:W#
250 END
260 POLL D,S;A
270 RETURN

```

**COMMENTS:**

```

150 SET 7D20 GPIB DEVICE ADDRESS
160 ENABLE SRQ HANDLER AT LINE 260
170 DIMENSION ASCII STRINGS P# AND W#
180 CLEAR STATUS FLAG
190 TELL THE 7D20 TO SEND ASCII ENCODED DATA
200 SELECT THE 7D20 WAVEFORM MEMORY TO BE READ
210 REQUEST THE WAVEFORM PREAMBLE DATA
220 INPUT AND STORE THE PREAMBLE DATA IN STRING P#
230 REQUEST THE WAVEFORM CURVE INFORMATION
240 INPUT AND STORE THE CURVE DATA IN STRING W#
250 STOP
260 SERIAL POLL THE 7D20: S=STATUS BYTE
270 RETURN FROM SRQ HANDLING ROUTINE

```

**4. Transfer Waveform Data to Controller—binary format.** Although it is much more simple to transfer data in ASCII format, the binary format is much faster. The 4050-series controllers can ignore or strip off ASCII characters when reading data into variables and arrays. It may sometimes be necessary, however, to retain this ASCII-encoded data. In lines 330-340 of the program below, the portion of the curve data that consisted of 'CURVE WFM# %' was saved by specifying '%' as an alternate delimiter for the controller. A routine for checksum error detection is also included that is not automatically done by the controller. This routine comprises lines 380-410.

**REMARKS:**

```

100 REM * TRANSFER WAVEFORM FROM MEMORY 1 TO 4050 VARIABLES & ARRAY
110 REM * A= 7D20 GPIB DEVICE ADDRESS
120 REM *
130 REM * INPUT WAVEFORM PREAMBLE INTO VARIABLES:W,N,X,P,Z,Y
140 REM *           W=WAVEFORM NUMBER
150 REM *           N=POINTS/WAVEFORM
160 REM *           X=HORIZONTAL INCREMENT
170 REM *           P=POINT OFFSET
180 REM *           Z=VERTICAL ZERO
190 REM *           Y=VERTICAL MULTIPLIER
200 REM * INPUT WAVEFORM CURVE INTO ARRAY: C
210 REM *

```

**PROGRAM:**

```

220 ON SRQ THEN 420
230 A=20
240 S=0
260 PRINT @A:"DATA ENCODG: BIN"
270 PRINT @A:"DATA MEMORY: 1"
280 PRINT @A:"WFMP?"
290 INPUT @A:W,N,X,P,Z,Y
300 DIM C(N),I(9)
310 PRINT @A:"CURVE?"
320 WBYTE @A+64:
330 RBYTE I
340 RBYTE C,C0
350 WBYTE @95:
360 A1=SUM(C)+I(9)+I(8)
370 A1=A1-256*INT(A1/256)
380 A1=256-A1
390 IF ABS(C0)=A1 THEN 410
400 PRINT "CHECKSUM ERROR"
410 END
420 POLL D,L;A
430 RETURN

```

**COMMENTS:**

```

220 ENABLE SRQ HANDLER AT LINE 420
230 SET 7D20 GPIB DEVICE ADDRESS
240 CLEAR STATUS FLAG
260 REQUEST BINARY ENCODED DATA FROM THE 7D20
270 REQUEST WAVEFORM REGISTER 1 AS DATA SOURCE
280 REQUEST WAVEFORM PREAMBLE DATA
290 INPUT AND STORE PREAMBLE DATA IN VARIABLES
300 DIMENSION ARRAYS FOR CURVE DATA
310 REQUEST WAVEFORM CURVE DATA
320 ASSERT ATTENTION AT 7D20 ADDRESS
330 INPUT CURVE % AND BYTE COUNT
340 INPUT CURVE DATA AND CHECKSUM
350 ASSERT UNTALK AT 7D20 ADDRESS
360-400 COMPUTE CHECKSUM AND CHECK FOR ERROR
410 STOP
420 SERIAL POLL THEN 7D20: S=STATUS BYTE
430 RETURN FROM SRQ HANDLING ROUTINE

```

**5. Transfer Waveform Data to Controller—ASCII format.** In this program, it is assumed that ASCII strings P\$ and W\$ already contain waveform preamble and curve data.

**REMARKS:**

```

100 REM * TRANSFER WAVEFORM AND PREAMBLE FROM 4050 ASCII STRINGS
110 REM * TO 7D20 WAVEFORM REGISTER I
120 REM * WAVEFORM PREAMBLE IS IN STRING P$
130 REM * CURVE DATA IS IN STRING W$
140 REM * A= 7D20 GPIB DEVICE ADDRESS
    
```

**PROGRAM:**

```

150 A=20
160 ON SRQ THEN 210
170 PRINT @A: "DATA MEM: 1"
180 PRINT @A: P$, W$
200 END
210 POLL D, S; A
220 RETURN
    
```

**COMMENTS:**

```

150 SET 7D20 GPIB DEVICE ADDRESS
160 ENABLE SRQ HANDLER AT LINE 210
170 SPECIFY DATA MEMORY 1 AS DESTINATION
180 SEND THE PREAMBLE STRING P$
190 SEND THE CURVE STRING W$
200 STOP
210 SERIAL POLL THE 7D20; S=STATUS BYTE
220 RETURN FROM SRQ HANDLING ROUTINE
    
```

**6. Transfer Waveform Variables and Arrays to 7D20.** To transfer waveforms from the controller to the 7D20, the waveform preamble and curve information must be reconstructed to include the ASCII characters that were previously stripped away by the controller.

**REMARKS:**

```

10 REM * TRANSFER WAVEFORM AND PREAMBLE FROM 4850 VARIABLES
100 REM * AND ARRAY TO 7D20 WAVEFORM REGISTER #1
110 REM * A= 7D20 GPIB DEVICE ADDRESS
120 REM *
130 REM * WAVEPREAMBLE CONSISTS OF THE FOLLOWING:
140 REM *      W=WAVEFORM NUMBER
150 REM *      N=POINTS/WAVEFORM
160 REM *      X=HORIZONTAL INCREMENT
170 REM *      P=POINT OFFSET
180 REM *      Y=VERTICAL MULTIPLIER
190 REM *      Z=VERTICAL ZERO
200 REM * WAVEFORM CURVE DATA IS STORED IN ARRAY "C"
210 REM * C0 CONTAINS CHECKSUM
220 REM *

```

**PROGRAM:**

```

230 A=20
240 ON SRQ THEN 430
250 S=0
260 REM * BUILD WAVEFORM PREAMBLE
270 W$=STR(W)
280 N$=STR(N)
290 X$=STR(X)
300 P$=STR(P)
310 Y$=STR(Y)
320 Z$=STR(Z)
330 REM * SEND WAVEFORM PREAMBLE
340 PRINT @A:"WFMPRE WFID:W";W$;" , ENCDG: BINARY, NR. PT: ";N$;
350 PRINT @A:"WFMPRE PT.FMT:Y, XINCR: ";X$;" , PT.OFF: ";P$;
360 PRINT @A:"WFMPRE XZERO:0, YMULT: ";Y$;" , YZERO: ";Z$;" , BYT/NR: 1";
370 PRINT @A:"WFMPRE BN.FMT:LF, BIT/NR:8, CRVCHK:CHKSM0"
380 REM * SEND CURVE INFORMATION
390 WBYTE @A+64;
400 WBYTE @A+32: I
410 WBYTE @A+32: C, C0
420 WBYTE @63, 95:
430 POLL D, S; A
440 RETURN

```

**COMMENTS:**

```

230 SET 7D20 GPIB DEVICE ADDRESS
240 ENABLE SRQ HANDLER AT LINE 430
250 CLEAR STATUS FLAG
260-320 CONVERT VARIABLES TO STRING VARIABLES
340-370 CONSTRUCT AND SEND WAVEFORM PREAMBLE STRING
390 ASSERT ATTENTION AT 7D20 ADDRESS
400 SEND CURVE % AND BYTE COUNT TO 7D20
410 SEND CURVE AND CHECKSUM DATA TO 7D20
420 SET 7D20 TO UNTALK, UNLISTEN STATE
430 SERIAL POLL THE 7D20: S=STATUS BYTE
440 RETURN FROM SRQ HANDLING ROUTINE

```

**7. Query Routine.** In the query routine, the parameters to be checked are sent over the GPIB, followed by a question mark to indicate a query. The 7D20 will then respond with information to satisfy the query.

**REMARKS:**

```
100 REM * THIS ROUTINE RETURNS THE FRONT PANEL SETTINGS FROM
110 REM * THE 7D20 AND STORES THEM IN AN ASCII STRING.
130 REM * SETTINGS ARE STORED IN S#
140 REM * A= 7D20 GPIB DEVICE ADDRESS
```

**PROGRAM:**

```
150 A=20
160 ON SRQ THEN 210
170 DIM S$(600)
180 PRINT @A:"SET?"
190 INPUT @A:S#
195 PRINT S#
200 END
210 POLL D,S;A
220 RETURN
```

**COMMENTS:**

```
150 SET 7D20 GPIB DEVICE ADDRESS
160 ENABLE SRQ HANDLER AT LINE 210
170 DIMENSION STRING S#
180 REQUEST FRONT PANEL SETTINGS
190 INPUT AND STORE SETTINGS IN STRING S#
195 DISPLAY SETTINGS CONTROLLER
200 STOP
210 SERIAL POLL THE 7D20: S=STATUS BYTE
220 RETURN FROM SRQ HANDLING ROUTINE
```



**8. Implement and store 7D20 settings.** In this program, it is assumed that a group of settings have been stored in string S\$, using the SET? query. When set over the GPIB, these settings are automatically implemented, then stored in one of the six 7D20 memory locations.

**REMARKS:**

```
100 REM * THIS PROGRAM STORES A STRING OF FRONT PANEL SETTINGS
110 REM * IN ANY ONE OF THE 7D20'S 6 FRONT PANEL MEMORY LOCATIONS
130 REM * A= 7D20 GPIB DEVICE ADDRESS SETTINGS ARE STORED IN
140 REM * STRING S$
150 REM *
```

**PROGRAM:**

```
155 A=20
160 ON SRQ THEN 230
170 PRINT "IN WHICH MEMORY LOCATION DO YOU WANT TO"
180 PRINT "STORE THE SETTINGS(1-6)?"
190 INPUT L
200 PRINT @A:S$
210 PRINT @A:"STORE ",L
220 END
230 POLL D,S;A
240 RETURN
```

**COMMENTS:**

```
155 SET 7D20 GPIB DEVICE ADDRESS
160 ENABLE SRQ HANDLER AT LINE 230
170-180 PROMPT FOR USER INPUT
190 INPUT NUMBER OF 7D20 SETTINGS REGISTER
200 SEND FRONT PANEL SETTINGS STORED IN STRING S$
210 STORE THE SETTINGS IN THE DESIGNATED REGISTER
220 STOP
230 SERIAL POLL THE 7D20:S=STATUS BYTE
240 RETURN FROM SRQ HANDLING ROUTINE
```

**9. Query Functions.** This program permits the user to query a number of 7D20 functions, such as front-panel settings, cursor status, events, and acquire modes. A routine is included in the program that ensures that the query is legal.

**REMARKS:**

```

100 REM * THIS PROGRAM IS A GENERAL PURPOSE QUERY ROUTINE
110 REM * IT DETERMINES THE STATUS OF VARIOUS PARAMETERS OF
120 REM * THE 7D20 AND PRINTS THE RESPONSE OF THE INSTRUMENT
130 REM * TO ANY ONE OF A NUMBER OF QUERIES ON THE 4050
140 REM * GRAPHIC DISPLAY WHEN A NULL STRING IS INPUT, THE
150 REM * PROGRAM WILL TERMINATE.
160 REM * A= 7D20 GPIB DEVICE ADDRESS
    
```

**PROGRAM:**

```

170 A=20
180 ON SRQ THEN 330
190 PRINT "WHICH OF THE FOLLOWING QUERIES WOULD YOU LIKE TO MAKE?"
200 PRINT ""
210 PRINT "CH1,CH2,TRIG,HORIZ,DISP,CSW,AQR,CURS,COORD,DT,SRQ,EVENT,"
220 PRINT "ALLEV,ENGTY,ID,SET,HELP."
230 INPUT Q$
240 IF LEN(Q$)<1 THEN 320
250 C$=CHR(63)
260 Q$=Q$&C$
270 PRINT @A:Q$
280 INPUT @A:R$
290 PRINT R$
300 PRINT "NEXT QUERY"
310 GO TO 230
320 END
330 POLL D,S;A
340 IF S<>9B THEN 370
350 PRINT "ILLEGAL COMMAND"
360 GO TO 300
370 RETURN
    
```

**COMMENTS:**

```

170 SET 7D20 GPIB DEVICE ADDRESS
180 ENABLE SRQ HANDLER AT LINE 330
190-220 LIST AVAILABLE QUERIES ON GRAPHIC DISPLAY
230 INPUT DESIRED QUERY
240 IF NO QUERY REQUESTED, GO TO END
250 LOAD STRING C$ WITH A QUESTION MARK
260 ADD STRING C$ TO STRING Q$
270 SEND THE QUERY TO THE 7D20
280 INPUT THE RESPONSE TO THE QUERY
290 PRINT THE RESPONSE CODE ON THE GRAPHIC'S DISPLAY
300 PROMPT THE USER FOR THE NEXT QUERY
310 DO IT AGAIN
320 STOP
330 SERIAL POLL THE 7D20: S=STATUS BYTE
340 CHECK FOR ILLEGAL QUERY COMMAND
350 PRINT ERROR MESSAGE
360 LET THE USER ENTER A NEW QUERY
370 RETURN FROM SRQ HANDLER ROUTINE
    
```

**10. Print Poll Statement.** The controller will read the status byte generated by the 7D20 on an SRQ and print the appropriate phrase that describes the SRQ. In a 4050-series poll statement, one must specify the addresses of all the instruments that are to be polled. Should an instrument that is not polled generate an SRQ, that SRQ could prevent execution of the program until it is cleared by a poll statement.

**REMARKS:**

```
100 REM * THIS PROGRAM IS A BASIC SRQ HANDLER AND DECODER*
110 REM * FOR THE 7D20.
120 REM * A= 7D20 GPIB DEVICE ADDRESS
130 REM *
```

**PROGRAM:**

```
140 A=20
150 ON SRQ THEN 170
160 END
170 POLL D,S;A
180 PRINT "SRQ ORIGINATED FROM ADDRESS",D
190 IF S<>243 THEN 210
200 PRINT "STATUS REPORT: FATAL ERROR"
210 IF S=65 THEN 230
220 IF S<>81 THEN 240
230 PRINT "STATUS REPORT: POWER ON"
240 IF S<>83 THEN 260
250 PRINT "STATUS REPORT: EXECUTION ERROR"
260 IF S=99 THEN 280
270 IF S<>115 THEN 290
280 PRINT "STATUS REPORT: INTERNAL ERROR"
290 IF S=101 THEN 310
300 IF S<>117 THEN 320
310 PRINT "STATUS REPORT: EXECUTION WARNING"
320 IF S=102 THEN 340
330 IF S<>118 THEN 350
340 PRINT "STATUS REPORT: INTERNAL WARNING"
350 IF S<>66 THEN 370
360 PRINT "STATUS REPORT: OPERATION COMPLETE"
370 IF S=0 THEN 390
380 IF S<>16 THEN 390
390 PRINT "STATUS REPORT: NO STATUS TO REPORT"
400 RETURN
```

**COMMENTS:**

```
140 SET 7D20 GPIB DEVICE ADDRESS
150 ENABLE SRQ HANDLER AT LINE 170
160 STOP
170 SERIAL POLL THE 7D20: S= STATUS BYTE
180 PRINT ADDRESS OF SRQ ORIGINATOR
190 CHECK FOR FATAL ERROR STATUS FLAG
200 PRINT SRQ MESSAGE
210-220 CHECK FOR POWER ON STATUS FLAG
230 PRINT SRQ MESSAGE
240 CHECK FOR EXECUTION ERROR STATUS FLAG
250 PRINT SRQ MESSAGE
260-270 CHECK FOR INTERNAL ERROR STATUS FLAG
280 PRINT SRQ MESSAGE
290-300 CHECK FOR EXECUTION WARNING STATUS FLAG
310 PRINT SRQ MESSAGE
320-330 CHECK FOR INTERNAL WARNING STATUS FLAG
340 PRINT SRQ MESSAGE
350 CHECK FOR OPERATION COMPLETE STATUS FLAG
360 PRINT SRQ MESSAGE
370-380 CHECK FOR STATUS FLAG EQUAL TO ZERO
390 PRINT SRQ MESSAGE
400 RETURN FROM SRQ HANDLER ROUTINE
```

## 4050-Series Application Programs

**1. Event Capture.** This program sets the 7D20 for the capture of infrequent events. Each time a trigger is received by the 7D20, the input waveform is digitized, transferred over the GPIB to the controller, and stored on the controller magnetic tape. After each occurrence, the controller increments to the next tape file, and resets the 7D20 to a ready state to prepare for the next event. This program requires about 100K bytes of magnetic tape memory.

### REMARKS:

```

100 REM * THIS 7D20 APPLICATION PROGRAM IS DESIGNED TO ALLOW
110 REM * THE 7D20 TO OPERATE IN "BABYSIT" MODE.
120 REM * WHEN THE DEVICE RECEIVES A TRIGGER, THE DIGITIZED
130 REM * DATA IS STORED ON TAPE AND THE ORIGINAL SETUP
140 REM * IS DUPLICATED IN ORDER TO CAPTURE THE NEXT EVENT.
150 REM * THIS PARTICULAR PROGRAM STORES UP TO 6 EVENTS ON TAPE
160 REM * A= 7D20 GPIB DEVICE ADDRESS
170 REM *

```

### PROGRAM:

```

180 A=20
190 S=0
200 ON SRQ THEN 600
210 PAGE
215 REM FORMAT THE 4050 TAPE
220 PRINT "ENTER THE BEGINNING 4050 TAPE FILE NUMBER"
230 INPUT F
240 F1=F
250 FIND F
260 MARK 6,10000
270 FIND F
275 REM DIMENSION THE STRINGS
280 DIM S$(1000)
290 DIM P$(250)
300 DIM W$(7000)
305 REM WAIT FOR USER TO SETUP THE FRONT PANEL
310 PRINT "WHEN FRONT PANEL SETUP IS COMPLETE, PUSH RQS"
320 IF S<>83 THEN 320
330 PRINT @A:"TRIG HOLDN:ON"
335 REM BEGIN EXECUTION FOR UP TO SIX WAVEFORMS
340 FOR I=1 TO 6
350 PRINT @A:"TRIG HOLDN:ON"
355 REM WAIT FOR OPERATION COMPLETE SRQ FROM 7D20
360 IF S<>66 THEN 360
365 REM TRANSFER WAVEFORM AND PREAMBLE TO TAPE FILE
370 FIND F
380 PRINT @A:"DATA MEM: 1"
390 PRINT @A:"DATA ENCDG: ASC"
400 PRINT @A:"WFMPRE?"
410 INPUT @A:P$
420 PRINT @A:"CURVE?"
430 INPUT @A:W$
440 PRINT @33:P$,W$
450 F=F+1
460 PRINT "END OF CYCLE #",I
470 S=0
480 NEXT I
485 REM SEND WAVEFORMS BACK TO THE 7D20
490 PRINT "WHICH WAVEFORM DO YOU WISH TO VIEW(1-6)?"
500 INPUT W
510 F=F1+W-1
520 FIND F
530 INPUT @33:P$,W$
540 PRINT @A:P$
550 PRINT @A:W$
560 PRINT "DO YOU WISH TO SEE ANOTHER(Y OR N)?"
570 INPUT Q$
580 IF ASC(Q$)=89 OR ASC(Q$)=121 THEN 490
590 END
600 POLL D,S:A
610 RETURN

```

**COMMENTS:**

180 SET 7D20 GPIB DEVICE ADDRESS  
200 ENABLE SRQ HANDLER AT LINE 600  
220-270 FORMAT THE MAG TAPE  
280-300 DIMENSION THE STRINGS  
310-330 WAIT FOR USER TO SET UP 7D20  
370-480 WAIT WAVEFORM ACQUIRE AND DO WAVEFORM  
TRANSFER TO 4050 TAPE FILE  
485-590 ROUTINE TO SEND WAVEFORMS BACK TO 7D20  
600-610 SRQ HANDLING ROUTINE

**2. Store 7D20 settings on Magtape.** This 7D20 application program is designed to provide permanent storage for and easy user access to a variety of front panel settings. After the tape is formatted and the 7D20 GPIB device address is loaded from the 4050 keyboard, all control is turned over to the 7D20 itself. Settings are stored and recalled from the magnetic tape by using the numeric keys on the 7D20 front panel and by pushing probe identify buttons. This particular program stores up to 10 different 7D20 setups.

**REMARKS:**

```

100 REM ***** 7D20 Learn Program *****
110 REM * PROGRAM TO LEARN AND STORE FRONT PANEL SETTINGS
120 REM * FROM THE 7D20 ON THE 4050 MAG TAPE
130 REM * 7D20 ADDRESS IS "A"

```

**PROGRAM:**

```

140 A=20
150 WBYTE @20:
160 PRINT @A:"PID ON"
170 ON SRQ THEN 1200
180 PRINT "This program obtains front panel settings from the 7d20"
190 PRINT "using the set query. These settings are then stored on"
200 PRINT "the 4050 mag tape. Program control is asserted via the"
210 PRINT "7D20 'RQS #' button and the P6053B probe identify button."
220 PRINT "To begin, enter the number of the 4050 mag tape file"
230 PRINT "on which you would like to store the first 7D20 settings."
240 PRINT "-->";
250 INPUT F
260 G=F
270 DELETE S$
280 DIM S$(2000)
290 DIM M$(1000)
300 Q%=CHR(34)
310 M%=Q%
320 A%=CHR(13)
330 B%="MENU:"&A%
340 C%="1. STORE SETTINGS"&A%
350 D%="2. RECALL SETTINGS"&A%
360 V%="3. MENU"&A%
370 E%="TO SELECT, PUSH F,RQS,NUMBER"&A%
380 F%=Q%
390 Z%="CURRENT SETTING IS NUMBER-"
400 F%=F%&Z%
410 G%="TO STORE SETTING"&A%
420 H%="TO SEQUENCE THROUGH SETTINGS"&A%
430 I%="PUSH PROBE IDENTIFY BUTTON"&A%
440 M%=M%&B%
450 M%=M%&C%
460 M%=M%&D%
470 M%=M%&V%
480 M%=M%&E%
490 M%=M%&Q%
500 N%=Q%
510 N%=N%&G%
520 N%=N%&I%
530 N%=N%&Q%
540 O%=Q%
550 O%=O%&H%
560 O%=O%&I%
570 O%=O%&Q%
580 REM I PRINT MENU ROUTINE
590 PRINT @A:"TEXT",M%
600 S=0

```

PROGRAM (CONT)

PROGRAM (CONT)

```

610 IF S<>B3 THEN 610
620 PRINT @A:"EVENT?"
630 INPUT @A:P$
640 IF VAL(P$)=451 THEN 670
650 IF VAL(P$)=452 THEN 950
660 GO TO 620
670 REM * STORE SETTINGS ROUTINE
680 F$=SEG(F$,1,27)
690 PRINT "FORMATTING TAPE"
700 FIND F
710 MARK 10,2000
720 PRINT @A:"TEXT",N$
730 PRINT "READY"
740 FOR I=1 TO 10
750 PRINT @A:"EVENT?"
760 INPUT @A:P$
770 IF VAL(P$)<>459 THEN 790
780 GOSUB 1200
790 IF VAL(P$)=453 THEN 580
800 IF VAL(P$)<>457 THEN 750
810 PRINT @A:"SET?"
820 INPUT @A:S$
830 Z$=STR(I)
840 F$=F$&Z$
850 F$=F$&Q$
860 PRINT @A:"TEXT",F$
870 F$=SEG(F$,1,27)
880 P$=""
890 FIND F
900 PRINT @33:S$
910 F=F+1
920 NEXT I
930 F=F-10
940 GO TO 580
950 REM * ROUTINE TO RECALL SETTINGS
960 PRINT @A:"TEXT",0$
970 FIND G
980 F=G
990 FOR I=1 TO 10
1000 PRINT @A:"EVENT?"
1010 INPUT @A:P$
1020 IF VAL(P$)<>459 THEN 1040
1030 GOSUB 1200
1040 IF VAL(P$)=453 THEN 580
1050 IF VAL(P$)<>457 THEN 1000
1060 S=0
1070 FIND F
1080 INPUT @33:S$
1090 F=F+1
1100 PRINT @A:S$
1110 F$=SEG(F$,1,27)
1120 Z$=STR(I)
1130 F$=F$&Z$
1140 F$=F$&Q$
1150 PRINT @A:"TEXT",F$
1160 P$=""
1170 NEXT I
1180 F=F-10
1190 END
1200 POLL D,B;A
1210 RETURN

```

COMMENTS:

```

140 SET 7D20 GPIB DEVICE ADDRESS
150 SEND A DEVICE CLEAR TO THE 7D20
160 ENABLE PROBE IDENTIFY SRQ
170 ENABLE THE SRQ HANDLER AT LINE 450
180-260 PRINT INSTRUCTIONS AND INPUT FILE NUMBER
270-570 DIMENSION AND DEFINE ASCII STRINGS
580-660 ROUTINE TO PRINT MENU ON 7D20 AND INPUT RESPONSE
670-910 ROUTINE TO STORE 7D20 SETTINGS ON TAPE
920-1160 ROUTINE TO RECALL 7D20 SETTINGS FROM TAPE

```

**3. SRQ Decoding Routine.** This program is an expanded SRQ decoding routine. On SRQ, the 4050 will read the status byte generated by the 7D20. The program then goes one step further and interrogates the 7D20 as to the quantity and the nature of the events that produced the SRQ. An event query is sent to the 7D20 for each event code pending on the system. These event codes, when read by the 4050, are translated into phrases that describe the nature of the events. This ability is valuable when debugging 7D20 programs.

**REMARKS:**

```

2000 REM * THIS PROGRAM IS FOR ERROR MESSAGE DISPLAY FROM THE 7D20
2010 REM * A BASIC SRQ HANDLING ROUTINE IS EXPANDED TO PRINT THE
2020 REM * EVENT CODES ORIGINATING FROM THE 7D20.
2030 REM *
2040 REM * 7D20 IS DEVICE #20
2050 REM *
    
```

**PROGRAM:**

```

2060 ON SRQ THEN 2000
2070 END
2080 POLL D,S;20
2090 PRINT "SRQ ORIGINATED FROM ADDRESS",D
2100 IF S<>243 THEN 2120
2110 PRINT "STATUS REPORT: FATAL ERROR"
2120 IF S=65 THEN 2140
2130 IF S<>81 THEN 2150
2140 PRINT "STATUS REPORT: POWER ON"
2150 IF S<>83 THEN 2170
2160 PRINT "STATUS REPORT: EXECUTION ERROR"
2170 IF S=99 THEN 2190
2180 IF S<>115 THEN 2200
2190 PRINT "STATUS REPORT: INTERNAL ERROR"
2200 IF S=101 THEN 2220
2210 IF S<>117 THEN 2230
2220 PRINT "STATUS REPORT: EXECUTION WARNING"
2230 IF S=102 THEN 2250
2240 IF S<>118 THEN 2260
2250 PRINT "STATUS REPORT: INTERNAL WARNING"
2260 IF S<>66 THEN 2280
2270 PRINT "STATUS REPORT: OPERATION COMPLETE"
2280 IF S=0 THEN 1453
2290 IF S<>16 THEN 2310
2300 PRINT "STATUS REPORT: NO STATUS TO REPORT"
2310 PRINT @20:"EVQTY?"
2320 INPUT @20:N
2330 DIM A(N)
2340 PRINT @20:"ALLEV?"
2350 INPUT @20:A
2360 FOR Q=1 TO N
2370 P=A(Q)
2380 GOSUB 2410
2390 NEXT Q
2400 RETURN
2410 P=S-500
2420 IF P>0 THEN 2520
2430 P=S-400
2440 IF P>0 THEN 2540
2450 P=S-300
2460 IF P>0 THEN 2560
2470 P=S-200
2480 IF P>0 THEN 2580
2490 P=S-100
2500 GOSUB 2600
    
```

PROGRAM (CONT)



PROGRAM (CONT)

```

2510 RETURN
2520 GOSUB 3850
2530 RETURN
2540 GOSUB 3600
2550 RETURN
2560 GOSUB 3390
2570 RETURN
2580 GOSUB 2990
2590 RETURN
2600 REM * COMMAND ERROR SUBROUTINE
2610 IF P=8 THEN 2950
2620 IF P=9 THEN 2970
2630 P=P-50
2640 IF P<11 THEN 2670
2650 P=P-10
2660 GO TO 2690
2670 GOSUB P OF 2710,2730,2750,2770,2790,2810,2830,2850,2870,2890
2680 RETURN
2690 GOSUB P OF 2910,2930
2700 RETURN
2710 PRINT "SYMBOL OR NUMBER TOO LONG"
2720 RETURN
2730 PRINT "INVALID OR CONTROL CHARACTER INPUT"
2740 RETURN
2750 PRINT "INVALID SPECIAL CHARACTER AFTER ESCAPE"
2760 RETURN
2770 PRINT "INVALID NUMBER INPUT"
2780 RETURN
2790 PRINT "INVALID STRING INPUT"
2800 RETURN
2810 PRINT "SYMBOL NOT FOUND"
2820 RETURN
2830 PRINT "SYNTAX ERROR"
2840 RETURN
2850 PRINT "INVALID EDI"
2860 RETURN
2870 PRINT "INVALID DELIMITER"
2880 RETURN
2890 PRINT "EXPRESSION TOO COMPLEX"
2900 RETURN
2910 PRINT "EXCESSIVE BINARY CURVE POINTS"
2920 RETURN
2930 PRINT "EXCESSIVE ASCII CURVE POINTS"
2940 RETURN
2950 PRINT "CHECKSUM ERROR"
2960 RETURN
2970 PRINT "BYTECOUNT ERROR"
2980 RETURN
2990 REM* EXECUTION ERRORS ROUTINE
3000 IF P>40 THEN 3030
3010 GOSUB P OF 3070,3090,3110,3130,3150,3170
3020 RETURN
3030 P=P-49
3040 GOSUB P OF 3190,3210,3230,3250,3270,3290,3310,3330,3350,3370
3050 RETURN
3060 REM* PRINT ROUTINE
3070 PRINT "COMMAND NOT EXECUTABLE IN LOCAL"
3080 RETURN
3090 PRINT "SETTINGS LOST DUE TO RTL"
3100 RETURN
3110 PRINT "I/O BUFFERS FULL, OUTPUT DUMPED"
3120 RETURN
3130 PRINT "SETTINGS CONFLICTS"
3140 RETURN
3150 PRINT "ARGUMENT OUT OF RANGE"
3160 RETURN
3170 PRINT "GROUP EXECUTE TRIGGER IGNORED"
3180 RETURN
3190 PRINT "NOT IN HOLD ERROR"
3200 RETURN
3210 PRINT "ILLEGAL WAVEFORM #"

```

PROGRAM (CONT)

## PROGRAM (CONT)

```

3220 RETURN
3230 PRINT "ILLEGAL SETTINGS MEMORY"
3240 RETURN
3250 PRINT "ILLEGAL CURSOR NUMBER"
3260 RETURN
3270 PRINT "SETTINGS RECALL ERROR"
3280 RETURN
3290 PRINT "DISPLAY REFERENCE ERROR"
3300 RETURN
3310 PRINT "TURNING OFF CSW WAVEFORM DISPLAY ERROR"
3320 RETURN
3330 PRINT "ILLEGAL DATA MEMORY NUMBER"
3340 RETURN
3350 PRINT "ROLL MODE, AVE, ENV, NEGATIVE TRIGGER POSITION ERROR"
3360 RETURN
3370 PRINT "WAVEFORM PREAMBLE ILLEGAL NR.PT."
3380 RETURN
3390 REM# INTERNAL ERRORS ROUTINE
3400 IF P=1 THEN 3500
3410 IF P=2 THEN 3520
3420 IF P=31 THEN 3540
3430 IF P=32 THEN 3560
3440 IF P=95 THEN 3580
3450 P=P-30
3460 IF P=64 THEN 3580
3470 PRINT "SELFTEST FAILURE MODULE",P
3480 RETURN
3490 REM# PRINT ROUTINE
3500 PRINT "INTERRUPT FAULT"
3510 RETURN
3520 PRINT "SYSTEM ERROR"
3530 RETURN
3540 PRINT "SELFTEST FAILURE MODULE 1"
3550 RETURN
3560 PRINT "SELFTEST FAILURE MODULE 2"
3570 RETURN
3580 PRINT "SELFTEST FAIL ERROR"
3590 RETURN
3600 REM# SYSTEM EVENTS ROUTINE
3610 IF P=1 THEN 3700
3620 IF P=2 THEN 3720
3630 IF P=3 THEN 3740
3640 IF P=50 THEN 3760
3650 IF P=59 THEN 3830
3660 IF P=60 THEN 3810
3670 IF P>50 THEN 3780
3680 IF P=51 THEN 3780
3690 REM# PRINT ROUTINE
3700 PRINT "POWER ON"
3710 RETURN
3720 PRINT "OPERATION COMPLETE"
3730 RETURN
3740 PRINT "USER REQUEST (RQS KEY)"
3750 RETURN
3760 PRINT "HOLD AFTER AVEN, ENVN, OR HOLD NEXT"
3770 RETURN
3780 P=P-50
3790 PRINT "RQS",P
3800 RETURN
3810 PRINT "SELFTEST OPERATION COMPLETE"
3820 RETURN
3830 PRINT "SRQ PENDING"
3840 RETURN
3850 REM# EXECUTION WARNINGS
3860 P=P-49
3870 IF P<11 THEN 3900
3880 P=P-10
3890 GO TO 3920
3900 GOSUB P OF 3950,3970,3990,4010,4030,4050,4070,4090,4110,4130
3910 RETURN
3920 GOSUB P OF 4150,4170,4190,4210

```

PROGRAM (CONT)

PROGRAM (CONT)

```

3930 RETURN
3940 REM# PRINT ROUTINE
3950 PRINT "VOLTS/DIV OUT OF RANGE"
3960 RETURN
3970 PRINT "VERTICAL POSITION OUT OF RANGE"
3980 RETURN
3990 PRINT "TRIGGER LEVEL OUT OF RANGE"
4000 RETURN
4010 PRINT "TRIGGER POSITION OUT OF RANGE"
4020 RETURN
4030 PRINT "HORIZONTAL TIME/DIV OUT OF RANGE"
4040 RETURN
4050 PRINT "VXPD COMMAND OUT OF RANGE"
4060 RETURN
4070 PRINT "CSW POSITION OUT OF RANGE"
4080 RETURN
4090 PRINT "SET N VALUE OUT OF RANGE"
4100 RETURN
4110 PRINT "CURSOR POSITION OUT OF RANGE"
4120 RETURN
4130 PRINT "ASCII POINT OUT OF RANGE"
4140 RETURN
4150 PRINT "WFMPRE XINCR OUT OF RANGE"
4160 RETURN
4170 PRINT "WFMPRE PT.OFF OUT OF RANGE"
4180 RETURN
4190 PRINT "WFMPRE YMULT OUT OF RANGE"
4200 RETURN
4210 PRINT "WFMPRE YZERO OUT OF RANGE"
4220 RETURN

```

COMMENTS:

```

2060 ENABLE SRQ HANDLER AT LINE 2080
2070 STOP
2080 SERIAL POLL THE 7D20: S=STATUS BYTE
2090-2300 DETERMINE THE NATURE OF THE SRQ
2310-2350 REQUEST THE QUANTITY AND NATURE OF THE EVENT CODES
2360-2590 DO A SORT TO DETERMINE THE CATEGORY OF THE EVENT
2600-2980 PRINT COMMAND ERROR EVENT CODES
2990-3380 PRINT EXECUTION ERROR EVENT CODES
3390-3590 PRINT INTERNAL ERROR EVENT CODES
3600-3840 PRINT SYSTEM EVENT CODES
3850-4220 PRINT EXECUTION WARNING EVENT CODES

```

## 4041 Operating Programs

1. ASCII Waveform Transfer to controller. In this program, the selected 7D20 waveform memory contents are transferred in ASCII format to the 4041 controller.

```

100  Rem Program to receive waveform preamble, and curve data from the
110  ! 7D20, and put the data into variables. The waveform preamble is
120  ! stored in string 'WFPRE$', and the waveform data values are
130  ! stored in numeric array 'WFDATA'.
140  Integer wfnum,dev,stb,addr,nr,temp ! Establish integer variables.
150  Rem 7D20 ADDRESS IS SET TO 10
160  Dev=10 ! establish address for dev.
170  Open #1:"gpib0(pri="&str$(dev)&","com=<0>):" ! open channel #1 for
180  ! device address.
190  ! Set the End Of Message character to EOI only (works with either
200  ! terminator selection).
210  Rem Dimension string for waveform preamble storage, and also
220  ! to reserve buffer space for the curve data.
230  Dim wfpre$ to 200
240  On srq then call srqhdr ! establish linkage for SRQ handling.
250  Enable srq ! Enable SRQ handler.
260  Proc0: ! re-entrant point for selection of storage location.
270  Input prompt "Enter waveform location for transfer (1-6):":wfnum
280  If wfnum<1 or wfnum>6 then goto proc0 ! see if data is within
290  ! allowable range.
300  Rem Set up 7D20 to get data from DATA MEMORY selected, with ASCII
310  ! encoding of the data.
320  Input #1 prompt "data me:"&str$(wfnum)&","enc:asc;wfpre?":wfpre$
330  Rem Waveform preamble loaded into wfpre$.
340  Rem Get the number of points transmitted from the preamble.
350  Temp=pos(wfpre$,"NR.PT",1) ! Find where NR.PT is in the preamble.
360  Nr=valc(wfpre$,temp) ! Extract the number of points from wfpre$.
370  Delete var wfdata ! make sure variable is deleted prior to
380  ! dimensioning it.
390  Dim wfdata(nr) ! Dimension 'WFDATA' to hold curve data.
400  Input #1 dels ", " prompt "curve?":wfdata ! Input curve data
410  ! separated with commas.
430  Stop ! The variable "wfpre$" holds the preamble, and "wfdata" holds
440  ! the curve data at this time.
450  End
500  Sub srqhdr ! Serial poll handler sub-program.
510  Poll stb,addr;dev ! Poll device DEV.
520  Print "Status byte from device #";addr;" is ";stb ! Print Status
530  ! Byte.
540  Resume ! Return to main program.
550  End

```

2. ASCII Waveform Transfer to 7D20. In this program, the selected 4041 waveform data are transferred to the 7D20 in ASCII format.

```

100  Rem Program to send waveform preamble, and curve data to the 7D20
110  / from variables. The waveform preamble is stored in string 'WFMPRES'
120  / and the waveform data values are stored in numeric variable 'WFDATA'.
130  Integer dev,wfnum,stb,addr,nr,temp / Establish integer variables.
140  Rem 7D20 ADDRESS IS SET TO 10
150  Dev=10 / establish address for device.
160  Open #1:"gpib0(pri="&str$(dev)&","eom=<0>):" / Open channel #1 for
170  / device address. Set the End Of Message character to EOI only
180  / (works with either terminator selection).
190  On srq then call srqhdr / Establish linkage for SRQ handler.
200  Enable srq / Enable SRQ handler.
210  Rem The WFMPRES and WFDATA must already be defined (from previous
220  / program).
230  Proc0: /Re-entrant point for selection of storage location.
240  Input prompt "Enter waveform location for storage (1-6):":wfnum
250  If wfnum<1 or wfnum>6 then goto proc0 / See if data is within
260  / allowable range.
270  Rem Set up 7D20 to accept data for DATA MEemory selected, with ASCII
280  / encoding of the data.
290  Print #1:"da me:"&str$(wfnum)&","enc:asc",wfpre$ / Send preamble.
300  Rem Create image string for transfer of 'WFDATA' curve data.
310  Temppos(wfpre$,"NR.PT",1) / Find where NR.PT is in the preamble.
320  Nr=valc(wfpre$,temp) / Extract the number of points to be sent.
330  Image$="fa,x,512(fg,',')",&str$(nr-513)&"(fg,',')fg" / Create the
340  / image string to put commas after each number sent.
350  Print using image$ #1:"curve",wfdata / Send the curve data.
360  Stop / Curve has been restored to the 7D20.
370  End
400  Sub srqhdr / Serial poll handler sub-program.
410  Poll stb,addr;dev / poll device DEV.
420  Print "Status byte from device #";addr; is ";stb / Print Status
430  / Byte.
440  Resume / Return to main program.
450  End

```

**3. Binary Numeric Array Waveform Transfer to Controller.** This routine is used to send the currently selected waveform from the 7D20 to the controller. The waveform data is transferred in Binary format with the waveform preamble transferred in ASCII.

```

100  Res Program to receive a binary encoded waveform from the
110  ! 7D20, and put the data into a string, and numeric array.
120  ! The waveform preamble is stored in string 'WFMPRE$', and
130  ! the waveform numeric data is stored in variable 'WFDATA'.
140  Integer wfnum,dev,stb,addr,nr,temp ! Establish integer variables.
150  Res 7D20 ADDRESS IS SET TO 10
160  Dev=10 ! Establish address for device.
170  Open #1:"gp1b0(pri="&str$(dev)&","com=<0>):" ! Open channel
180  ! #1 for device address. Set the End Of Message to EOI only
190  ! (works with either terminator selection).
200  Res Dimension strings to hold waveform preamble, and to hold the
210  ! curve data.
220  Dim wfmpre$ to 200,curve$ to 1040 ! Preamble holds no more than
230  ! 200 characters, and longest curve data is 1024 + header, checksum,
240  ! and byte-count.
250  On srq then call srqhdlr ! Establish linkage for SRQ handler.
260  Enable srq ! Enable SRQ handler.
270  Proc0: ! Re-entrant point for selection of storage location.
280  Input prompt "Enter waveform location for transfer (1-6):":wfnum
290  If wfnum<1 or wfnum>6 then goto proc0 ! See if data is within
300  ! allowable range.
310  Res Set up the 7D20 to send data from the DATA Memory selected,
320  ! with binary encoding, sending waveform preamble first.
330  Input #1 prompt "da me:"&str$(wfnum)&","enc:bin;wfmp?":wfmpre$
340  Input #1 prompt "curve?":curve$ ! Now get the curve data.
350  Res Now find out how many data points have been sent.
360  Temp=pos(wfmpre$,"NR.PT",1) ! Locate the NR.PT header in WFMPRE$.
370  Nr=valc(wfmpre$,temp) ! This extracts the number from the NR.PT
380  ! link argument.
390  Temp=pos(curve$,"%",1) ! Find where the binary data starts.
400  Rep$(curve$,1,temp-1)=" ! Get rid of the header prior to decoding.
410  Delete var wfdata ! Delete variable before dimensioning it.
420  Res Dimension the variable to hold the numeric data.
430  Integer wfdata(nr) ! Integer, and dimension at the same time.
440  Res Convert the binary data to numeric array.
450  Getmem buffer curve$ using "8X":wfdata ! Decode the binary data.
460  Stop
470  End
500  Sub srqhdlr ! Serial poll handler sub-program.
510  Poll stb,addr,dev ! Poll device DEV.
520  Print "Status byte from device #";addr;" is ";stb ! Print Status
530  ! Byte.
540  Resume ! Return to main program.
550  End

```

**4. Binary Numeric Array Waveform Transfer to 7D20.** This routine is used to send the currently-selected waveform from the 7D20 to the controller. The waveform data is transferred in Binary format with the waveform preamble transferred in ASCII.

```

100  Rem Program to send a binary encoded waveform to the 7D20
110  / from the 4041. The waveform preamble comes from a string
120  / variable; the curve data comes from a numeric array.
130  / The variable 'WFNPRES' contains the preamble, and the array
140  / 'WFDATA' contains the curve data.
150  Rem 7D20 ADDRESS IS SET TO 10.
160  Integer wfnum,dev,stb,addr,nr,temp / establish integer variables.
170  Dev=10 / Establish address for 7D20.
180  Open #1:"gpib0(pri=%str$(dev)&,"eom=<0>):" / Open channel #1
190  / for device address. Set the End Of Message to EOI only (works
200  / with either terminator selection).
210  Rem Variables must already be defined from previous program (or
220  / be recalled from tape data file).
230  On srq then call srqhdlr / Establish linkage for SRQ handler.
240  Enable srq / Enable SRQ handler.
250  Proc0: / re-entrant point for selection of storage location.
260  Input prompt "Enter waveform location for transfer (1-6):":wfnum
270  If wfnum<1 or wfnum>6 then goto proc0 / See if data within
280  / allowable range.
290  Rem Set up the 7D20 to receive data into the DATA MEMORY selected,
300  / With binary encoding.
310  Print #1:"data:":&str$(wfnum)&,"enc:bin",wfmpres / Send preamble.
320  Putmem buffer curve$ using "fax8%": "curve",wfdata / Create curve$.
330  Print #1:curve$ / Send the curve data.
340  Stop
350  End
400 Sub srqhdlr / Serial poll handler sub-program.
410  Poll stb,addr;dev / Poll device DEV.
420  Print "Status byte from device #:;addr;" is ";stb / Print Status
430  / Byte.
440  Resume / Return to main program.
450  End

```

**5. Input 7D20 Front-panel Settings from String Array.** This routine is used to transfer one set of front-panel settings from the controller to the 7D20.

```

100  Re» Program to send front-panel settings to the 7D20.  The
110  / settings come from the string 'FPSET$'.
120  Integer dev, stb, addr / Establish integer variables.
130  Re» 7D20 ADDRESS IS SET TO 10.
140  Dev=10 / Establish address for device.
150  Open #1:"gpib0(pri="&str$(dev)&","eom=<0>):" / open channel #1 for
160  / device address.  Set End Of Message character to EOI only (works
170  / with either terminator selection).
180  On srq then call srqhdr / Establish linkage for SRQ handling.
190  Enable srq / Enable SRQ handler.
200  Re» The variable 'FPSET$' comes from previous program.
210  Print #1:fpset$ / Send front-panel settings to 7D20.
220  Stop / Front-panel now reset to original settings.
230  End
300 Sub srqhdr / Serial poll handler sub-program.
310  Poll stb, addr; dev / KPoll device DEV.
320  Print "Status byte from device #"; addr; " is "; stb / Print Status
330  / Byte.
340  Resume / Return to main program.
350  End

```



**6. Output 7D20 Front-panel Settings to String Array.** This routine is used to transfer one set of front-panel settings from the 7D20 to the controller.

```

100  Rem Program to receive front-panel settings from 7D20, and
110  / put the settings in string 'FPSET#'.
120  Integer dev, stb, addr / Establish integer variables.
130  Rem 7D20 ADDRESS IS SET TO 10.
140  Dev=10 / Establish address for device.
150  Open #1:"gpib0(pri="&str$(dev)&",eom=<0>);" / open channel #1 for
160  / device address. Set End Of Message character to EOI only (works
170  / with either terminator selection).
180  Rem Dimension string 'FPSET#' to hold front-panel settings.
190  Dim fpset$ to 600 / Setting information is less than 600 characters.
200  On srq then call srqhdlr / Establish linkage for SRQ handling.
210  Enable srq / Enable SRQ handler.
220  Input #1 prompt "set?":fpset$ / Ask for settings, and store them.
230  Stop / Information now in 'FPSET#'.
240  End
300 Sub srqhdlr / Serial poll handler sub-program.
310  Poll stb, addr; dev / KPoll device DEV.
320  Print "Status byte from device #"; addr; " is "; stb / Print Status
330  / Byte.
340  Resume / Return to main program.
350  End

```

**7. Setting the 7D20 to Remote.** This routine disables the 7D20 front-panel keys. Thus, during remote operation, accidental changes to the front-panel settings are ignored.

```
100  Rem Routine to send the 7D20 to REMOTE-ONLY.
110  Rem 7D20 ADDRESS IS 10.
120  Integer dev / Establish integer variable.
130  Dev=10
140  Wbyte 110,atn(32+dev,un1) / Send 'LL0', MLA-10 to put in remote,
150  / then unlisten device. The 7D20 will be in REMOTE-ONLY at this
160  / time.
170  Rem End of routine - may be converted to a subroutine, or to a
180  / sub-program as desired.
190  End
```

**8. Setting the 7D20 to Local.** This routine returns the 7D20 to local control following its operation in "Remote Only".

```

100  Rem Routine to return the 7D20 to LOCAL. This routine uses the
110  / addressed command 'GTL' to send the device to LOCAL. If REM is
120  / released, all devices will return to local, not just the addressed
130  / devices as when using 'GTL'.
140  Rem 7D20 ADDRESS IS 10.
150  Integer dev / Establish integer variable.
160      Dev=10
170  Mbyte gtl(dev),atn(uni) / Send the 'GTL' command, then unlisten the
180  / device, as the gtl function listen addresses the device's to be
190  / sent to local. The 7D20 will be in LOCAL at this time.
200  Rem End of routine - may be converted to a subroutine, or to a
210  / sub-program as desired.
220  End

```

**9. Text Transfer: 7D20 to Tape.** Using this routine, the 7D20 sends all displayed readout information and text from the 7D20 to the controller for storage on the magnetic tape.

```

100  Re» Program to transfer readout information, and any screen text on
110  ! the 7D20 screen to the controller, and have that information
120  ! stored on the internal magnetic tape.
130  Integer dev, stb, addr ! Establish integer variables.
140  Re» 7D20 ADDRESS IS SET TO 10.
150  Dev=10
160  Open #1:"gpib0(pri="%str$(dev)&",eom=<0>):" ! Open channel #1 for
170  ! device address.
180  Open #2:"text(ope=rep,siz= 500)" ! Open channel #2 for storage of
190  ! readout and text information.
200  On srq then call srqhdr ! Establish linkage for SRQ handler.
210  Enable srq ! Enable SRQ handler.
220  Re» Dimension string variable to hold all readout and text data.
230  Dim rdout$ to 500
240  Input #1 prompt "rdo?;text?":rdout$ ! Ask for, and receive readout &
250  ! text data.
260  Print #2:rdout$ ! Store the data on magnetic tape file 'TEXT'.
270  Close all
280  Stop
290  End
400 Sub srqhdr ! Serial poll handler sub-program.
410  Poll stb,addr;dev ! Poll device DEV.
420  Print "Status byte from device #";addr;" is ";stb ! Print Status
430  ! Byte.
440  Resume ! Return to main program.
450  End

```

**10. Text Transfer: Tape to 7D20.** Using this routine, the controller inputs the text information from the tape and sends it to the 7D20. Only the text information is sent to the 7D20; the readout information cannot be sent.

```

100  Rem Program to transfer information from magnetic tape to the screen
110  / of the 7D20. The data that is on tape, contains both the readout,
120  / and text data. All the information is sent to the 7D20, which will
130  / only respond to the text data. The readout data will be ignored.
140  Integer dev, stb, addr / Establish integer variables.
150  Rem 7D20 ADDRESS IS SET TO 10.
160  Dev=10
170  Open #1:"gpib0(pri="&str$(dev)&","com=<0>):" / Open channel #1 for
180  / device address.
190  Open #2:"text" / open channel #2 for retrieval of the
200  / readout and text information.
210  On srq then call srqhdlr / Establish linkage for SRQ handler.
220  Enable srq / Enable SRQ handler.
230  Rem Dimension string variable to hold all readout and text data
240  / on the magnetic tape file.
250  Dim rdout$ to 500
260  Input #2:rdout$ / Receive readout and text data from magnetic tape.
270  Print #1:rdout$&chr$(13); / Send the data to the 7D20.
280  If not(eof(2)) then goto 260
290  Close all
300  Stop / The text information is now back on the screen of the 7D20.
310  End
400 Sub srqhdlr / Serial poll handler sub-program.
410  Poll stb, addr; dev / Poll device DEV.
420  Print "Status byte from device #"; addr; " is "; stb / Print Status
430  / Byte.
440  Resume / Return to main program.
450  End

```

**11. Text Generation: Controller to 7D20.** This program allows the user to input to the controller the desired message to send to the 7D20. After the message is input to the controller, the entire message is sent to the 7D20 and displayed on the crt.

```

100  Rem Program to allow generation of a text message for the 7D20
110  ! from the controller program. The general rule of thumb (100
120  ! character limit) will be used in this program.
130  Integer dev, stb, addr, total, i ! Establish integer variables.
140  Data "More than 100 characters, re-enter data from beginning."
150  Read error$
160  Rem 7D20 ADDRESS IS 10.
170  Dev=10 ! Establish device address.
180  Open #1:"gpib0(pri="%str$(dev)&","com=<0>):" ! Open channel #1
190  ! for device address.
200  On srq then call srqhdlr ! Establish linkage for SRQ handler.
210  Enable srq ! Enable SRQ handler.
220  Rem Dimension string variable to hold characters for text display.
230  Dim text$ to 120, char$ to 36 ! Temporary storage of 36 characters
240  ! per line (7D20 maximum limit) in string 'CHAR$'.
250  Text$="" ! Pre-define 'TEXT$' with a "".
260  Rem Input up to 12 lines of code, and less than 100 printable
270  ! characters.
280  Proc0: total=0 ! Initialize character counter.
290  For i=1 to 12
300  Input prompt "-->" : char$
310  If len(char$)=0 then exit 1 to proc2
320  Total=total+len(char$)
330  If total<100 then goto proc1
340  Print error$
350  Exit 1 to proc0
360  Proc1: rep$(text$,len(text$)+1,0)=char$&chr$(13) ! Concatenate
370  ! strings, and add <CR> as line delimiter.
380  Next i
390  Proc2: ! Exit point from for-next loop
400  Text$=text$&" " ! Add final " " to text string for transfer.
410  Rem Now send the data to the 7D20 screen.
420  Print #1:"text";text$
430  Stop ! End of program, message delivered.
440  End
500  Sub srqhdlr ! Serial poll handler sub-program.
510  Poll stb,addr;dev ! Poll device DEU.
520  Print "Status byte from device #";addr; is ";stb ! Print Status
530  ! Byte.
540  Resume ! Return to main program.
550  End

```

## 4041 Application Programs

**1. SRQ Handler.** This routine decodes the 7D20 status byte and displays the actual status message. This is useful in the program development stage of software writing, where the programmer is thus not constrained to look up the error code meaning.

```

100  Rem Program to detect SRQ, poll device list, and decode 7D20 status
110  ! byte. Display actual status message, and the binary equivalent.
120  ! 'STB' is the device's reported status byte, and 'ADDR' is the
130  ! address of the device sending this particular stb.
140  !
150  !       < SRQ STATUS BYTE (STB) >
160  !
170  !   DIO #'S -> 8 7 6 5 4 3 2 1
180  !   STATUS CODING:
190  !   SYSTEM --> 0 R E B S S S
200  !
210  !   DEVICE --> 1 R D D D D D D
220  !
230  !       R = RQS BIT, E = ERROR BIT, B = BUSY BIT
240  !       S = SYSTEM STATUS, D = DEVICE STATUS
250  !
260  !
270  ! This program may be treated, as a whole, as a sub-program with all
280  ! variables local to the sub-program. Alternatively, it may be used
290  ! as a subroutine within the mainline program.
300  ! Integer stb,addr,pos1,pos2,errbit,devbit,dev ! Establish integers.
310  ! Dim stb*(2) to 150 ! Dimension strings.
320  Rem 7D20 ADDRESS IS 10
330  Dev=10
340  Serpol: ! Here is where the program really starts.
350  Stb*(1)=";POWER ON;OPERATION COMPLETE;USER REQUEST;REQUEST CONTROL
;PASSED CONTROL;"
360  Stb*(2)=";COMMAND ERROR;EXECUTION ERROR;INTERNAL ERROR;POWER FAIL
;EXECUTION WARNING;"
370  Poll stb,addr;dev ! serial poll device.
380  If (stb band 128)=128 then goto device ! Branch if device bit set.
390  If (stb band 32)=32 then errbit=2 else errbit=1 ! Error?
400  Goto decode1
410 Device: ! Message is device dependent
420  Gosub liststb
430  Print "Device dependent status reported from device #";addr;".
440  Print "Refer to device's manual for definition."
450  Resume
460 Decode1: ! Decode error or normal message and list it.
470  Gosub liststb
480  Stb=stb band 31 ! Remove all but busy and code bits.
490  If (stb band 16)=16 then print "Device presently busy" else goto 510
500  Stb=stb-16 ! Get rid of busy bit if set.
510  Pos1=posn(stb*(errbit),";",1,stb) ! Find start of message.
520  Pos2=posn(stb*(errbit),";",2,stb) ! Find end of message.
530  Decode%=seg$(stb*(errbit),pos1+1,pos2-pos1) ! Get decoded message.
540  Print "Device's status message is ";decode%
550  Resume
560 Liststb: ! List the status byte, and binary equivalent.
570  Putmem buffer decode% using "8b":stb
580  Rep$(decode%,5,0)=" " ! put in a space for readability.
590  Print "Status byte ";stb;" reported. Binary is ";decode%
600  Return ! Return to main program.
610  End

```

**2. Store and Recall Front-panel Settings.** This routine allows the user to generate up to six different front-panel settings and store them on the controller magnetic tape. The user is given a menu display that allows selection of the desired operations. The program advances by front-panel key closures and the use of the probe identify button, located at the probe tip.

```

100 Rem Program to Store and Recall Front-panel Settings. This routine
110 / allows the user to generate up to ten different front-panel
120 / settings, and store them on the controller magnetic tape.
130 / The front-panel settings are stored when the operator presses the
140 / 'probe identify' button on the probe. Each successive press, will
150 / store another front panel setting. When all 6 are defined, further
160 / operation of the 'probe identify', will cause the front panel
170 / settings to be recalled, and sent back to the 7D20.
180 Integer dev, stb, addr, setnum, selnum / Establish integer variables.
190 Rem 7D20 ADDRESS IS 10
200 Dev=10 / Establish device address.
210 Dim fpsets$(10) to 600, menus to 200 / Dimension strings.
220 Fpsets$="" / Initialize fpsets$ string array.
230 Open #1:"gpib0(pri)="&str$(dev)&"&com=<0>):" / Open channel #1 for
240 / device address. Set End Of Message to EOI only, works with any
250 / selected terminator.
260 On srq then call srqhdlr / Establish linkage for SRQ handling.
270 Enable srq / Enable SRQ handler.
280 Whyte sdc(dev) / Clear input/output buffers, and any existing SRQ's
290 / and error conditions. (selected device clear)
300 Print #1:"pid on" / Enable Probe ID rqs bit.
310 Gosub menu / List menu to screen of 7D20.
320 Wait1: wait / Wait for next interrupt.
330 Routine1: goto selnum of menu, store, recall, tstore, rstore
340 Goto wait1 / Wait for user generated SRQ.
350 Menu: / Routine to send user menu to 7D20
360 Menu=chr$(13)&chr$(13)&chr$(9)&"menu:"&chr$(13)
370 Menu=menu&chr$(13)&chr$(9)&"1. sto ft pnl setting"&chr$(13)
380 Menu=menu&chr$(9)&"2. rcl ft pnl setting"&chr$(13)
390 Menu=menu&chr$(9)&"3. sto settings on tape"&chr$(13)
400 Menu=menu&chr$(9)&"4. rcl settings from tape"&chr$(13)&chr$(13)
410 Menu=menu&chr$(9)&"e.g. press 'f rqs 1' for"&chr$(13)
420 Menu=menu&chr$(9)&chr$(9)&"store routine."
430 Print #1:"text";chr$(34)&menu&chr$(34) / Send 'TEXT "menu"'
440 Goto wait1 / return to wait for user SRQ.
450 Store: / Routine to acquire settings from 7D20
460 Setnum=1
470 Store1: gosub prompt1 / Send prompt to screen of 7D20
480 Wait2: wait / Wait for user generated SRQ for this routine.
490 If selnum=1 then goto routine1 / If menu, goto select point.
500 If selnum<>5 then goto wait2 / If probe id, goto store settings.
510 Input #1 prompt "set?":fpsets$(setnum) / Input complete front panel
520 / settings from device, put in array element 'setnum'.
530 Setnum=setnum+1 / Increment array element counter.
540 If setnum<11 then goto store1 / Loop back if less than 11 settings.
550 Selnum=1 / Select menu
560 Goto routine1 / Go to operation select point.
570 Recall: / Routine to send settings to 7D20
580 Setnum=1 / Select menu.
590 Recall1: gosub prompt1 / Print prompt on 7D20 screen.
600 Print #1:fpsets$(setnum) / Sent complete front panel settings to
610 / device from the array element 'setnum'.
620 Setnum=setnum+1 / Increment array element counter.
630 If setnum<11 then goto wait3 / Loop if all ten have not been sent.
640 Selnum=1 / Select menu.
650 Goto routine1 / Go to operation selection point.
660 Wait3: wait / Wait for probe user generated SRQ.
670 If selnum=1 then goto routine1
680 If selnum<>5 then goto wait3
690 Goto recall1
700 Tstore: / Tape storage routine
710 Open #2:"fpsets(ope=rep,siz=6000)" / Open channel to write to tape
720 / file. The REP command will either new, or replace existing file.

```

PROGRAM (CONT)



PROGRAM (CONT)

```

730 Print #2 using "10(fa/):fsets" / Write to the tape placing <CR>
740 / after each array element to allow easy retrieval of information.
750 Close #2 / Close the channel, update tape directory.
760 Selnum=1 / Select menu.
770 Goto routine1 / Return to operation select point.
780 Restore: / Load settings from tape file
790 Open #2:"fsets" / Open channel to read/write data to tape file.
800 Input #2:fsets / Input data from tape file (each array element is
810 / delimited with a <CR>.
820 Close #2 / Close the channel to tape file.
830 Selnum=1 / Select menu
840 Goto routine1 / Return to operation select point.
850 Prompt1: / Send prompt message to 7D20 - build text string first.
860 Menu="current settings number is "&str$(setnum)&chr$(13)
870 Menu=menu&chr$(13)
880 Menu=menu&"press probe identify to advance"&chr$(13)
890 Menu=menu&"press 'rqs' to return to menu."
900 Print #1:"text";chr$(34)&menu&chr$(34) / Send menu as text.
910 Return / Return from subroutine.
920 End
1000 Sub srqhdlr local evnt / Serial poll handler sub-program
1010 Integer evnt
1020 Poll stb,addr;dev / Poll device DEV.
1030 If stb=83 then goto event / User request status byte.
1040 Print "Status byte from device #";addr;" is ";stb / Print Status
1045 / Byte.
1050 Input #1 prompt "event?":evnt
1060 Print "Event";str$(evnt);"occured."
1070 Resume / Return mechanism to main program.
1080 Event: / send an event?, get the response, and continue according to
1090 / the event that occurred.
1100 Input #1 prompt "event?":evnt
1110 If evnt=403 then selnum=1
1120 If evnt=451 then selnum=2
1130 If evnt=452 then selnum=3
1140 If evnt=453 then selnum=4
1150 If evnt=454 then selnum=5
1160 If evnt=457 or evnt=458 then selnum=5
1170 Resume / return from the srq handler routine.
1180 End

```

**3. Event Capture.** Sometimes called a "babysit routine", this program allows the user to set up the 7D20 as desired, then have a user-selected number of waveforms acquired and stored on the controller magnetic tape. This allows the user to leave the system unattended, but still capture the required data.

```

100  Rem Program to capture and record events that trigger the 7D20.
110  ! This program (sometimes called a 'babysit routine' allows the
120  ! user to set up the 7D20 as desired, then have a user-selected
130  ! number of waveforms acquired, and stored on the controller's
140  ! magnetic tape. This allows the user to leave the system un-
150  ! attended, but still capture the required data.
160  Rem Routine awaits a 'hold next complete' SRQ from the 7D20.
170  ! When the proper SRQ is received, the waveform is transferred
180  ! to the controller, then stored on tape.
190  Integer dev, stb, addr, cycnum, datame, numwf, evt, siz, numcyc
200  Rem 7D20 ADDRESS IS 10
210  Dev=10 ! Establish device address.
220  Open #1:"gpiB0(pri="&str$(dev)&","eos=<0>);" ! Open channel #1 for
230  ! device address. Set the End Of Message terminator to EOI only
240  ! (works with any termination selection).
250  On srq then call srqhdlr ! Establish linkage to SRQ handler routine
260  Enable srq ! Enable SRQ handler.
270  Rem Dimension strings to appropriate length.
280  Dim fpset$ to 600,wfmpre$ to 200,text$ to 100
290  Data 1,1,1
300  Tab%=chr$(9)
310  C%=chr$(13)
320  Quot%=chr$(34)
330  Read cycnum,datame,numwf
340 Proc0: ! Give user prompts on screen of 7D20, and get responses
350 ! from the 7D20 user keys.
360 Call prom1
370 Wait ! Wait for user input, if not what expected, erase screen, and
380 ! write it out again.
390 If evnt<1 or evnt>6 then goto proc0
400 Proc1: numcyc=evnt ! Set cycle counter, and tell user to set up 7D20
410 Call prom2
420 Wait
430 If evnt=10 then goto proc2
440 Goto proc1
450 Proc2: ! Get settings from 7D20, and decode whether one or two
460 ! channels in use.
470 Disable srq
480 Input #1 prompt "aqr? mod":aqr$
490 If pos(aqr$,"both",1)>0 then numwf=2 ! Number of waveforms for
495 ! file size.
500 If pos(aqr$,"ch2",1)>0 then datame=2
510 Cycnum=1 ! Initialize cycle counter
520 Siz=6000*numwf ! Calculate space for waveform data on tape.
530 Proc3: ! Start of babysitting routine, send message to screen.
540 Call prom5 ! Prompt message for 7D20.
550 Enable srq
560 Print #1:"trig hold:n" ! Re-arm the 7D20 for next trigger.
570 Wait ! Wait for hold-next interrupt from 7D20
580 Input #1 prompt "event?":evnt
590 If evnt=450 then goto proc4
600 Print "Invalid event code received - EVENT ";evnt
610 Goto proc3
620 Proc4: ! Proper status and error code received, continue processing.
630 Call prom3 ! Tell user that program is working on data.
640 Open#="wf- "&str$(cycnum)&"(ope=rep,siz="&str$(siz)&")"
650 Open #2:open$
660 Input #1 prompt "da me:"&str$(datame)&","enc:asc;wfmp?":wfmpre$
670 Call curve
680 If numwf=1 then goto closeit
690 Input #1 prompt "da me:2;wfmp?":wfmpre$
700 Call curve
710 Closeit: close 2 ! Close present tape file.
720 Cycnum=cycnum+1

```

PROGRAM (CONT)

## PROGRAM (CONT)

```

730   If cycnum>numcyc then goto done ! If number of cycles completed.
740   Goto proc3
750 Done: ! End of program, tell user that the waveforms have been stored.
760   Call proc4
770   End
800 Sub srqhdr ! Serial poll handler sub-program.
810   Poll stb,addr;dev ! Poll device DEU.
820   If stb=83 then goto 850
830   Print "Status byte from device #";addr;" is ";stb / Print Status
835   / Byte.
840   Resume ! Return mechanism to main program.
850 Event: ! Must be a user key request, decode response into evnt.
860   Input #1 prompt "event?":evnt
870   If evnt<451 or evnt>456 then goto special
880   Evnt=evnt-450 ! Decode event into number 1 to 6 corresponding to
890   / front panel user rqs keys.
900   Resume ! Return to mainline when done here.
910 Special: ! Not a user rqs key, see if normal user key. If not, do
920   / an event? and send the data to controller screen.
930   If evnt=403 then evnt=i0 else goto 950
940   Resume ! Rqs key was pressed, return to main program.
950   Print "EVENT ";evnt;" reported under status byte 83."
960   Resume ! Return to mainline.
970   End
1000 Sub curve local temp,nr ! Get curve data, then store wfmpref & wfdata
1010 / on the magnetic tape.
1020   Temp=pos(wfmpref,"nr.pt",1) ! Find where 'NR.PT' is in the preamble.
1030   Nr=valc(wfmpref,temp) ! Extract number of points from wfmpref.
1040   Delete var wfdata
1050   Dim wfdata(nr) ! Dimension 'WFDATA' to hold curve data.
1060   Input #1 dels ", " prompt "curve?":wfdata
1070   Print #2:wfmpref ! Put the waveform preamble on tape.
1080   Print #2:wfdata ! Put the curve data points on tape.
1090   Return
1100   End
1200 Sub clrscrn local cr$ ! Sub-program to clear the screen of the 7D20.
1210   Dim cr$(12) to 1
1220   Cr$=chr$(13)
1230   Image$=""text",x,fa,12(fa),fa" / Build the image for 'print using'.
1240   Print #1 using image$:chr$(34),cr$,chr$(34) ! Send "<>,<CR>,<>".
1250   Return / Return from whence it came.
1260   End
1300 Sub prom1 ! sub-program for 7D20 screen prompt #1.
1310   Call clrscrn ! Clear the 7D20 screen.
1320   Text$=quot&c&tab&"Enter number of cycles"&c&
1330   Text$=text&tab&"to store."&c&c&tab&
1340   Text$=text&"e.g. press 'f rqs 1' for"&c&tab&
1350   Text$=text&"one cycle."&quot&
1360   Print #1:"text";text$
1370   Return
1380   End
1400 Sub prom2 ! sub-program for 7d20 screen prompt #2.
1410   Text$=quot&c&tab&"Please configure the 7d20 as"&c&
1420   Text$=text&tab&"needed for capturing data."&c&c&
1430   Text$=text&tab&"press 'rqs' when ready."&quot&
1440   Print #1:"text";text$
1450   Return
1460   End
1500 Sub prom3 ! Sub-program for 7D20 screen prompt #3.
1510   Call clrscrn
1520   Text$=chr$(34)&c&c&tab&"working on waveform"&c&
1530   Text$=text&tab&" " &str$(cycnum)&". please wait."&quot&
1540   Print #1:"text";text$
1550   Return
1560   End

```

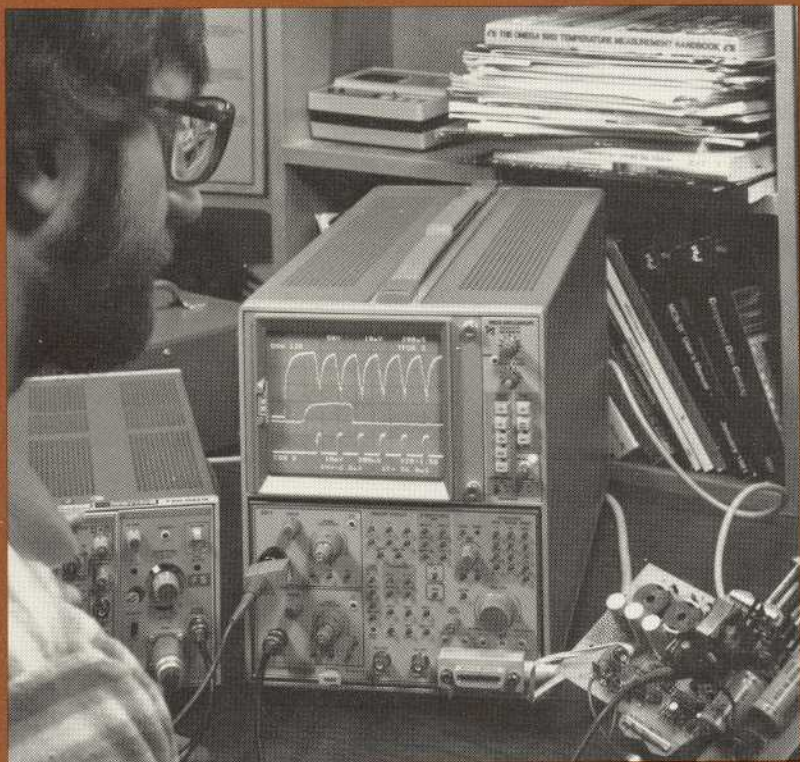
PROGRAM (CONT)

PROGRAM (CONT)

```

1600 Sub prom4 ! Sub-program for 7D20 screen prompt #4.
1610 Call clrscrn ! Clear the 7D20 screen.
1620 Text%=quot%c%&tab%&tab%"Program completed!"&c%
1630 Text%=text%&c%&tab%"waveforms stored on tape."&quot%
1640 Print #1:"text";text%
1650 Return
1660 End
1700 Sub prom5 ! Sub-jprogram for 7D20 screen prompt #5.
1710 Call clrscrn
1720 Text%=quot%&tab%"7D20 now babysitting!"&quot%
1730 Print #1:"text";text%
1740 Return
1750 End

```



In this section we give some examples of how the 7D20 can be used in specific applications. These applications are but a few of the many ways your 7D20 will reliably simplify your measurement needs. By bringing accurate digital storage and GPIB communications capabilities to the conventional oscilloscope mainframe, your 7D20 is a friendly partner in a whole new area of test measurements.

# SECTION 5 CONTENTS

---

APPLICATION 1—MONITORING SLOWLY CHANGING EVENTS WITH THE 7D20 AND 7D11 .....	5-1
INSTRUMENTATION SETUP #1 .....	5-2
APPLICATION 2—ULTRASONIC, NON-DESTRUCTIVE TESTING USING 7D20 .....	5-3
INSTRUMENTATION SETUP #2 .....	5-4
APPLICATION 3—MONITORING INTERCELLULAR NEURONAL DISCHARGE..	5-6
APPLICATION 4—MEASURING PULSE JITTER, FREQUENCY SHIFT, AND AMPLITUDE VARIATIONS USING THE 7D20 .....	5-7
APPLICATION 5—AC LOADLINE ANALYSIS USING THE 7D20 AND 7A13 .	5-9
INSTRUMENTATION SETUP #5 .....	5-10
CALIBRATION .....	5-11
OPERATION .....	5-11
APPLICATION 6—USING THE 7D20 WITH THE 7854 FOR WAVEFORM PROCESSING .....	5-13
INSTRUMENTATION SETUP #6 .....	5-14
MANUAL TRANSFER .....	5-15
COMPUTER CONTROLLED TRANSFER .....	5-15

# APPLICATIONS

---

## APPLICATION 1 MONITORING SLOWLY CHANGING EVENTS WITH THE 7D20 AND 7D11

Monitoring events or phenomena which change at incredibly slow rates such as  $10^{-4}$  Hz, is a difficult task for conventional oscilloscopes. The first problem arises in not being able to sweep the trace slowly enough. The second is producing a comprehensive waveform which represents the changes.

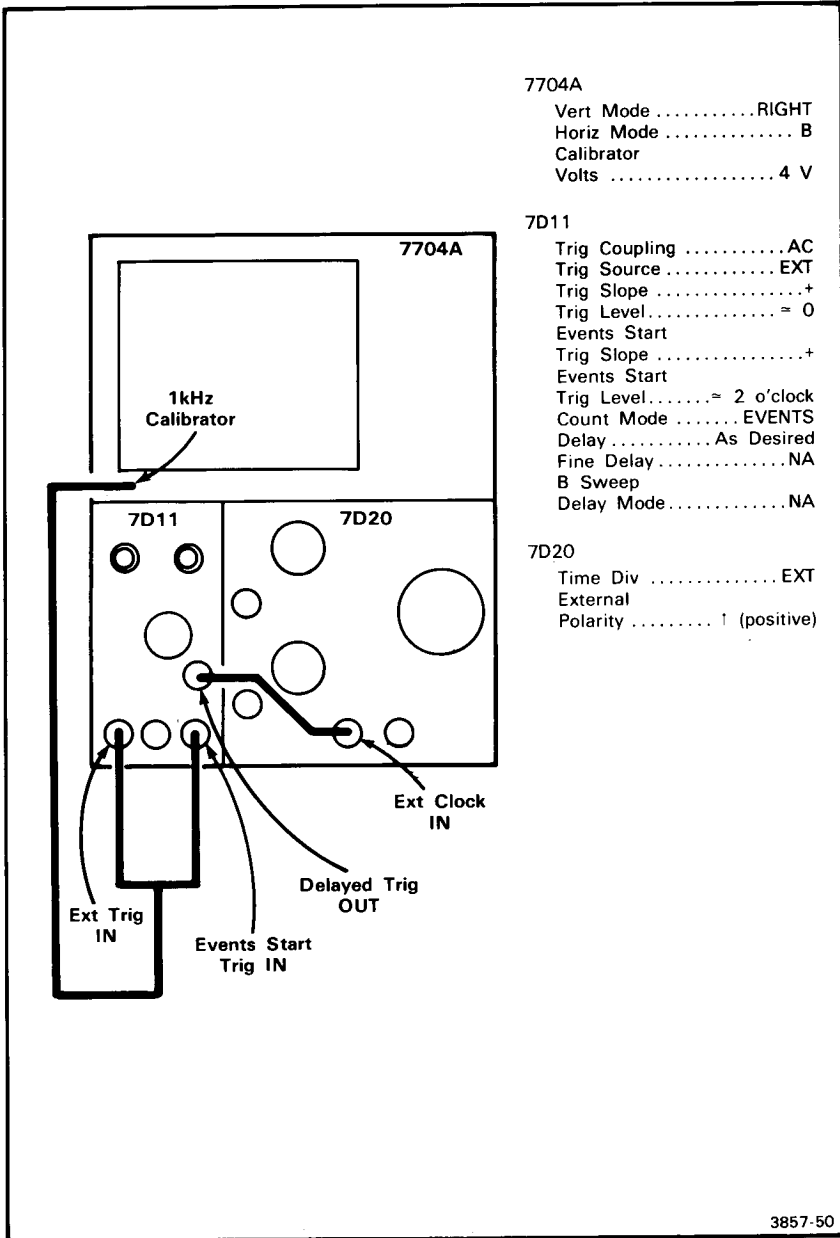
By using the 7D20 to digitize and store samples of the event, and a Tektronix 7D11 to serve as a calibrated source supplying the very slow timing (sampling information for the 7D20, both of these problems may be overcome. The 7D11 is a digitally controlled delaying unit with the capability to delay by Events or Time. For this application it will be used as a delay by event device or, simply put, a "divide by n counter". The 1-kHz calibrator signal provided by the 7704A mainframe can be divided down by 1 to  $10^7$  in steps of one. This is selectable at the 7D11 front panel. This "divided down" signal is obtained at the DLY'D TRIG OUT connector on the 7D11 and applied to the EXT CLOCK input of the 7D20. You can control the sampling interval for data acquisition by varying the EVENT DELAY count displayed on the crt of the 7704A.

Since the calibrator has a 1 ms period, the crt readout of EVENT DELAY may be directly read as N milliseconds per sample point. This provides a range from 1 ms/sample to  $10^4$  sec/sample. So a 1024 point record at the slowest speed represents 115.7 days of information. Obviously, long term stability of this system may need to be considered for such a rare occasion.

It should be pointed out that the ENV and AVE continuous functions of the 7D20 work in a similar manner to that when ROLLing. The total number of samples processed between displayed points is fixed at 100 when externally clocking the 7D20. It has the effect of further dividing the displayed time per point by 100.

The use of such a set-up is especially useful for long term monitoring of temperature and pressure, fluid levels, chemical reactions, and seismic activity.

## INSTRUMENTATION SET-UP #1



3857-50



## APPLICATION 2 ULTRASONIC, NON-DESTRUCTIVE TESTING USING 7D20

Non-destructive testing of material for uniformity is typically performed using ultrasonic techniques. Frequencies in the ultrasonic range make possible the resolution necessary to measure material thickness and locate flaws. An inspector using ultrasonic equipment can "see" beneath the surface of materials to locate even minute flaws.

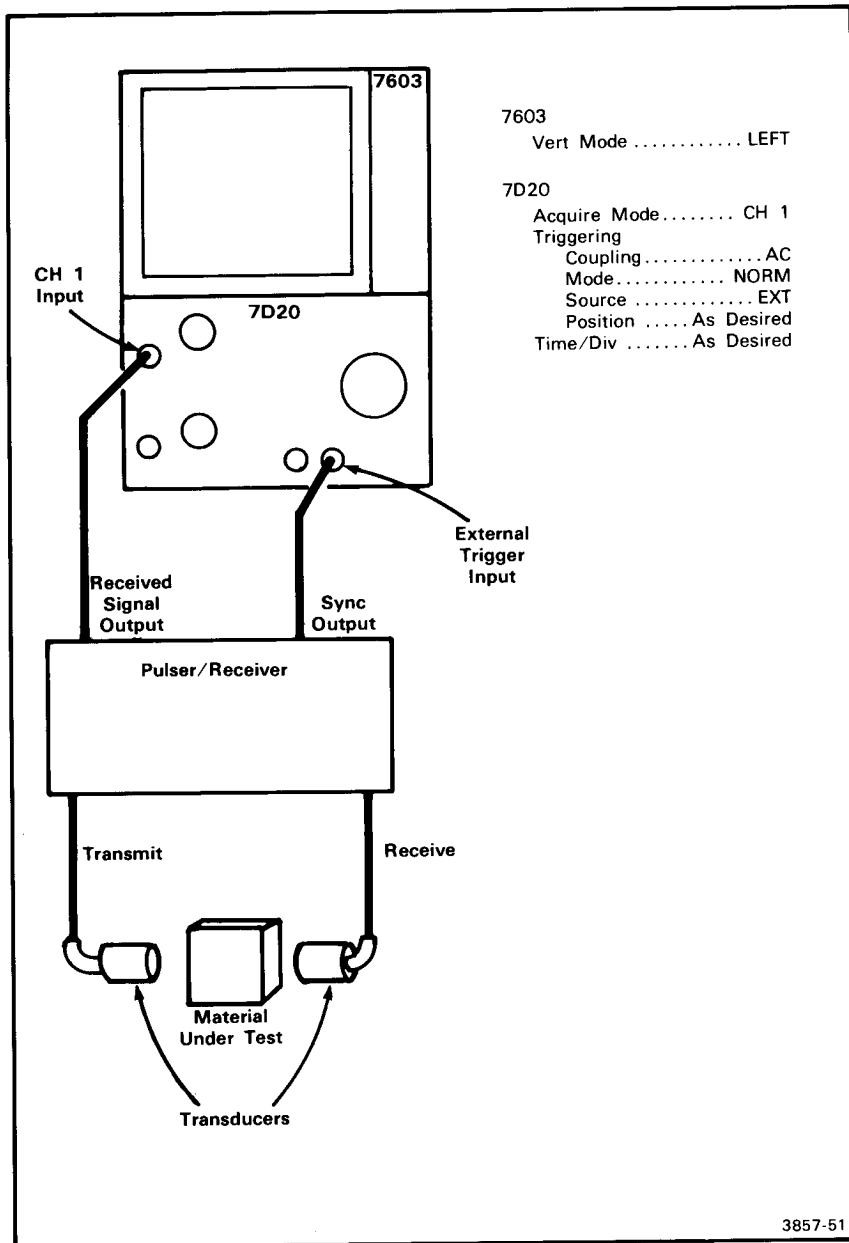
To do this, quartz crystals are often used as transducers to convert an electrical pulse from a transmitter to ultrasonic waves and, after the ultrasonic burst has traveled through the material, convert it back to an electrical signal at the receiver end. The crystals are chosen to resonate far above the audio range, often in the megahertz region. The pulse repetition rate is chosen low enough so the pulse and any reflections reach the receiver before the next pulse is sent. Water is often used as a transfer medium, with the crystals and part under test placed in a tank.

Two test modes are common—pulse/echo and pitch/catch. Pulse/echo is similar to radar. A single crystal is used for both transmitting and receiving. In a pitch/catch mode, two crystals are used—one transmits and the other receives, often on the other side of the object from the transmitting crystal. The transit time between the transmitted pulse and reflections from surface boundaries or internal defects reveals the quality of the material or the location of the defects. This time may be measured using a counter timer or viewing the output of the transducer on an oscilloscope display. In the past, the counter/timer was preferred because of the higher accuracy and resolution when compared with the oscilloscope. However, with the onset of transient digitizers and digital oscilloscopes, a highly accurate alternative is now available.

The 7D20 can capture a reflected signal and allow you to measure time referenced to the trigger and also to analyze the signal amplitude using a single cursor ( $\Delta$ OFF). Because the time base of the 7D20 is derived from a crystal clock, the horizontal timing accuracy is much better than that of conventional, ramp driven time bases. The improvement is most significant when delayed sweep is compared with the 7D20's delayed trigger position. For example, to have a variable delay up to 10 ms, a dual time base with delay must have the main sweep at 1 ms/div. If the trigger jitter is 0.05% of the Time/div, this results in 500 ns of jitter. If the delayed sweep is set at 1  $\mu$ s/div, the accuracy of the display is about 3%. At this Time/div, the jitter is nearly half of a division!

Whereas with the 7D20 at 10  $\mu$ s/div, 15 ms of delay is possible. The approximate timing accuracy for 10 ms delay is  $\pm 10 \mu$ s. The display jitter is one sample interval  $\pm 300$  ps. For this example it is approximately 100 ns. If less delay is required, then a faster sweep speed may be selected resulting in even better accuracy. The stated timing accuracy of the 7D20 is 0.1% of Full Scale  $\pm 300$  ps. Depending upon the TIME/DIV selected, measurement timing accuracies can approach that of a counter/timer.

## INSTRUMENTATION SET-UP #2



3857-51

To operate, the incident or outgoing transmitted pulse must be used to trigger the 7D20. You must select the delay time required to view the reflected signal. This is done by positioning the 7D20's trigger horizontally to read the desired number of divisions. The readout indicates a negative position when the trigger occurs beyond the left hand side of the screen. Only one cursor should be selected ( $\Delta$ OFF). This will allow you to make accurate time measurements referenced to the trigger (time=0).

The Pulse/Receiver in the setup illustration is a commercially available unit. It supplies the stimulus to the transmitting transducer and preconditions the received signal supplying sufficient output so it can be viewed on the oscilloscope display. The Sync Out is coincident with the transmitted pulse and therefore serves as the trigger for the 7D20.

As shown in the simulated signal in Figure 5-1, the cursor's horizontal coordinate value indicates the total time from the Sync Out, used as the trigger, to the viewed position on the crt screen. Note that the resolution is on the order of one part in  $10^5$  when compared to the total delay time available. The accuracy of the measurement is 0.1% of this reading.

Other 7D20 features such as multi-trace display and dual channel acquisition may be used to compare results from previous tests or to simultaneously capture two reflected signals. These may be acquired as fast as  $2 \mu\text{s}/\text{div}$  for a single event and as fast as  $50 \text{ ns}/\text{div}$  for repetitive events. Also through the IEEE-488 interface, further manipulation of the data by a controller or calculator may provide spectral information using Fourier analysis (FFT). For most ultrasonic, non-destructive testing, the 7D20 is much more flexible than a counter/timer.

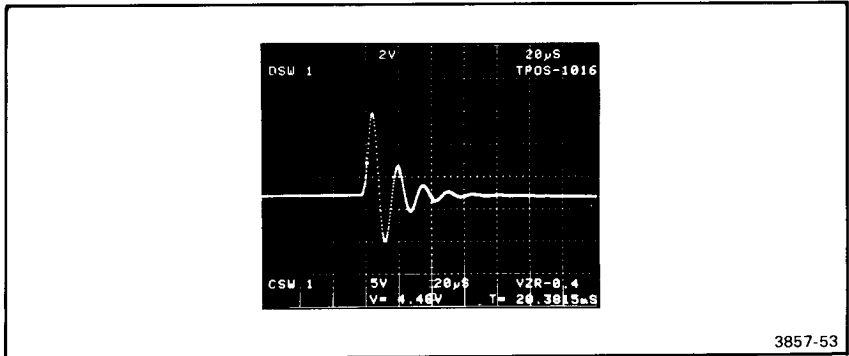


Figure 5-1. Viewing a reflected pulse signal showing the cursor position relative to the trigger pulse.

### APPLICATION 3 MONITORING INTERCELLULAR NEURONAL DISCHARGE

The measuring of nerve activity in stimulus-response experiments requires viewing long windows of time relative to the pulse width of the neuronal discharge. For a given record of time, such activity appears as a shower or burst of spikes resembling a partly deteriorated picket-fence. For example, the pulse width of a unit discharge may be 700  $\mu$ sec and viewed on an oscilloscope screen representing a total window of 1 to 5 seconds. On occasion, longer periods of time are used. Traditionally, bistable storage (DVST) oscilloscopes have been the preferred device for viewing such events. Seldom have digital storage techniques been used because of the need for extremely dense information.

The difficulty lies with the inability of digitizers to properly represent displacements which appear on conventional oscilloscopes as a vertical line without any width. In order to digitally capture any information, it must be sampled often enough to represent salient characteristics. For a simple rectangular pulse, only one sample on the top of the pulse is required to accurately represent its amplitude. One sample will be guaranteed as long as the digitizer's sample interval is shorter than the width of the pulse. If such a pulse width has 520  $\mu$ s sampled at 500  $\mu$ s per point to build a 5 second long record, 10,000 horizontal points would be required. Whereas, a bistable storage scope has horizontal locations which are only one molecule apart, so a very narrow pulse appears as a vertical line. This is like having billions of "dots". Although, the total number of perceived vertical lines is only a few hundred side-by-side.

In order to optimize the viewing of medium and slow speed events, the 7D20's ROLLing display presents a comprehensive, uninterrupted picture of events viewed at 0.1 s/div and slower. In the roll digitizing mode, it is possible to still detect the presence of events which are 500  $\mu$ s or greater in width by ENVelooping continuously. Based on a 500  $\mu$ s sample interval the equivalent number of horizontal points required without using ENV are listed in Table 5-1 below. Keep in mind that this is only true under certain special conditions.

**TABLE 5-1**  
**Effective Number of Points When Using ENV While ROLLing**

No. of Points Required	Total Displayed Time in Seconds (10*Time/div)
2,000	1
4,000	2
10,000	5
20,000	10
40,000	20
100,000	50
200,000	100
400,000	200

For the application of neuronal activity, intracellular responses are between  $500 \mu\text{s}$  and  $1 \text{ ms}$  in width which makes this approach valid. Caution must be taken to **never attach the 7D20 directly to human subjects**. Intracellular electrical potentials are on the order of  $50 \text{ mV}$  to  $100 \text{ mV}$  and may be applied to the 7D20 without preamplification.

#### APPLICATION 4 MEASURING PULSE JITTER, FREQUENCY SHIFT, AND AMPLITUDE VARIATIONS USING THE 7D20

Detecting subtle variations of amplitude and time in a signal is extremely difficult to do with a non-storage oscilloscope. Variable persistence scopes provide a means for recording such signal changes, which allow you to easily view and inspect signal characteristics. However, when the need arises to compare one record with a previous one, a certain amount of inconvenience is encountered. One difficulty is that although multiple sweeps are recorded, display modification is not possible. Comparisons are typically made between photos or photos and the crt display. If long term trends need to be analyzed and cataloged or a great number of devices need to be tested, photos become an unacceptable burden.

The 7D20 solves these problems. The ENveloping feature accumulates maximum and minimum values of successive sweeps and effectively simulates a variable persistence display. However, the 7D20 also provides all of the advantages of digital storage; bright and clear displays, indefinite storage and view time, display modification (position up and down, etc.), readout of waveform coordinate values, multiple waveform storage, and computer interface (IEEE-488).

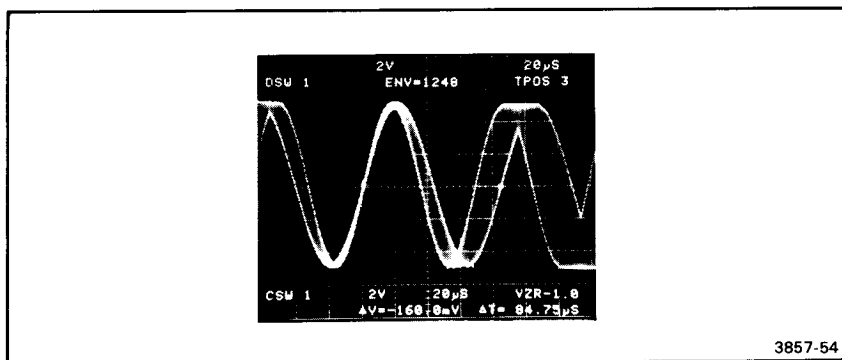


Figure 5-2. Using the ENveloping mode to show frequency shift.

## Applications—7D20

The bright and clear displays of up to six waveforms simultaneously, improves the visual comparison of acquired signals. The cursors allow for quantitative measures of signal variations and the IEEE-488 interface permits long term trend analysis and recording using a calculator or computer.

Of course, the key to this measurement is the ENVELOping. There are generally two forms of this, continuous or fixed. Fixed means a predetermined number of waveform acquisitions will be used to generate the envelope. You select this number via the SET N function ( $N = 2^n$ ,  $n = 3, 4, 5, 6, 7, 8$ ) of the 7D20. When N number of waveforms have been processed, the 7D20 terminates the ENV N mode and enters the HOLD state. When using the continuous or infinite ENVELOping, the 7D20 will process waveforms indefinitely until the HOLD button is pressed. Both ENV and ENV N also cancel by an alternate push of that button or by selecting AVE, AVE N, or HOLD NEXT.

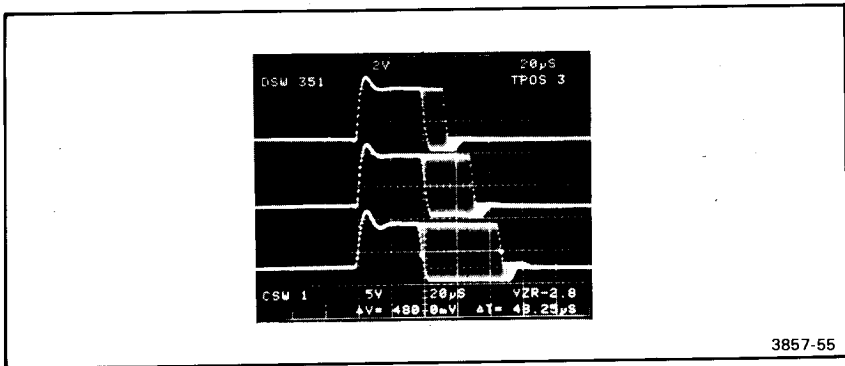


Figure 5-3. Using signal ENVELOping and multi-display feature to view pulse jitter.

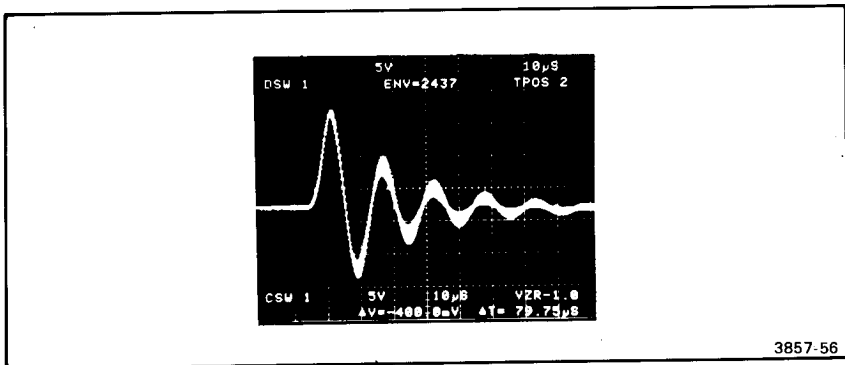


Figure 5-4. Using the 7D20 ENVELOping mode to show waveform variations.

Once a satisfactory envelope is created and held in memory, it may be analyzed using the cursors or by overlaying it with a previously generated envelope. To do this, simply use the CURSOR WAVEFORM modifiers to position the record up or down.

Now, even slight and subtle waveform variation may be easily revealed.

## APPLICATION 5 SOA ANALYSIS USING THE 7D20 AND 7A13

When testing any transistor switch where the collector current and collector-emitter voltage are out of phase, determining the energy dissipated by the device is essential. Since this energy may destroy the switching device, its Safe Operating Area (SOA) must be determined. This is a common concern when reactive loads must be driven such as inductive loads in power supplies and motors, or capacitive loads in liquid crystal displays.

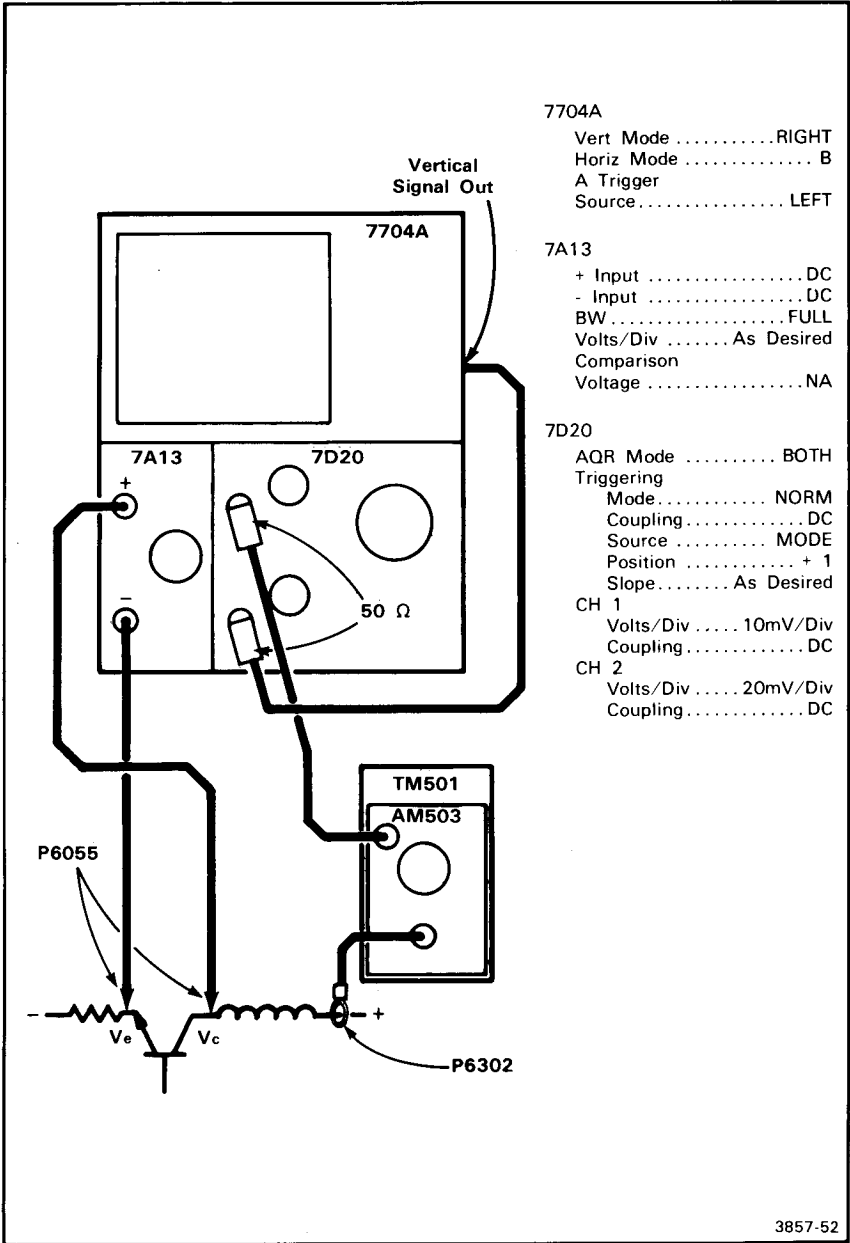
Some simple visual techniques can save a lot of time in finding the SOA without resorting to exacting calculations. Certainly, absolute accuracy is desired as provided by signal processing, but visual inspections reveal qualitative data in a very short time.

The simplest approach is to use an oscilloscope to display the collector current ( $i_c$ ) versus the collector-emitter voltage ( $V_{ce}$ ). This allows you to view the actual switching characteristics while the device is operating. It is also valuable to be able to see how both signals relate to time. In order to probe the circuit under test, a differential amplifier is needed to acquire a true  $V_{ce}$ . A current probe which can respond to dc as well as ac is recommended. However, if a "current sense resistor" is available, this is the most economical choice for representing  $i_c$ .

Another point to consider is destructive testing of the switching device. The display must be capable of capturing the signal when the failure occurs.

By using the 7D20 for acquisition and display, the TEKTRONIX 7A13 Differential Amplifier for preconditioning, and the AM 503/P6302 current probe and amplifier, SOA information can be easily viewed. The current probe accurately measures from dc to 50 MHz and provides a single-ended output to the 7D20. The 7A13 is a DC to 100 MHz differential amplifier. When installed in a 4 compartment 7000 Series mainframe with the 7D20, the signal conditioned by the 7A13 may be routed externally to the 7D20. This is provided via the mainframe's trigger signal output located at the rear of the mainframe. This signal is simply cabled and terminated at the 7D20's vertical input. The bandwidth of this patching is about 50 MHz when using a 50 ohm termination at the 7D20. If a 4 compartment mainframe is not available, the P6046 differential probe and amplifier may be used. This provides 100 MHz bandwidth but lacks the flexibility of the 7A13.

## INSTRUMENTATION SET-UP #5





## CALIBRATION

Since the signals are being preconditioned, the 7D20 cannot display the true sensitivity of the current probe's amps/div or the 7A13's volts/div. However, the AM 503 front panel can be read to determine its amp/div. This will be accurate provided its output is terminated into 50 ohms and displayed at 10 mV/div on the 7D20. A similar situation is true for the 7A13 but it will be necessary to adjust the gain at the 7A13. This is a simple front panel adjustment. To calibrate the 7A13/7D20 deflection and sensitivity, do the following:

1. Set the 7D20 CH 2 for 20 mV/div and for 500  $\mu$ s/div.
2. Set the 7A13 to 1 V/div.
3. Apply the 4 volt calibrator signal from the host mainframe to the + input of the 7A13.
4. Connect the instruments as shown in the setup illustration.
5. Adjust the "GAIN" adjustment on the 7A13's front panel to obtain 4 divisions of deflection displayed by the 7D20. If more adjustment range is required, use the Variable Volts/Div on the 7A13.

At this point, the actual volts/div may be read directly from the 7A13. Remember when selecting different sensitivities, DO NOT change the volts/div settings on the 7D20. Make all Amps/div and Volts/div selection at the AM 503 and 7A13.

## OPERATION

Connect the two P6055 probes to the circuit under test. If acquiring  $V_{ce}$  is desired, connect the + input to the collector and the - input to the emitter. The 7A13 will algebraically subtract the emitter voltage from the collector voltage to yield  $V_{ce}$ . Connect the P6302 around a convenient loop or element in series with the collector. Adjust the AM 503's and 7A13's deflection factors to produce signals of about three divisions each in amplitude. Set up the 7D20's display as follows:

1. Press "VS" followed by a "2" to create an X-Y display of  $i_c$  vs  $V_{ce}$ .
2. Press "REF" to view the "CSW" (cursor waveform),  $i_c$  vs time to serve as a time reference, and
3. Press "2" to display  $V_{ce}$  (Channel 2) versus time if desired.

Notice that cursors appear on the X-Y display as well as on the REFERENCE waveform. This is especially valuable to determine the time which a particular power level was sustained. Ultimately, this information can be used to determine the energy dissipated by the device.

Figure 5-5 shows an example of the information provided by the 7D20. If the REFERENCE waveform is not displayed, the cursors readout the X and Y coordinates

of the versus display. If the REF is displayed then the horizontal cursor coordinate reads time. So, by selecting the REF on and off, and positioning the cursors, data points may be extracted for computation. This will permit comparing measured information with the device manufacturers specifications. Figure 5-6 is an example of such specifications.

For repetitive signals the 7D20 may be used at any time/div setting. Because of the destructive nature of secondary breakdown, the  $i_c$  and  $V_{ce}$  information must be acquired in a single sweep in the event of a device failure. In this case, the 7D20 will acquire both channels in a single sweep as fast as  $2 \mu\text{s}/\text{div}$ . Since NORM

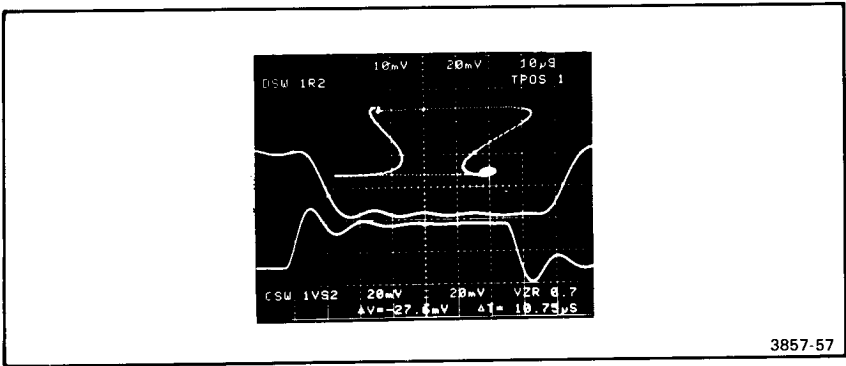


Figure 5-5. 7D20 display showing safe operating areas and other information.

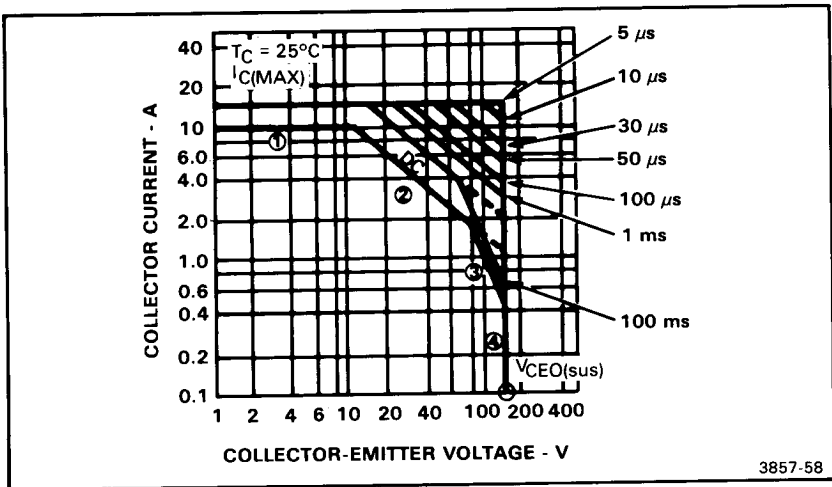


Figure 5-6. Example of manufacturer's spec's.

triggering is used, the 7D20 will simply retain the last triggered information. At that point, HOLD should be pressed to eliminate the chance of a stray signal or noise from destroying the data. Alternately, HOLD NEXT may also be used for a triggered hold.

This application is intended to aid in the SOA process and by no means represents a complete or rigorous method for making such measurements. A more complete discussion of this subject is available from Tektronix, Instrument Division, Lab Scopes Marketing.

## APPLICATION 6 USING THE 7D20 WITH THE 7854 FOR WAVEFORM PROCESSING

The 7854 is a waveform processing oscilloscope which allows you to manipulate waveforms similar to using a scientific programmable calculator. The 7854 digitizer has a single event capture speed slightly above the audio spectrum. The 7D20 extends this ability to capture single events into the video spectrum.

Also, the settings of the 7D20 may be remotely programmed for use in an automated measurement environment. In this application, it is important to point out that this combination is not the most efficient system component when speed is considered. However, it may be desirable since it greatly reduces the software impact on the system engineer. The 7854's waveform functions eliminate the need to create and verify many commonly used algorithms. In this sense, the 7854 and 7D20 provide the shortest path to designing a waveform measurement system.

Before proceeding, a few technical points must be considered to explain the use of these two instruments.

The 7D20 is a waveform digitizer. Its measurement accuracy is derived from its waveform memory. The 7D20 produces a refreshed display (when installed in any 7000 Series mainframe) which is in no way time related to the events which it captures. Also, the height, width, and position of the display can be adjusted from the front panel which can seriously affect measurements made with the graticule. However, the contents in memory are not disturbed and the cursor readout will remain accurate because it is independent of the display adjustments.

The 7854 acquires waveform information from the plug-ins based on the height, width, and position of the display. Accuracy of the 7854 is limited primarily by the plug-ins installed in it. Hence, the display output of the 7D20 does not present the most accurate representation of its memory contents. Errors are also compounded when the 7854 digitizes the 7D20's output. Quantization, noise, and linearity all contribute to this error.

There is a simple solution to this dilemma, the IEEE-488 interface. The only logical way to accurately represent a 7D20 waveform in the 7854 is to transfer the actual digital data.

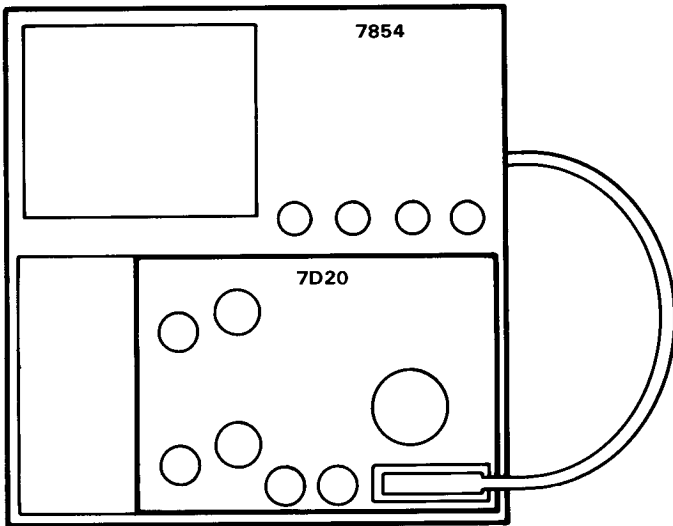
## INSTRUMENTATION SET-UP #6

7854

IEEE-488 (REAR PANEL SELECTS)  
MODE ..... LISTEN ONLY (CODE 11)  
STATE ..... ON LINE  
DISPLAY MODE ..... STORED

7D20

IEEE-488 (MENU SELECTED)  
MODE ..... TALK ONLY



3857-59

## MANUAL TRANSFER

When using the 7854 and the 7D20 without an IEEE-488 controller, provisions have been made on both instruments to allow you to transfer data manually.

Initially, set up the 7854 for a SCOPE display, install the 7D20 as shown in the set-up illustration, and acquire a waveform into the 7D20. If you have more than one waveform in the 7D20's memory, you can transfer them one at a time to the 7854. Use the CSW key to designate the waveform to be transferred. Press the 7D20's ID key to set the transfer mode to TALK ONLY and select the UTILITIES menu. Select 1024 points per waveform on the 7854.

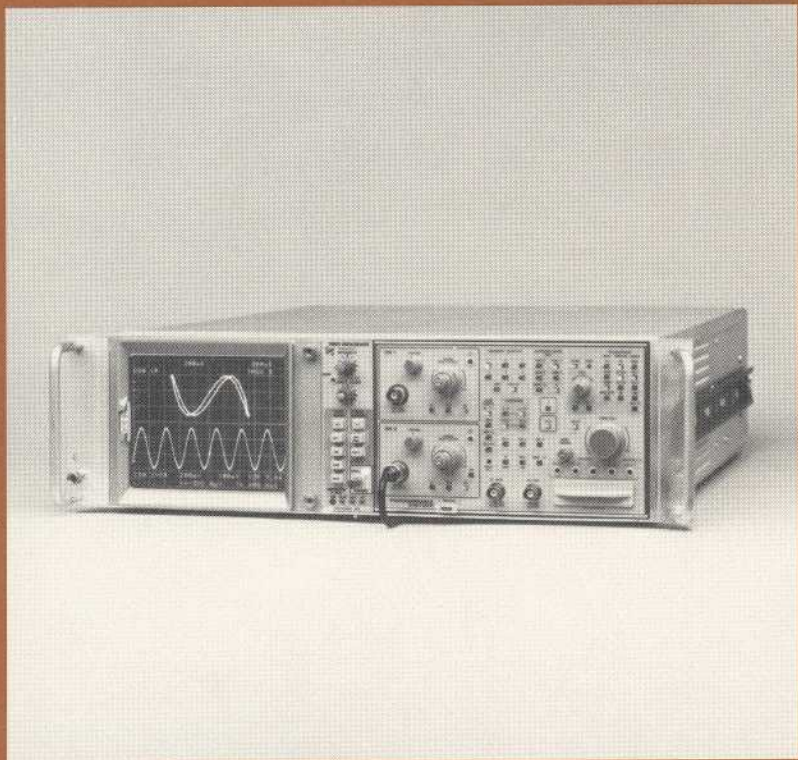
To initiate the transfer, press READX on the 7854 and select SEND CSW ASCII from the UTILITIES MENU of the 7D20 by pressing key number 1. The resulting display on the 7854 will have a horizontal scale factor which is not an integer value; however, the timing information is correct. The difference in scale factors occur because the 7D20 is calibrated for 100 points/div, where the 7854 is 102.4 points/div (when  $P/W=1024$ ). If desired, an integer scale factor can be produced by a 1.024 HXP command on the 7854. If the 7D20 acquired its waveform in the Extended Real-time range (200  $\mu\text{s}/\text{div}$  to 2  $\mu\text{s}/\text{div}$ ), 820 point waveforms will result. If an 820 point waveform is sent to the 7854, all of the points will transfer but the value of points 821 through 1024 on the 7854 will be filled with zeros and the 7854 will issue a warning.

This waveform may be expanded to fit the 820 original points into an interpolated 1024 points. On the 7854, turn off the cursors; key in 0, WFM, 1.28 and press HXP.

## COMPUTER CONTROLLED TRANSFER

Data transfer from the 7D20 to the 7854 may be conducted using an IEEE-488 controller. In this environment, both the 7854 and the 7D20 should be set up for two way communication, TALK/LISTEN. From the controller, the 7854 should be instructed to listen and initiate a READX, and the 7D20 instructed to talk and send the desired waveform. By a command from the IEEE-488 interface, the 7D20 can interpolate the data before it is sent. When used, the 7D20 sends 1024 points to the 7854.

This technique relies on the controller to coordinate the transfer but not actually handle the waveform data.



Your 7D20 Programmable Digitizer may include one or more options. This section includes a brief description of each option. For further information about available options, refer to the Tektronix Products Catalog or contact a Tektronix Field Office.

# SECTION 6

---

# INSTRUMENT OPTIONS

---

No options were available for the 7D20 at the time of this printing.

Information about any future options can be found in the Change Information section at the back of this manual.



# INDEX

- A** ACCESSORIES:  
    Optional: 1-12  
    Standard: 1-12  
APPLICATIONS: 5-1  
AQR:  
    Explanation of: (*see* CONTROLS AND CONNECTORS)  
    Use of: (*see* EXERCISES)  
AVE:  
    Explanation of: (*see* CONTROLS AND CONNECTORS)  
    Use of: (*see* EXERCISES)
- B** BLOCK DIAGRAM:  
    Description of: 3-1  
    Illustration for: 3-2, 3-3
- C** COLOR, IN THIS MANUAL: 1-1  
COMMANDS: (*see* GPIB)  
CONTROLS AND CONNECTORS: (*also see* EXERCISES for usage)  
    Description and Illustration of: 2-2  
    Operation of:  
        AC, DC, GND: 2-93,  
        ADD: 2-23, 2-95  
        AQR GAIN: 2-94  
        AQR MODE: 2-95  
            ADD  $\phi$ 1: 2-95  
            BOTH: 2-23, 2-95  
            CH1  $\phi$ 1: 2-23, 2-95  
            CH2  $\phi$ 2: 2-23, 2-96  
        AVE: 2-68, 2-116  
        AVE N: 2-70, 2-118  
        CURSORS: 2-32, 2-112  
            ALIGN: 2-36, 2-115  
             $\Delta$ OFF: 2-35, 2-115  
             $\Delta$ ON: 2-34, 2-115  
            INDEP: 2-36, 2-115  
             $\phi$ 1 $\phi$ : 2-32, 2-33, 2-114  
             $\phi$ 2 $\phi$ : 2-35, 2-115  
        CURSOR WFM: 2-42, 2-108  
        ALL: 2-56, 2-110  
        HMAG: 2-49, 2-110

## C

- VCMP  $\frac{\square}{\square}$ : 2-46, 2-109
  - VPDN  $\frac{\square}{\square}$ : 2-45, 2-108
  - VPUP  $\frac{\square}{\square}$ : 2-43, 2-108
  - VS: 2-63, 2-110
  - VXPD  $\frac{\square}{\square}$ : 2-46, 2-109
  - ENV: 2-69, 2-118
  - ENV N: 2-71, 2-120
  - EXT CLOCK: 2-79, 2-122
  - EXT TRIG: 2-123
  - DISPLAY CAL: 2-10, 2-84, 2-90
  - $f$ : 2-52, 2-96
  - GPIB Connector: 2-123
  - HOLD: 2-31, 2-96
  - HORIZ POSITION: 2-105
  - ID (ADDR): 2-121
  - Input Connectors:
    - Signal Connection: 2-92
  - INV (CH2): 2-94
  - MEMORY DISPLAY: 2-24, 2-105
    - COPY: 2-25, 2-106
    - CSW: 2-28, 2-107
    - REF: 2-52, 2-107
  - POSITION: 2-93
  - REMOTE ONLY: 2-121
  - RQS: 2-121
    - #: 2-121
  - SET N: 2-67, 2-70, 2-115
  - MENU TEST: 2-83, 2-120
  - TIME/DIV: 2-72, 2-96
  - TRIGGERING:
    - COUPLING: 2-100
    - LEVEL: 2-104
    - MODE: 2-101
    - +SLOPE: 2-103
    - SOURCE: 2-99
    - TRIG'D: 2-104
    - $\frac{\square}{\square}$ TRIG POS  $\frac{\square}{\square}$ : 2-74, 2-104
  - VECTOR: 2-105
  - VOLTS/DIV (VARIABLE): 2-21, 2-92
- COPY:  
Function of: 2-106
- CURSORS:  
Explanation of: (see CONTROLS AND CONNECTORS)  
Use of: (see EXERCISES)

## D

- DIGITIZING MODES: (see MODES)
- DISPLAY:
  - Alphanumeric Readout Fields: 2-91
  - Calibration, description of: 2-90 (see also EXERCISES)
  - Illustration of: 2-6, 2-7

- E** ENV:  
Explanation of: (*see* CONTROLS AND CONNECTORS)  
Use of: (*see* EXERCISES)
- EXERCISES, GET-ACQUAINTED:  
Details of:  
AQR Mode: 2-22  
Average and Envelope Functions and SET N: 2-67  
Changing the Designated Cursor Waveform: 2-28  
Cursors: 2-32  
Cursor Waveform:  
Display Modifier Keys: 2-42  
Horizontal Magnification: 2-49  
HMAG and the REFERENCE Waveform: 2-52  
Keys Active for Waveforms 1 or 2: 2-44  
Vertical Expansion and Compression: 2-46  
Vertical Reposition: 2-43  
Digitizing Mode and the TIME/DIV Control: 2-72  
Equivalent Time Mode: 2-73  
Real-Time and Extended Real Time: 2-72  
Roll Mode and EXTERNAL CLOCK: 2-72  
External Clock: 2-79  
HMAG ALL: 2-56  
Hold Next Key: 2-77  
Holding a Waveform in Memory: 2-31  
Input Coupling: 2-19  
Initialize, Routine: 2-14  
REFERENCE with VS: 2-65  
Trigger Position:  
How to Set: 2-74  
Vectors and Dots: 2-66  
Vertical Controls: 2-17  
Vertical Zero Reference: 2-19  
VOLTS/DIV Control: 2-21  
Variable: 2-21  
VS Function: 2-63  
Waveform Memory: 2-24  
Display Calibration:  
Procedure: 2-11  
Preliminary Set-up: 2-9
- EXT CLOCK:  
Explanation of: (*see* CONTROLS AND CONNECTORS)  
Use of: (*see* EXERCISES)

- F** FRONT-PANEL CONTROLS: (*see* CONTROLS AND CONNECTORS)  
FUNCTION KEYS:  
Description of: 2-8, 2-96

**G**

## GPIB:

- Chart, ASCII & IEEE 488 Code: 4-18
- Command List (Set): 4-38
  - Acquisition Group: 4-56
  - Calibration Group: 4-65
  - Channel 1 Group: 4-39
  - Channel 2 Group: 4-42
  - Cursor Group: 4-59
  - Cursor Waveform Group: 4-53
  - Device Trigger: 4-62
  - Display Group: 4-50
  - Initialization Group: 4-63
  - Programming Aids: 4-75
  - Readout/Text: 4-74
  - Selftest: 4-64
  - Service Request Group: 4-66
  - Stored Settings Group: 4-61
  - Time/Div Group: 4-48
  - Triggering Group: 4-45
  - Waveform Curve: 4-72
  - Waveform Preamble and Curve: 4-73
  - Waveform Preamble Group: 4-68
- Command Usage: 4-11
- Curve Commands: 4-23
- Curve Queries: 4-23
- Description of: 4-2
- Device Dependent Messages: 4-11
- Event Codes: 4-36
- Event Queries: 4-32
- Format Diagrams:
  - Complex Command: 4-11
  - NR1 Data: 4-13
  - NR2 Data: 4-14
  - NR3 Data: 4-15
  - Query: 4-15
  - Simple Command: 4-12
- Introduction to: 4-1
- Messages:
  - Device Dependent: 4-11
  - Interface: 4-9
- Memory Display Keypad Code, Table for: 4-7
- Mode, Terminator, and Address Selection: 4-4
- Operation, Remote and Local: 4-7
- Power Up: 4-3
- Sample Programs, List of: 4-77
  - 4041:
    - Application Programs:
      - Event Capture: 4-111
      - SRQ Handler: 4-108
      - Store and Recall Front-Panel Settings: 4-109

**G**

## Operating Programs:

- ASCII Waveform Transfer to Controller: 4-97
- Binary Numeric Array Waveform Transfer to Controller: 4-99
- Binary Numeric Array Waveform Transfer to 7D20: 4-100
- Input 7D20 Front-Panel Settings from String Array: 4-101
- Output 7D20 Front-Panel Settings to String Array: 4-102
- Setting the 7D20 to Local: 4-104
- Setting the 7D20 to Remote: 4-103
- Text Generation: Controller to 7D20: 4-107
- Text Transfer: Tape to 7D20: 4-106
- Text Transfer: 7D20 to Tape: 4-105

## 4050-Series:

## Application Programs:

- Event Capture: 4-89
- SRQ Decoding Routine: 4-93
- Store 7D20 Settings on Magtape: 4-91

## Operating Programs: 4-78

- Implement and Store 7D20 Settings: 4-86
- Print Poll Statement: 4-88
- Query Functions: 4-87
- Query Routine: 4-85
- Text Generation: 4050-Series to 7D20: 4-78
- Transfer Waveform Data to Controller—ASCII Format: 4-81, 4-83
- Transfer Waveform Data to Controller—Binary Format: 4-82
- Transfer Waveform Variables and Arrays to 7D20: 4-84

## Service Requests:

- Masks: 4-31
- Status Bytes, table of: 4-26
- Use of: 4-25

## Status Indicators: 4-7

- Waveform Data Commands: 4-24
- Waveform Preamble Commands: 4-20
- Waveform Preamble Queries: 4-21

**H**

## HMAG:

- Explanation of: 2-110 (*see also* EXERCISES)
- Use of: (*see* EXERCISES)

## HOLD:

- Explanation of: (*see* CONTROLS AND CONNECTORS)
- Use of: (*see* EXERCISES)

**I**

## ID:

- Version Identification: 2-121 (*see also* GPIB: Mode)

INITIALIZE: (*see* EXERCISES)

- Front Panel: (*see* MENU FUNCTIONS)

INSTALLATION: (*see* INSTRUMENT)

**I** INSTRUMENT:

- Application for: (see APPLICATIONS)
- Description of: 1-1
- Display: (see DISPLAY)
- Drawing of: 1-13
- Installation of: 1-3
- Options: (see OPTIONS)
- Power Up: 2-81

**K** KEYS: (see FUNCTION KEYS)

**M** MEMORY:  
Display: (see CONTROLS AND CONNECTORS)  
Waveform: (see EXERCISES) (see also WAVEFORM)

MENU FUNCTIONS: 2-83

MASTER MENU: 2-83

- 1, # STORE PANEL #: 2-83
- 2, # RECALL #: 2-84
- 3 DISPLAY CAL PATTERN: 2-84
- 4 UTILITIES: 2-84

SELFTEST:

- Details of: 2-86

TEST MENU:

- Details of: 2-86

UTILITIES Menu: 2-85

- 1 SEND CSW ASCII: 2-85
- 2 SEND CSW BINARY: 2-85
- 3 READOUT ON/OFF: 2-85
- 4 EXT CLOCK POLARITY: 2-86
- 5 INIT FRONT PANEL: 2-86
- 6 MASTER MENU: 2-86

MESSAGES:

Error:

- General: 2-8

MODES:

Digitizing:

- Explanation of: 3-4
- Table of: 2-97

Illustrations of:

- Equivalent Time Digitizing: 3-8
- Extended Real-Time Digitizing: 3-8
- Real-Time Digitizing: 3-7
- Roll: 3-7

**O** OPTIONAL ACCESSORIES: (see ACCESSORIES)  
OPTIONS: 6-1

- P** PACKAGING, for Shipment: 1-4  
POWER UP: (*see* INSTRUMENT)  
PROGRAMS: (*see* GPIB)  
PROMPTS AND WARNINGS: 2-88
- R** RACKMOUNT MAINFRAMES: 1-4  
READOUT DISPLAY: 2-91  
REAL-TIME SIGNAL ACQUISITION CAPABILITIES: 3-9
- S** SAFETY SUMMARY: (*see* FRONTMATTER)  
SHIPPING: (*see* PACKAGING)  
SPECIFICATIONS, Tables for:  
    Electrical:  
        Digitizer: 1-9  
        Trigger: 1-7  
        Vertical: 1-6  
    Environmental: 1-11  
    Physical: 1-11  
    Illustration: 1-13  
STANDARD ACCESSORIES: (*see* ACCESSORIES)
- T** TRIGGERING:  
    Explanation of: 2-99 (*see also* CONTROLS AND CONNECTORS)
- V** VECTOR:  
    Explanation of: (*see* CONTROLS AND CONNECTORS)  
    Use of: (*see* EXERCISES)  
VOLTS/DIV:  
    Explanation of: (*see* CONTROLS AND CONNECTORS)  
    Use of: (*see* EXERCISES)
- W** WAVEFORM:  
    Cursor: 2-42, 2-108, 2-113 (*see also* EXERCISES)  
    Display, Operational Theory: 3-11  
    Obtaining a: 2-16  
    Memory: 2-24, 2-105
- Z** ZERO REFERENCE:  
    Setting (*see* EXERCISES)