TEK SPS BASIC
V02/V02XM
DPO Driver Package

CP57004/CP57504

Tektronix®

COMMITTED TO EXCELLENCE

BEFORE READING

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

Tektronix®
COMMITTED TO EXCELLENCE

TEK SPS BASIC
V02/V02XM
DPO Driver Package
CP57004/CP57504

INSTRUCTION MANUAL

Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077

Serial Number _____

First Printing JUN 1979

PRODUCED BY SPS DOCUMENTATION GROUP

070-2739-01

SOFTWARE SUPPORT POLICY

Unless otherwise provided, Tektronix, Inc., agrees that during the one (1) year period following installation, if the customer encounters a problem with this software which the customer's diagnosis indicates is caused by a software defect, the customer may submit a Software Performance Report to Tektronix, Inc. For problems occurring in current, unaltered releases of software, Tektronix, Inc., will respond to Software Performance Reports via a software maintenance periodical. The software maintenance periodical will be provided at no cost to the customer for one year following installation and will contain information for correcting or bypassing verified problems where possible, and will give notice of availability of corrected software.

Any software updates released by Tektronix, Inc., to correct problems during the one (1) year period will be provided to the customer at no charge on the standard distribution media specified in the software documentation. If media other than the standard distribution media is requested, the customer will only be charged for the current cost of the optional media.

SOFTWARE LICENSE

This software product, including subsequent improvements or updates, is furnished under a license for use on a single controller. It may only be copied, in whole or in part (with the proper inclusion of the Tektronix, Inc., copyright notice on the software), for use on that specific controller.

Specification and price change privileges are reserved.

Although the material in this manual has been thoroughly edited and checked for accuracy, Tektronix, Inc., makes no guarantees against typographical or human errors. Also, Tektronix, Inc., assumes no responsibility or liability, consequential or otherwise, of any kind arising from misinterpretation or misuse of the material in this manual. The contents of this manual are subject to change without notice.

Copyright © 1979, 1980 by Tektronix, Inc., Beaverton, Oregon. Printed in the United States of America. All rights reserved.

U.S.A. and foreign TEKTRONIX products covered by U.S. and foreign patents and/or patents pending.

TEKTRONIX and TEK are registered trademarks of Tektronix, Inc.

DEC, LSI-11, PDP, RT-11, and UNIBUS are registered trademarks of Digital Equipment Corporation.

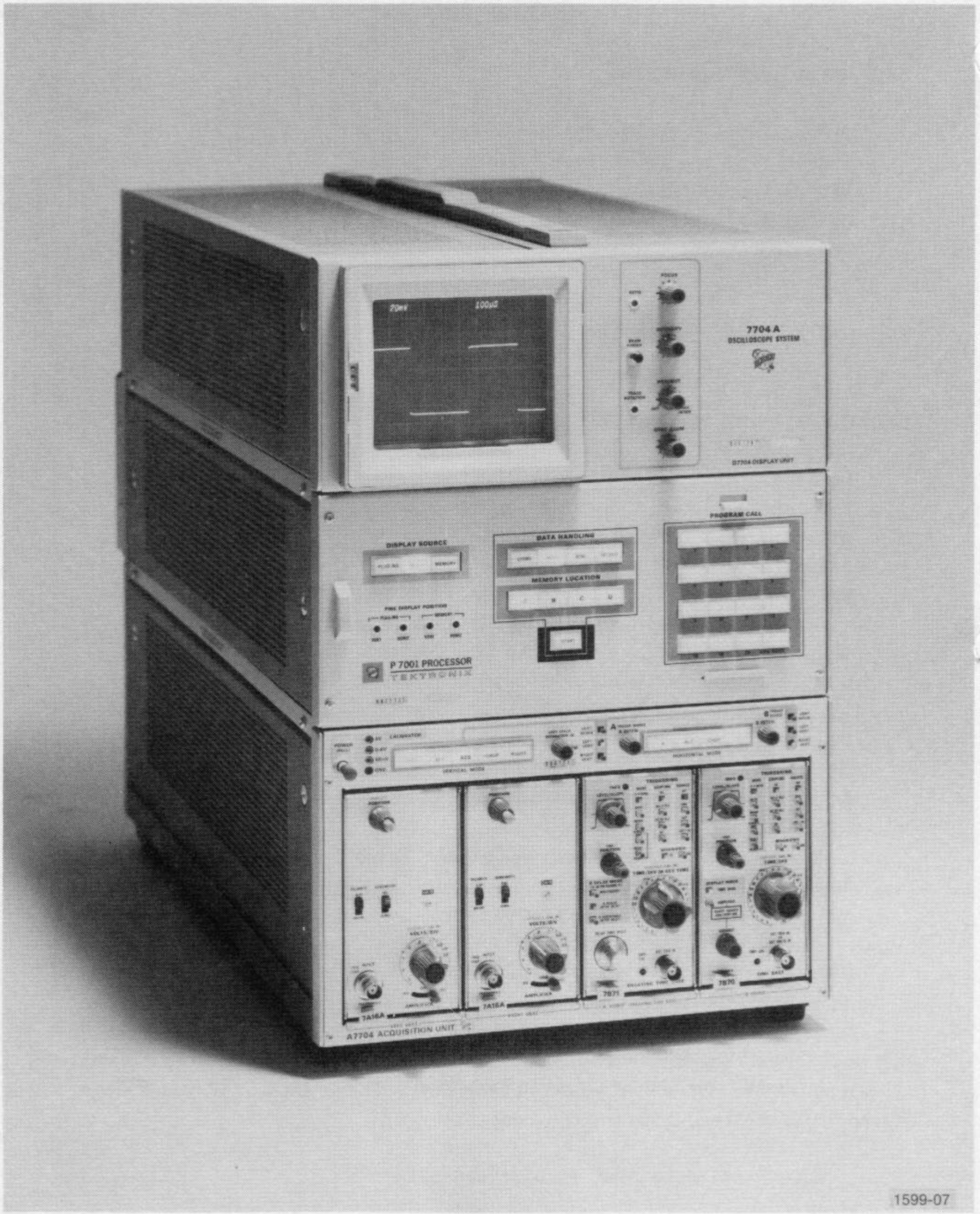
PREFACE

This manual describes the DPO Driver Package for TEK SPS BASIC VØ2 and VØ2XM software. Any exception to an option or a capability of this driver being supported by a specific release of the software is noted where appropriate.

The prerequisite software for executing the driver in this package is the corresponding version of the TEK SPS BASIC System Software. The VØ2 package (CP57004) requires the VØ2 System Software (CP57000); the VØ2XM package (CP57504) requires the VØ2XM System Software (CP57500).

TABLE OF CONTENTS

@ FEB 1980



The TEKTRONIX Digitizing Oscilloscope.

SECTION 1

INTRODUCTION

Prerequisite Software and Hardware

This manual describes the instrument driver for the TEKTRONIX Digitizing Oscilloscope (DPO). This software module, called "DPO.SPS" is meant to be used with the TEK SPS BASIC VØ2 System Software in systems including up to four DPOs. In addition to one or more DPOs, your hardware configuration must include a controller with at least 28K words of memory, a terminal, and either a dual-drive mass-storage device or two single-drive mass-storage devices.

This manual assumes you are familiar with the prerequisite software and hardware. Please refer to the appropriate manuals when you have a specific software or hardware question. The software manuals are TEK SPS BASIC VØ2 System Software, Peripheral Drivers, and this manual. The hardware manuals needed will depend on the controller, terminal, and mass storage devices used, but should include the DPO Operators manual.

Integrating the DPO Driver Into Your System

If you purchased the DPO driver as a part of a new or updated system, skip this section; the driver has already been integrated. However, if you obtained the module separately, you need to integrate it into your software system. The method for transferring this module from the purchased medium to the medium of your system software is discussed in general here. For greater detail, refer to your Peripheral Drivers manual.

When the system device medium is the same as that of the purchased module, integration is simple. With the working copy of your software in drive Ø of your system device, put your copy of the DPO driver into drive 1. Now COPY the driver to your working copy. For example, if you have a floppy disk system, type in the command:

```
COPY DX1:"DPO.SPS" TO DXØ:"DPO.SPS"
```

When the system device medium is <u>not</u> the same as that of the purchased module, integration requires some extra steps. First, LOAD the

driver for the device used by the module. With the working copy of your software in drive Ø of the system device, place the module copy into drive Ø of the proper device for that medium. Now COPY the driver to your working copy. Finally, RELEASE the driver you LOADed.

For example, to integrate the DPO driver onto a hard disk working copy in drive Ø (DKØ) from a floppy disk in drive Ø (DXØ), use the following statements:

```
LOAD "DX.SPS"
COPY DXØ:"DPO.SPS" TO DKØ:"DPO.SPS"
RELEASE "DX.SPS"
```

DPO Driver Operations

The DPO driver can communicate with up to four DPOs cabled to a system. This communication may be summarized as:

1. Storing waveforms in DPO memory
2. Transferring waveforms in floating-point format
3. Sending waveforms to the DPO memory for display
4. Signal averaging
5. Reading the status of the single-sweep function
6. Setting the single-sweep function
7. Displaying ASCII text on the DPO CRT
8. Acquiring ASCII readout information
9. Setting up DPO interrupts

Syntax and Descriptive Forms of the Standard Instrument Commands

The DPO Driver Package does not include a set of special nonresident commands to accomplish these operations. Instead, the DPO Driver uses the six standard instrument commands that are a part of the System Software: ATTACH, DETACH, PUT, GET, WHEN, and IGNORE. Since each

instrument driver uses a subset of the legal forms of these commands, the syntax and descriptive forms allowed by the DPO driver are shown here. (For an explanation of the syntax form and descriptive form, including the notation used, see Section 4 of the TEK SPS BASIC VØ2 System Software manual.)

Forms of ATTACH Used by the DPO Driver

Syntax Form:

[line no.] **ATTACH #**expression **AS DPO**[constant]**:**

Descriptive Form:

[line no.] **ATTACH #**ilun **AS DPO**[hardware unit number]**:**

Forms of DETACH Used by the DPO Driver

Syntax Form:

[line no.] **DETACH** { **#**expression / **ALL** }

Descriptive Form:

[line no.] **DETACH** { **#**ilun / **ALL** iluns }

Forms of PUT Used by the DPO Driver

Syntax Form:

[line no.] **PUT** { array expression / waveform expression / string expression } [, { array expression / waveform expression / string expression }] ...

INTO #expression[,string expression]...

Descriptive Form:

[line no.] **PUT** {source array expression / source waveform expression / source string expression}

[,{source array expression / source waveform expression / source string expression}]...

INTO #ilun[,driver-dependent specification of data or status information to be sent to DPO]...

Forms of GET Used by the DPO Driver

Syntax Form:

[line no.] **GET** {variable / floating-point array / floating-point waveform / string variable}[,{variable / floating-point array / floating-point waveform / string variable}]...

FROM #expression,string expression[,string expression]...

Descriptive Form:

[line no.] **GET** {target variable / target array / target waveform / target string variable}[,{target variable / target array / target waveform / target string variable}]...

FROM #ilun, driver-dependent specification of data or status information to be obtained from DPO [,driver-dependent specification of data or status information to be obtained from DPO]...

Forms of WHEN Used by the DPO Driver

Syntax Form:

[line no.] **WHEN** **#**expression **HAS** string expression
[**AT** expression] [**AS TASK** expression]
GOSUB line number

Descriptive Form:

[line no.] **WHEN** **#**ilun **HAS** driver-dependent interrupt specification
[**AT** priority level] [**AS TASK** task number]
GOSUB line number

Forms of IGNORE Used by the DPO Driver

Syntax Form:

[line no.] **IGNORE** { **#**expression [, { **TASK** expression / string expression }] / **TASK** expression / **ALL** }

Descriptive Form:

[line no.] **IGNORE** { **#**ilun [, { **TASK** task number / driver-dependent interrupt specification }] / **TASK** task number / **ALL** interrupt conditions for all instruments }

How these commands are used to communicate with the DPO and what specific driver-dependent strings are recognized by the DPO driver are covered in Section 3 of this manual. However, before the details of each of these operations is discussed, how the DPO works is briefly explained in Section 2.

SECTION 2

THE DIGITIZING OSCILLOSCOPE

The TEKTRONIX Digitizing Oscilloscope (DPO) is a computer controllable oscilloscope. This controllability is provided by taking a standard TEKTRONIX 7704A Oscilloscope, separating the Acquisition and Display Units, and reassembling them with a P7001 Processor placed in between. Figure 2-1 illustrates this. Except for the P7001, the DPO is operated just like a conventional oscilloscope. Hence the operation of the 7704A oscilloscope mainframe and its plug-ins is not discussed here.

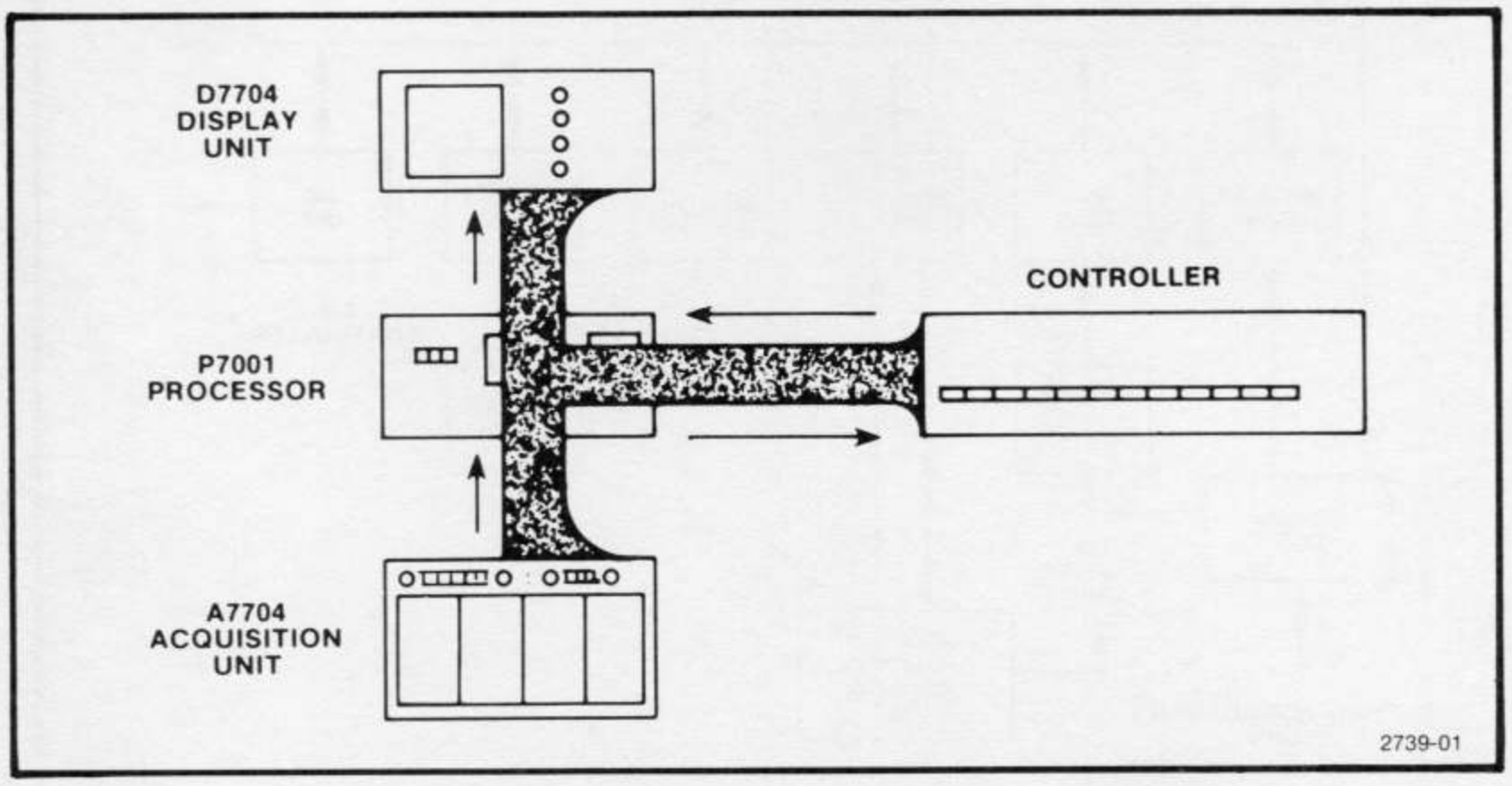


Fig. 2-1. The P7001 Processor makes the 7704A Oscilloscope computer controllable.

The P7001 Processor serves as a digitizer and three-way interface between the two portions of the 7704A Oscilloscope and the controller (minicomputer). On one arm, it accepts inputs from the acquisition portion of the oscilloscope so that waveforms can be digitized and stored in memory. On the other arm, it directs outputs to the display portion of the oscilloscope so that digitized and processed waveforms from the controller can be displayed on the CRT. The two arms of the P7001 join to provide a two-way communication link with the controller. The result is a computerized 175-megahertz oscilloscope.

With the aid of the block diagram in Fig. 2-2, let's take a closer look at the P7001 Processor to see how it stores, transfers, and displays data. First, the Signal Interface block acts as a traffic interchange for signal paths. This interface provides a path for analog waveforms from the A7704 Acquisition Unit to the Sample and Hold circuitry and the Analog-to-Digital Converter. It also provides a path for analog waveforms to the D7704 Display Unit, where they can be displayed in the normal oscilloscope format. Further, a path exists through the Signal Interface for digitized waveforms received from the P7001 Memory via the Display Generator.

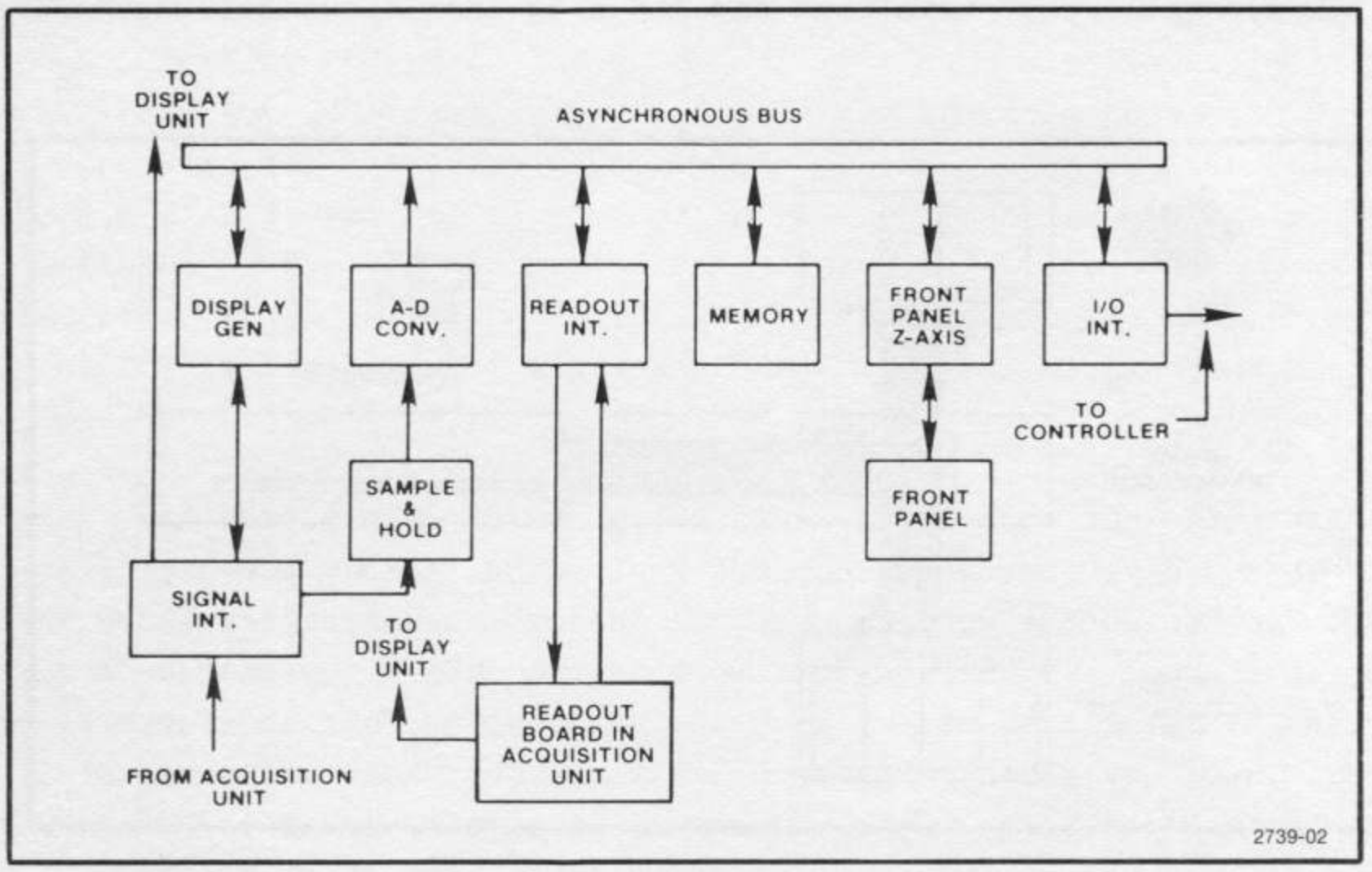


Fig. 2-2. P7001 Processor block diagram.

Waveforms from the plug-in units exist in the form of an amplified analog signal. The DPO converts this to a digitized waveform that can be processed by the controller. The first step in digitizing is accomplished by the Sample and Hold circuit.

The Sample and Hold circuit samples waveforms at 6.5 microsecond intervals and outputs a horizontal and vertical analog value for each sample. The vertical value is the instantaneous voltage appearing at the Y-axis input. The horizontal value is the instantaneous value of the ramp or X-axis voltage.

To fully represent one trace on the DPO CRT, it is necessary to sample 512 points. On sweep speeds slower than Ø.5 ms/div, all 512 samples can usually be taken during one sweep on the DPO, thereby providing real-time sampling. Further sweeps merely provide updated samples. However, on sweep speeds faster than Ø.5 ms/div, all 512 samples cannot be taken during one sweep. Thus, sampling of repetitive waveforms is done on an equivalent-time basis. In this case, sampling occurs at different points on the trace, and full sampling is built up over several sweeps.

As each sample is taken, the Analog-to-Digital Converter digitizes the sample. The vertical analog value is converted to a 1Ø-bit word in the range of Ø-1777 octal; this is equivalent to Ø-1Ø23 decimal. The point Ø corresponds to one division below the bottom CRT graticule line and point $1Ø23_{1Ø}$ corresponds to one division above the top graticule line. The horizontal analog value of the sample is converted to a 9-bit word in the range of Ø-777 octal; this is equivalent to Ø-511 decimal. In this case, point Ø corresponds to the left-hand graticule line and point 511_{10} corresponds to the right-hand graticule line. (See Fig. 3-1 in Section 3.)

After the horizontal and vertical values of each sample have been converted to digital words, they are sent to the Memory board of the P7001. The horizontal word becomes the address (location) at which the value is stored, and the vertical word is the data value stored in that address. Assuming a 4K memory is being used, up to four waveforms may be stored. These waveforms consist of 512 mantissa values and associated scale factors and units. Space in memory is also reserved for storing zero-reference values, zero-reference messages, and other messages generated by the controller.

Once the digitized data values have been stored in memory, they can be displayed on the CRT by means of the Display Generator. This is essentially a Digital-to-Analog Converter that converts the vertical and horizontal words back to an analog format suitable for display. The Display Generator also draws vectors between data points when an incomplete set of waveform samples is encountered.

Associated with the Display Generator is the Readout Interface. It processes analog scale-factor signals from the plug-ins into an ASCII format that can be stored in memory. Also, when a waveform is called up for display through the Display Generator, the Readout Interface converts the ASCII scale-factor values back to analog values for display on the CRT.

In most cases, it is necessary to send the digitized waveform values to the controller for processing. This transfer is accomplished by the Input/Output interface, more specifically known as the DPO/CP Bus Interface. It provides the control link between the P7001 Processor and the controller. Thus, values can be sent to the controller for processing, and the processed values can be received by the DPO for display.

The circuit board for the Front-Panel contains all of the pushbuttons on the P7001 front panel. It monitors which buttons are pressed, sets the DPO status accordingly, and lights the appropriate buttons to indicate the DPO status.

All of the blocks in the P7001, except the Signal Interface and the Sample and Hold circuit are connected directly to an Asynchronous Bus. This bus allows each block to work interactively with other blocks on the bus so that various P7001 operations can be performed.

The DPO front-panel buttons allow you to store a waveform automatically or to select the display source (either from the plug-ins or a particular memory location). With the use of a WHEN command, you can program certain buttons to send a waveform to the controller for processing or to receive a waveform for display on the DPO. The DPO Program Call buttons allow you to generate interrupts which can be used to schedule routines for execution in controller memory. Each of these operations will be discussed in the next section.

SECTION 3

THE DPO DRIVER

Loading and Releasing the DPO Driver

The first step in DPO communication is to load the DPO driver into controller memory from the peripheral device on which it is stored. Normally, the driver is stored on the system device, so the following command can be used:

```
LOAD "DPO.SPS"
```

or simply

```
LOAD "DPO"
```

where "DPO.SPS" is the name of the DPO driver. The LOAD command assumes the .SPS extension.)

When the DPO driver is no longer needed, it may be deleted from memory by executing:

```
RELEASE "DPO"
```

Notice that the .SPS extension is again assumed. (Before the DPO driver can be RELEASEd, all DPOs must be DETACHed from the system -- as described in this section.)

Putting Instruments On-Line

Normally, your system will have been cabled and installed on-site by a Tektronix Field Engineer. In this case, the system should be operational as soon as the master power switch is turned on. Even so, it is necessary to understand a few facts about the strapping options which allow the controller (and software) to communicate with a particular DPO.

Since TEK SPS BASIC supports an instrument configuration with up to four DPOs, there must be some means of selecting a particular instrument via both the hardware and software. In terms of hardware, this

instrument selection is facilitated by a number referred to as the hardware unit number (HUN). The HUN may be thought of as the hardware name by which the controller addresses a particular DPO. Depending on interface strapping options, a DPO may be assigned any HUN between Ø and 31, inclusive; however, no more than one instrument (DPO or R7912) may have the same HUN. Otherwise, there will be a loss of identity and resulting confusion.

The HUN is determined by two sets of straps: The Group Select straps and the Device Select straps. The Group Select straps are located on the CP1100/CP Bus Interface or CP4100/CP Bus Interface (a circuit card inside the controller) and determine the Group Number of the instrument. The Device Select straps are located on the DPO/CP Bus Interface (a circuit card inside the DPO) and determine the Device Number of the instrument. The Hardware Unit Number is thus determined according to Table 3-1. (More information on determining the Hardware Unit Number of an instrument and setting the strap options can be found in the CP1100/CP Bus Interface and CP4100/CP Bus Interface manuals.)

TABLE 3-1

Determining Hardware Unit Numbers

Group Select Strap Settings	0	1	2	3	4	5	6	7
0	0	1	2	3	4	5	6	7
1	8	9	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31

Device Select Strap Settings

1603-13

NOTE

The DPO driver ("DPO.SPS") and the R7912 driver ("TD.SPS") should not be used simultaneously unless each type of instrument is confined to separate CP1100/CP Bus Interface or CP4100/CP Bus Interface cards (inside the controller). This is because the DPO and R7912 use the same interrupt vectors.

ATTACHing and DETACHing a DPO

Assuming that each instrument has been assigned to an unique HUN, there is still another step in establishing communication between the instrument and controller. Specifically, an instrument logical unit number (ILUN) must be assigned to the instrument by the ATTACH command. Until an instrument has been ATTACHed to the system, the software will not allow the controller to recognize the instrument.

The form of the ATTACH command used by the DPO Driver is:

ATTACH #ilun **AS DPO**[hun]**:**

where the ILUN is the instrument logical unit number assigned to the DPO and the HUN is the hardware unit number of the DPO. For example:

ATTACH #5 AS DPOØ:

assigns an instrument logical unit number of 5 to the DPO with a hardware unit number of Ø. Hereafter, this DPO is always referenced as #5 until it is DETACHed from the system.

The form of DETACH is:

DETACH { **#**ilun / **ALL** }

Thus the statement:

DETACH #5

will dissociate the instrument from the ILUN 5. Therefore, an ILUN of 5 could be assigned to some other instrument. While the statement:

DETACH ALL

dissociates every attached instrument in the system from the ILUN assigned to it. When a DPO is DETACHed, no more communication can occur with it until it is ATTACHed again.

Selecting the DPO Display Source

The Display Source of the DPO CRT may be selected either manually or under program control. To select the Display Source remotely, use the following form of PUT:

PUT " { **PI** / **BOT** / **MEM** } **" INTO #**ilun

where the ILUN is the instrument logical unit number of an ATTACHed DPO. This selects the PLUG-INS ("PI"), BOTH ("BOT"), OR MEMORY ("MEM") Display Source. For example, the following statement selects the MEMORY Display Source of the DPO ATTACHed as ILUN 5.

```
PUT "MEM" INTO #5
```

This is the same as manually pushing the MEMORY button of the DPO Display Source group.

Waveform Acquisition and Transfer

Waveform storage in the P7001 memory is accomplished with the PUT command. This operation can also be performed manually by using the STORE, MEMORY LOCATION, START, and HOLD buttons on the DPO. Assuming that the P7001 contains a 4K memory, up to four waveforms may be stored in the DPO at one time. Regardless of the method of storage, three classes of information are stored for each waveform:

1. 512 unscaled mantissa values.

2. Up to 8Ø characters of ASCII text corresponding to the readout for the vertical and horizontal scale-factor information.

3. A zero-reference value.

Before discussing how to store waveforms under program control, let's review how to store a waveform with the P7001 pushbuttons. In this and further explanations it is assumed that the DPO driver is LOADed and the DPO is on-line and ATTACHed.

Storing Waveforms with P7001 Pushbuttons. With the P7001 in the PLUG-INS mode, the first step in storing a waveform from the plug-ins is to input and display the waveform on the DPO. Once you have determined that the waveform is indeed the one you want to store, it may be stored in the P7001 by following this sequence:

1. Press STORE. Note that the DISPLAY SOURCE indicator lights change from PLUG-INS to BOTH, due to the hardware in the DPO. (When BOTH is lighted, the DPO will display the real-time waveform from the plug-ins as well as any waveforms stored in DPO memory.)

2. Press the MEMORY LOCATION button(s) corresponding to the desired P7001 storage location(s). For example: if button A is pressed, the waveform will be stored in P7001 memory location A. Allowable memory locations for multiple waveform storage are specified in the DPO Operators Manual.

3. Press START. This begins storing values in the P7001 memory.

4. Press HOLD after the waveform has been satisfactorily displayed on the CRT. Pressing HOLD causes a message to be displayed on the CRT asking you to provide a zero reference by grounding the probe and pressing PROGRAM CALL button 14. This zero-reference request (generated by the DPO driver) appears for each waveform stored; it may be satisfied by performing either step 5a or 5b below.

5. Perform step 5a or 5b once for each waveform stored.

5a. Press PROGRAM CALL button 14 after grounding the probe. This determines the ground (zero-reference) value and stores it in the P7001 memory.

-or-

5b. Press some other PROGRAM CALL button or any other button that generates an interrupt (i.e., HOLD, START, A, B, C, D). This uses the previously stored zero reference as the zero reference for the newly stored waveform. It also causes the interrupt (generated by the PROGRAM CALL button) to be processed by the controller if a WHEN statement that allows that interrupt to be recognized has been executed.

In the described sequence of P7001 pushbutton operations, pressing START causes the selected memory location to be cleared of non-zero mantissa values. The existing scale-factor information is overwritten by the scale-factor information for the waveform being stored. (The previously stored zero reference is left undisturbed until later options are taken.) More importantly, the waveform is pseudorandomly sampled at 6.5 microsecond intervals (under optimum conditions), and the resulting horizontal and vertical amplitude samples are digitized.

Each horizontal sample that occurs corresponds to one of 512 possible mantissa addresses for the chosen storage location. These mantissa addresses are arranged so that the first address (P7001 array element Ø) corresponds to the left-hand CRT graticule line, and the last mantissa address (P7001 array element 511) corresponds to the right-hand CRT graticule line.

As each pseudorandom sample is taken during the horizontal sweep time, the vertical amplitude of the waveform at the sample point is digitized and expressed as a 1Ø-bit unscaled mantissa value. This mantissa value represents the amplitude value relative to a digitizing reference that exists one division below the bottom CRT graticule line. There are 1Ø24 possible amplitude levels available for expression of the mantissa value, and the mantissa values are stored in the P7001 as unscaled integers in the range of Ø to 1Ø23. (The stored vertical scale factor provides vertical scaling information.) A mantissa value of Ø corresponds to one division below the bottom CRT graticule line, a mantissa value of 1Ø23 corresponds to one division above the top CRT graticule line. (See Fig. 3-1.)

When HOLD is pressed, the sampling and digitizing ceases and the existing digitized samples are fixed in the selected P7001 memory location. Also, after pressing HOLD, a software generated request for zero reference appears on the DPO CRT. When this request is acknowledged by grounding the probe and pressing PROGRAM CALL button 14 (step 5a), 32 samples of the provided ground reference are sampled and digitized. The average of these 32 samples is then used to replace the previously existing zero-reference value. If step 5b is executed instead, the existing zero-reference value is retained.

When more than one trace is stored, either manually or under program control, the zero-reference level must be acquired for <u>each</u> memory location in which data is stored. For each trace stored, the CRT displayed zero-reference request must be followed precisely so that the

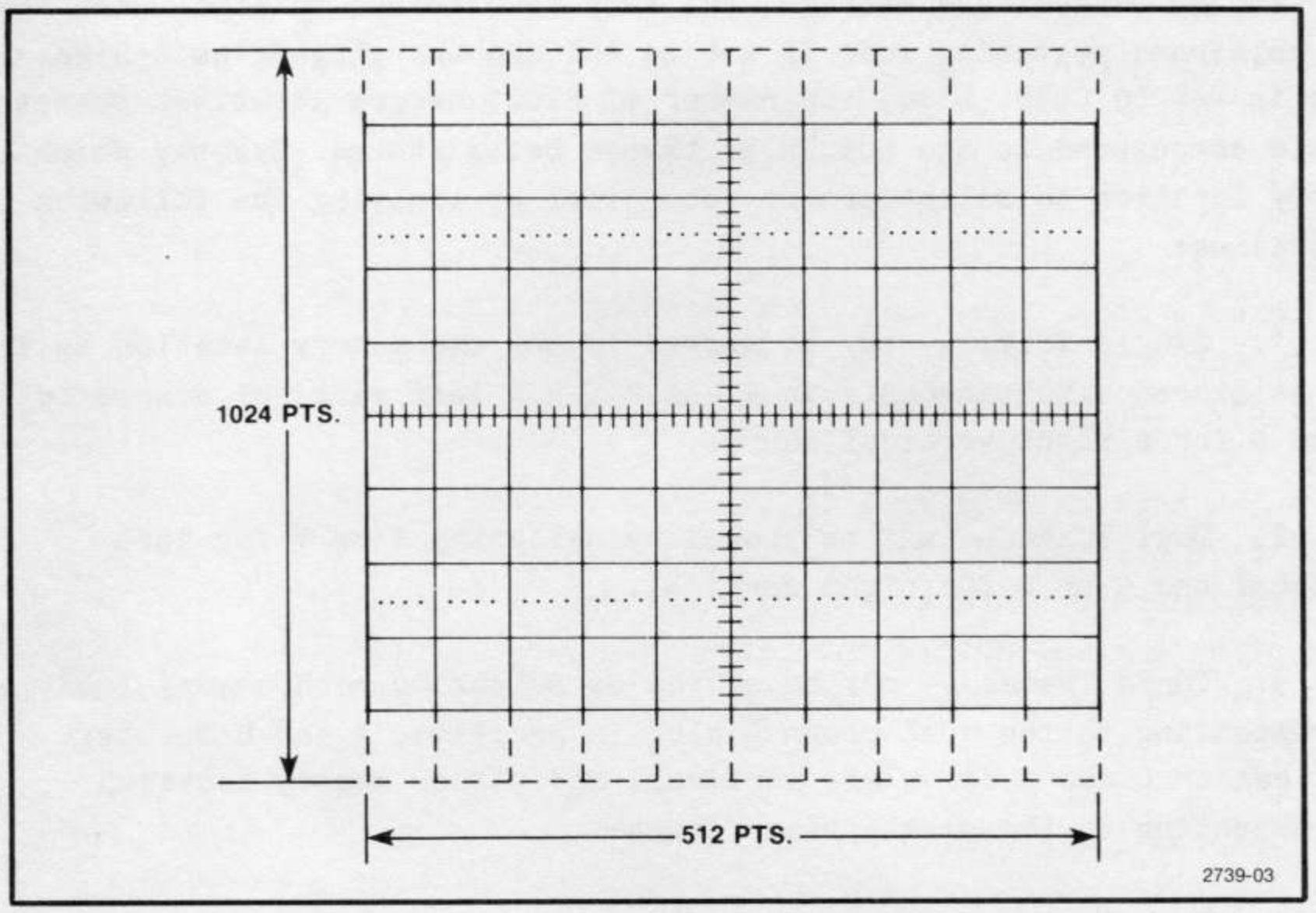


Fig. 3-1. Diagram of DPO CRT. Top and bottom divisions shown with dotted lines are not displayed.

correct zero-reference values are individually collected and stored with their respective waveforms.

For example, when multiple waveforms are stored with the DPO, a message of the folowing type will appear on the CRT:

TO STORE ZERO REF FOR MEM LOC A SELECT
PROPER CHANNEL. GROUND PROBE. PRESS 14.

For this message, you would set the DPO mainframe to LEFT vertical mode and set the left vertical plug-in to CHANNEL 1 mode if it is a dual-trace unit. Then ground the probe and press PROGRAM CALL Button 14. The zero reference for each memory location must be supplied by appropriately setting the mainframe VERTICAL MODE buttons (located at the top of the A7704) and the plug-in DISPLAY MODE switch (located on dual-channel plug-ins only). (They are set solely for the waveform stored in the memory location being supplied with zero-reference information.)

For multiple-trace storage, the best results are obtained when the DPO mainframe switching mode is set to ALT and the plug-in switching mode is set to CHOP. Also, the number of P7001 memory locations selected should correspond to the number of traces being stored. Exactly which memory location to select can be determined by applying the following guidelines:

1. Single Trace -- may be stored in any one memory location or it may be stored simultaneously in A and B for a left vertical source or in C and D for a right vertical source.

2. Dual Trace -- may be stored by selecting A or B for left vertical and C or D for right vertical.

3. Three Traces -- may be stored by selecting both memory locations corresponding to the dual channel plug-in position, A and B for left vertical or C and D for right vertical, and either memory location corresponding to the single trace source.

4. Four Traces -- may be stored by selecting all memory locations simultaneously. Left vertical CH 1 will go to A and CH 2 to B. Right vertical CH 1 will go to C and CH 2 to D.

These rules should be followed for both multiple-trace storage and multiple-trace signal averaging (explained shortly).

Waveform Storage with the PUT Command. The just described sequence of pushbutton operations may be performed under program control with the PUT command. To initiate the STORE operation in the DPO, the following form of PUT may be used:

PUT "STO" INTO #ilun,"$\begin{Bmatrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{Bmatrix}\left[\,,\begin{Bmatrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{Bmatrix}\right]$..."

where the ILUN is the instrument logical unit number of an ATTACHed DPO. In this and other command forms shown in this section, **the string constant in quotes may be replaced by an equivalent string expression.**

This command sets all four status registers (front panel, A/D, readout, and display generator) of the ATTACHed DPO to the STORE more. The second argument in quotes specifies which memory location(s) (A, B, C, and/or D) are to be set. For example:

```
1ØØ PUT "STO" INTO #5,"A,C"
```

sets memory locations A and C of the DPO ATTACHed as ILUN 5 to the STORE mode. The 512 waveform samples will be continuously updated until DPO memory locations A and C are set to the HOLD mode. The DPO front panel lights will indicate that waveforms are being stored in memory locations A and C.

To display a DPO memory location or to stop the storing of a waveform, set the corresponding memory location(s) to the HOLD mode. This is done with the following form of the PUT command:

PUT "HOL" INTO #ilun,**"** $\left\{\begin{matrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{matrix}\right\}\left[\left[\ ,\left\{\begin{matrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{matrix}\right\}\right]\right]$ **...**"

where the ILUN is the instrument logical unit number ATTACHed to the DPO.

This command sets all four status registers of the ATTACHed DPO to the HOLD mode. The DPO front panel lights will indicate which memory locations are being held for display. For example:

```
11Ø PUT "HOL" INTO #5,"A,C"
```

sets memory locations A and C of the DPO ATTACHed as ILUN 5 to the HOLD mode, thereby terminating the storage process begun at line 1ØØ above. However, **the waveform will be stored without regard for the zero-reference level**. (That is, the previous zero-reference level will be used.)

To store a waveform corrected for the new ground-reference level, execute the following command form in place of PUT "HOL":

PUT "ZHO" INTO #ilun,**"** $\left\{\begin{matrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{matrix}\right\}\left[\left[\ ,\left\{\begin{matrix}\mathbf{A}\\\mathbf{B}\\\mathbf{C}\\\mathbf{D}\end{matrix}\right\}\right]\right]$ **...**"

If the PUT "STO" command was previously issued and this command is executed, the storing ceases and a message requesting a zero reference is displayed on the DPO CRT (assuming the DPO is in the BOTH or MEMORY display mode). To store a zero reference, ground the probe (or the plug-in) and press PROGRAM CALL button 14 on the DPO. This procedure should be followed once for each memory location specified. If you press any other button that can cause an interrupt, the previous zero reference is kept and the message is erased. For multiple-trace storage, the previously mentioned rules apply (see "Storing Waveforms with P7001 Pushbuttons").

In general, it is best to put a delay between the STORE and HOLD operations. The delay should be long enough to allow the waveform to be stored in its entirety before the HOLD process starts. Also, the program should suspend execution after a PUT "ZHO" statement until call button 14 is pressed. The WAIT command is suited for either purpose, as demonstrated in the following acquisition routine. (A general purpose zero-reference routine using a WHEN "CB14" statement is shown in "Handling DPO Interrupts.")

```
200 PUT "STO" INTO #5,"C"
210 WAIT 500
220 PUT "MEM" INTO #5
230 PUT "ZHO" INTO #5,"C"
240 PRINT "PERFORM GROUND-REFERENCE ACQUISITION SEQUENCE"
250 PRINT "WHEN DONE, PRESS RETURN TO CONTINUE"
260 WAIT
    .
    .
```

Line 200 puts DPO #5 into a mode which stores the input waveform into memory location C. (The PLUG-INs and STORE-C lights of the DPO are lit to indicate this condition.) Line 210 causes the controller to pause approximately 500 milliseconds before executing line 220. Line 220 sets the DPO display source to MEMORY so that the zero-reference request message (generated at line 230) can be seen. When line 230 is reached, the storage process ceases as indicated by the lighting of the HOLD button. Also, since a "ZHO" argument was specified, a message is displayed on the DPO CRT requesting the user to supply a ground-reference level. When PROGRAM CALL button 14 is pressed, the current ground-reference level is acquired by the DPO driver and stored in DPO memory; it will be used to adjust the waveform values when a waveform is acquired into controller memory with a GET statement. The second WAIT (line 260)

provides time to perform the ground-reference acquisition sequence. Pressing the RETURN key continues the program.

Transferring Waveforms into Controller Memory. After a waveform has been stored in DPO memory, it may be transferred to a controller array or waveform with the GET command. The form of GET used for this purpose is:

GET { floating-point array / floating-point waveform } [, { floating-point array / floating-point waveform }] ...

FROM #ilun,"{ **A** / **B** / **C** / **D** }" [, "{ **A** / **B** / **C** / **D** }"] ...

The target argument must be either a 512-element floating-point array or a 512-element floating-point waveform. The ILUN specifies the instrument logical unit number of the DPO from which the data is to be acquired. The string expressions select which DPO memory locations are to be the source of the data. Since there is a one-to-one correspondence between target arrays or waveforms and the source memory locations, there must be an equal number of each.

Each of the target arrays or waveforms is filled with "normalized" data. This means first that any points not stored are interpolated or extrapolated to provide a complete set of waveform samples. Then the zero-reference is subtracted from each sample and the resulting values are multiplied by the vertical scale factor. Finally, if the destination argument is a waveform, the data sampling interval, horizontal units and vertical units are returned in appropriate variables.

As an example of waveform acquisition, let's suppose that memory locations A and C of a DPO ATTACHed as ILUN 5 contain waveforms. To read these waveforms into controller memory, the following routine could be used:

```
300 WAVEFORM WA IS A(511),SA,HA$,VA$
310 WAVEFORM WC IS C(511),SC,HC$,VC$
320 GET WA,WC FROM #5,"A","C"
```

Line 300 defines waveform WA to be a 512-element array A, with a data sampling interval SA, and horizontal and vertical units HA$ and

VA$, respectively. Similarly, line 31Ø defines waveform WC to be a 512-element array C, with a data sampling interval SC, and horizontal and vertical units HC$ and VC$. Notice that arrays A and C are dimensioned by the WAVEFORM statements. With waveforms WA and WC now defined, line 32Ø GETs the waveforms stored in memory locations A and C. The waveform in DPO memory location A is read into array A in the controller, and the waveform in DPO memory location C is read into array C in the controller. The units and data sampling intervals are also acquired in the process.

Sending Data to the DPO. The PUT command can be used to transfer an array or waveform to the DPO. When used for this purpose, the form of PUT is:

PUT {array expression / waveform expression} [, {array expression / waveform expression}] ...

INTO #ilun " {**A** / **B** / **C** / **D**} " [, " {**A** / **B** / **C** / **D**} "] ...

The source must be either a 512-element array expression or a 512-element waveform expression. The ILUN specifies the instrument logical unit number of the target DPO. The string expressions are the memory locations which serve as the destinations for the transferred data. For example:

```
PUT WA,WC INTO #5,"B","D"
```

sends waveform WA to memory location B and waveform WC to memory location D of a DPO ATTACHed as ILUN 5. Memory locations B and D could then be displayed on the DPO CRT with the concatenated statement:

```
PUT "MEM" INTO #5\PUT "HOL" INTO #5,"B,D"
```

which first sets the DPO Display Source to MEMORY and then sets Memory Locations B and D to HOLD.

DPO Signal Averaging. TEK SPS BASIC provides two modes of stable signal averaging for DPO waveforms -- a normal mode and a fast mode. The normal mode of signal averaging clears each sample from the P7001 Processor after it has been added to the average. Then it checks the next sample to ensure that it is valid (i.e., the waveform has been sampled

and a nonzero sample provided) before adding it to the average. The fast averaging mode does not check the validity of each sample; hence, it executes faster than normal averaging. However, each point in fast averaging may deviate from the true signal amplitude at that point. Thus, the fast-average mode should be used for quick previews only.

To signal average waveforms from a DPO, a zero reference should be stored for each memory location in which data is averaged. This may be done by first storing the signal to be averaged in one of the P7001 memory locations. Then, following the message displayed on the DPO CRT at the time of waveform storage, ground the probe and press PROGRAM CALL Button 14.

Normal averaging requires that the waveform being averaged occupy a full ten centimeters horizontally on the CRT of the D7704 DISPLAY UNIT. If the real-time waveform display is not occupying the full ten centimeters of graticule, the horizontal position control on the Time-Base plug-in should be adjusted to provide the required display. If this is not done, the averaging operation will not complete, and a restart from location Ø will be required for an exit from the averaging routine.

With the zero-reference value(s) stored and the real-time waveform applied to selected vertical plug-ins of the DPO, the average operation may be begun. An averaged DPO signal is obtained with the GET command as follows:

GET { floating-point array / floating-point waveform } [, { floating-point array / floating-point waveform }] ...

FROM #ilun, **"AVE**n[**F**], { **A** / **B** / **C** / **D** } [, { **A** / **B** / **C** / **D** }] ..."

The target arrays are either 512-element floating-point arrays or 512-element floating-point waveforms. The ILUN is the instrument logical unit number of the source DPO. The "n" in the source argument specifies the number of averages; it must be an integer between 1 and 32767, inclusive. Up to four signals can be simultaneously averaged depending on which combinations of memory locations (A through D) are specified. If the "F" is included in the string expression, a fast average is performed. This means that the driver does not wait for the DPO to store a point before including that point in the average as explained previously.

The averaging operation first sets the DPO to store mode. After the specified number of complete waveform samples have been acquired and averaged, the DPO is returned to the mode it was in before the start of the averaging operation. The data returned is normalized. If a waveform is specified as the destination, the data sampling interval, horizontal units, and vertical units are also returned.

The averaging operation uses the previously stored zero-reference value in normalizing the averaged data. Thus, you will probably want to store a zero-reference value for the memory location(s) whose data is to be averaged. For example, the following routine stores a zero-reference value for memory location C and then averages data (16 times) in that memory location. This program assumes that the DPO driver has been LOADed and that a DPO is ATTACHed as instrument logical unit number 1.

```
1Ø REM PROGRAM TO ACQUIRE AN AVERAGED WAVEFORM
2Ø PUT "PI" INTO #1
3Ø PRINT "INPUT WAVEFORM TO VERTICAL PLUG-IN"
35 PRINT "WAVEFORMS MUST SPAN 1Ø DIVS"
4Ø PRINT "UNGROUND PROBE OR PLUG-IN AND ADJUST SIGNAL"
5Ø PRINT "PRESS RETURN WHEN READY"
6Ø WAIT
7Ø WAVEFORM WC IS C(511),SC,HC$,VC$
8Ø PUT "STO" INTO #1,"C"
85 WAIT 5ØØ
9Ø PUT "MEM" INTO #1
1ØØ PUT "ZHO" INTO #1,"C"
11Ø PRINT "TO ACQUIRE GROUND-REFERENCE LEVEL,"
115 PRINT "GROUND PROBE OR PLUG-IN AND"
12Ø PRINT "PRESS CALL BUTTON 14 ON DPO"
125 PRINT "PRESS RETURN WHEN READY"
13Ø WAIT
14Ø PUT "PI" INTO #1
15Ø PRINT "TO ACQUIRE AVERAGED WAVEFORM,"
155 PRINT "UNGROUND PROBE OR PLUG-IN"
16Ø PRINT "PRESS RETURN WHEN READY"
17Ø WAIT
18Ø GET WC FROM #1,"AVE16,C"
19Ø RETURN
```

The program begins by setting the DPO to PLUG-INS display mode (line 2Ø). Lines 3Ø through 5Ø instruct the user to input the waveform and set up the DPO for proper triggering and display. After the WAIT is exited by a keyboard interrupt (such as pressing the Return key), waveform WC is

defined. AT line 8Ø, the DPO is set to store in Memory Location C, and the display source is changed to MEMORY (for display of the zero-reference request message). At line 1ØØ the storage process terminates with the zero-reference request message. This message is reiterated on the terminal screen. After the probe or plug-in is grounded and PROGRAM CALL button 14 is pressed, pressing the RETURN key exits the WAIT.

At this point the user is instructed to unground the probe and to press RETURN when ready. Finally, line 18Ø signal averages the input waveform 16 times into waveform WC via memory location C.

The same basic program could be modified for multiple signal averaging. The important difference is that a zero-reference level must be acquired for **each** memory location from which data is averaged. When multiple traces are stored, the CRT displayed zero-reference request must be followed precisely so that the correct zero-reference values are individually collected and stored with their respective waveforms. Also, the previously mentioned rules for multiple-trace storage also apply to multiple-trace signal averaging.

The question sometimes arises as to how many averages should be specified. There is no one answer to this question. However, as a rule of thumb, several hundred to several thousand averages may be required to recover signals nearly buried in noise. For signals with moderate noise, 128 averages may be enough. With low noise, 32 may be all that is required. In any case, the exact number of averages required is dependent upon the desired improvement in signal-to-noise ratio.

Summing a signal M times improves the signal-to-noise ratio (S/N) by $\sqrt{M}$. If M can be expressed as a power of 2, then the improvement in the S/N ratio will correspond to 3dB per power of 2. TEK SPS BASIC signal averaging allows a maximum of $2^{15}-1$ (32767) averages, which corresponds to approximately 15x3dB or 45dB maximum improvement in the S/N ratio.

Reading the A/D Converter. In some applications it is particularly advantageous to read the last value processed by the DPO Analog-to-Digital Converter. This value can be obtained with the following form of the GET command:

GET variable **FROM #**ilun,**"AD"**

The value returned in the specified variable is an integer between Ø and 1Ø23, inclusive. The ILUN is the instrument logical unit number of the ATTACHed DPO.

One of the more useful applications of this form of GET is acquiring long-term records of slowly varying phenomena. This can be done by enclosing the GET command in a FOR loop. For example, to acquire a 512-element array X, the following routine could be used to acquire data from a DPO ATTACHed as ILUN 1.

```
1Ø DIM X(511)
2Ø FOR I=Ø TO 511
3Ø GET X(I) FROM #1,"AD"
4Ø NEXT I
5Ø RETURN
```

The execution time (i.e., acquisition time) can be lengthened by including a WAIT statement in the loop and specifying the desired number of milliseconds. After a period of trial and error, the loop can be adjusted to match the desired acquisition time. Of course the data returned in array X is in integer form; thus you may want to normalize it and adjust it for the ground-reference level before processing it.

Setting and Reading DPO Status

Arming the Single Sweep. There is only one programmable feature of the DPO plug-ins. This is the single-sweep function on the time base. The single-sweep feature of the DPO time base can be armed with a PUT command of the following form:

PUT "SSA" INTO #ilun

where ILUN is the instrument logical unit number of an ATTACHed DPO.

Reading the Status of the Single Sweep. To read the status of the single sweep, use a GET command of the following form:

GET string variable **FROM #**ilun**,"SWE"**

where ILUN is the instrument logical unit number of an ATTACHed DPO. The sweep status is returned in the specified string variable as one of the following strings:

String returned	Meaning
"SSU"	single sweep unarmed
"SSA"	single sweep armed

As examples, suppose you want to arm the sweep of a DPO ATTACHed as ILUN 5. This could be done with the statement:

```
PUT "SSA" INTO #5
```

You could read the sweep status with the statement:

```
GET A$ FROM #5,"SWE"
```

An "SSA" or "SSU" would be returned in A$ depending on whether the sweep was armed or unarmed at the time the GET command was executed.

Handling DPO Interrupts

Recognizing DPO Interrupts. The WHEN command allows a BASIC program to recognize and respond to an instrument interrupt and to schedule a user-written subroutine when the interrupt occurs. Various DPO-generated interrupts can be recognized after the appropriate WHEN statements are executed.

The form of the WHEN command used by the DPO driver is:

WHEN #ilun **HAS** interrupt specification [**AT** priority number]
[**AS TASK** task number] **GOSUB** line number

where the ILUN is the instrument logical unit number of an ATTACHed DPO.

Any one of the following strings may be used as the interrupt-specification string to define the type of interrupt to be recognized:

"SND {**A** / **B** / **C** / **D**} **"**	This type of WHEN command schedules the interrupt routine when a sequence of SEND-MEMORY LOCATION (A, B, C, or D)-START buttons are pressed on the DPO front panel. More than one waveform button can be pressed and recognized if the appropriate WHENs are set.
"CB1" through **"CB15"**	This schedules the interrupt routine when the corresponding DPO PROGRAM CALL button is pushed. "CB14" can be used for terminating a zero-reference acquisition as well as for generating an interrupt.

"REC {**A** / **B** / **C** / **D**} **"**

This works the same way as SEND except that the RECEIVE button is recognized instead of SEND.

"SSC"

This schedules the interrupt routine when the single-sweep-complete interrupt occurs. It is only recognized if the single sweep was armed with a PUT "SSA" command. (Manual arming is not recognized.)

The priority number specifies the priority at which the interrupt routine will be executed. It may range from a high of 126 to a low of Ø. If none is specified, the default priority of the driver (51) is assumed. When the specified event occurs, the corresponding interrupt routine (starting at the specified line number) is executed if its priority is higher than the system priority. (System priority ranges from a high of 126 to a low of Ø; it defaults to 5Ø unless modified by a PRIORITY command or an interrupting routine.) The task number specifies the task to be associated with the interrupt routine. The line number is the starting line of the interrupt routine. For more information on interrupt processing see "Instrument Communications" in Section 3 of the TEK SPS BASIC VØ2 System Software manual.

Ignoring DPO Interrupts. The IGNORE command cancels the actions of one or more WHEN statements. After a WHEN is canceled, the occurrence of the interrupt condition specified in the WHEN no longer causes an interrupt subroutine to be scheduled for execution. However, IGNORE has no effect on the execution of any already scheduled interrupt routines.

The form of the IGNORE command used by the DPO driver is:

IGNORE { #ilun [, { **TASK** task number / interrupt specification }] / **TASK** task number / **ALL** }

where the ILUN is the instrument logical unit number of an ATTACHed DPO. The interrupt specification must be one of the strings listed with the WHEN command.

If only the ILUN is specified, all WHEN statements with that ILUN are canceled. If the interrupt specification string is given with the ILUN, only the WHEN statement that specifies both is canceled. If the task number is specified with the ILUN, any WHEN statements with that ILUN and task number are canceled. If only the task number is given, any WHENs with that task number are canceled. If the keyword **ALL** is specified, all WHEN statements are canceled for all instruments.

As examples, consider these WHEN statements:

```
1ØØ WHEN #2 HAS "SSC" GOSUB 5ØØ
11Ø WHEN #2 HAS "CB14" AS TASK 25 GOSUB 6ØØ
12Ø WHEN #2 HAS "RECB" AS TASK 25 GOSUB 7ØØ
```

The statement:

```
IGNORE #2
```

cancels the WHENs in lines 1ØØ, 11Ø, and 12Ø. The statement:

```
IGNORE #2,"SSC"
```

cancels only the WHEN in line 1ØØ, and the statement:

```
IGNORE #2,TASK 25
```

cancels the WHENs in line 11Ø and 12Ø.

One of the uses of the WHEN command is recognizing interrupts from the DPO PROGRAM CALL buttons. These buttons were so named because they can be programmed to begin program execution at a particular line number. This is particularly useful where you have a series of short routines which must frequently be executed. By pressing the appropriate button you can immediately select the next routine to be executed.

As an example of programming the PROGRAM CALL buttons, consider the following routine. This assumes that the DPO is attached as ILUN 1.

```
1ØØ WHEN #1 HAS "CB1" GOSUB 2ØØ
11Ø WHEN #1 HAS "CB2" GOSUB 21Ø
12Ø WHEN #1 HAS "CB3" GOSUB 22Ø
13Ø WHEN #1 HAS "CB15" AT 126 GOSUB 3ØØ
15Ø FL=Ø
16Ø IF FL=Ø THEN 16Ø
17Ø END
```

```
200 PRINT "BUTTON - 1"\RETURN
210 PRINT "BUTTON - 2"\RETURN
220 PRINT "BUTTON - 3"\RETURN
300 IGNORE ALL
310 FL=1
320 RETURN
```

The program begins by designating interrupt routines for call buttons 1 through 3. These interrupt routines have equal priority. (Since none is specified, the priority of 51 is used by default.) The associated interrupt routines are found in lines 200 through 220. These simply print which button was pressed. Line 130 sets up an interrupt routine for call button 15, which in this case functions as a button to terminate the program. Since the routine it calls has a priority of 126 it can be used to interrupt any other routines of lesser priority.

Line 150 sets a program control flag (FL) to Ø. Then the program loops at line 160 until one of the four defined call buttons is pushed to schedule an interrupt routine. As the scheduled routine completes, if no other routine is scheduled, program control continues to loop at line 160 as long as FL remains Ø.

The routine called by call button 15 executes an IGNORE ALL. This cancels the actions of all WHEN statements but has no effect on any interrupt routines already scheduled. The routine also sets FL to 1. After this routine exits, any previously scheduled routines will execute. However, when program control returns to line 160, since FL no longer is Ø, the program ENDs. The program must be RUN again before the PROGRAM CALL interrupts can be recognized.

Another useful function of the WHEN command is in detecting interrupts from the SEND and RECEIVE buttons on the DPO. (The SEND and RECEIVE buttons do not automatically transfer waveforms in TEK SPS BASIC.) For example, the following program defines four waveforms in controller memory. Then when a sequence of SEND-MEMORY LOCATION (A, B, C, or D)-START is pressed, a routine is scheduled which transfers data from the selected memory location to the corresponding waveform in controller memory. Finally, the program enters a busy loop waiting for another interrupt to occur. This program accesses the DPO ATTACHed as ILUN 5.

```
110 WAVEFORM WA IS A(511),SA,HA$,VA$
120 WAVEFORM WB IS B(511),SB,HB$,VB$
130 WAVEFORM WC IS C(511),SC,HC$,VC$
140 WAVEFORM WD IS D(511),SD,HD$,VD$
150 WHEN #5 HAS "SNDA" GOSUB 250
```

```
160 WHEN #5 HAS "SNDB" GOSUB 260
170 WHEN #5 HAS "SNDC" GOSUB 270
180 WHEN #5 HAS "SNDD" GOSUB 280
200 GOTO 200
250 GET WA FROM #5,"A"
255 RETURN
260 GET WB FROM #5,"B"
265 RETURN
270 GET WC FROM #5,"C"
275 RETURN
280 GET WD FROM #5,"D"
285 RETURN
```

In order to interrupt this routine, Control-P can be entered on the terminal. However, since Control-P cancels the actions of all WHEN statements, the program must be RUN again before the SEND buttons can be used to acquire data into memory.

A similar routine could be written which would detect DPO interrupts from a sequence of RECEIVE-MEMORY LOCATION (A, B, C, or D)-START. This could be used to transfer data from controller arrays to designated memory locations in the DPO.

As noted earlier, executing a WHEN "CB14" statement can give call button 14 two functions. CB14 not only terminates the zero-reference acquisition started by a PUT "ZHO" statement, but also causes an interrupt routine to be scheduled. Here is a general purpose zero-reference routine that allows CB14 to have two purposes. The technique shown here eliminates the need for a WAIT statement directly following the PUT "ZHO". Instead, an IF statement with a program control flag prevents the program from continuing until call button 14 is pressed. In this example a DPO is ATTACHed as ILUN 1.

```
100 PUT "PI" INTO #1
110 PRINT "ADJUST SIGNAL; PRESS RETURN WHEN READY"
120 WAIT
130 PUT "STO" INTO #1,"A"
140 WAIT 500
150 FL=0
160 WHEN #1 HAS "CB14" GOSUB 500
170 PUT "BOT" INTO #1
180 PUT "ZHO" INTO #1,"A"
190 IF FL=0 THEN 190
200 PRINT "RESET PLUG-IN TO AC OR DC MODE,
205 PRINT "THEN PRESS RETURN WHEN READY"
```

```
21Ø WAIT
      .
      .
      .
5ØØ IGNORE #1,"CB14"
51Ø FL=1
52Ø RETURN
```

Line 1ØØ displays the incoming signal on the DPO CRT. Then a message is printed and the controller WAITs so the incoming signal can be adjusted and positioned. Pressing the RETURN key continues the program. Next, line 13Ø starts the signal storing in DPO array A and line 14Ø WAITs approximately 5ØØ milliseconds to allow a full waveform to be stored. Then the program control flag (FL) is set to Ø and a WHEN "CB14" is set up. Next, the DPO CRT is set to both memory and plug-in mode so the zero-reference message generated by the PUT "ZHO" in line 18Ø is displayed.

When CB14 is pressed in response to the zero-reference message after the plug-in is grounded, the zero-reference value is stored and the subroutine at line 5ØØ is executed. This subroutine disarms CB14 and changes FL to 1. Thus, until CB14 is pressed, FL equals Ø and the program loops at line 19Ø. After CB14 is pressed, a message to unground the plug-in is printed. Again, pressing RETURN continues the program.

Display and Acquisition of ASCII Text

TEK SPS BASIC also uses the PUT and GET commands to deal with readout information displayed on the DPO CRT. The PUT command is used to display text on the DPO CRT and to label a particular DPO memory location. Conversely, the GET command is used to acquire ASCII readout information from a particular DPO memory location. Before examining each of these operations we need to take a brief look at which characters can be stored and where they are stored in P7001 memory.

The memory of the P7001 Processor consists of two major sections. The first section of memory is dedicated to storing waveforms for display on the DPO CRT. The second section is used for storing scale factors and messages for display on the DPO CRT by the readout interface. This latter section of memory is divided into 16 fields (four fields per waveform location) six of which are available to the user through TEK SPS BASIC.

Four of the available areas, field zero of waveforms A through D, (ØA, ØB, ØC, and ØD) are used for storing scale-factor messages that occur during waveform storage and transfer operations. Also, labeled messages generated by the PUT command are contained within field zero of the waveform location addressed by the PUT command. Field one of waveform A and field one of waveform B are the two remaining areas. Field one of A (1A) is used to store displayed messages generated by the PUT command. Field one of B (1B) is used to store zero-reference requests generated by the controller during waveform storage. Figure 3-2 shows a functional diagram of the 4K P7001 memory. (Refer to the DPO Operators manual for more information on smaller memory options.)

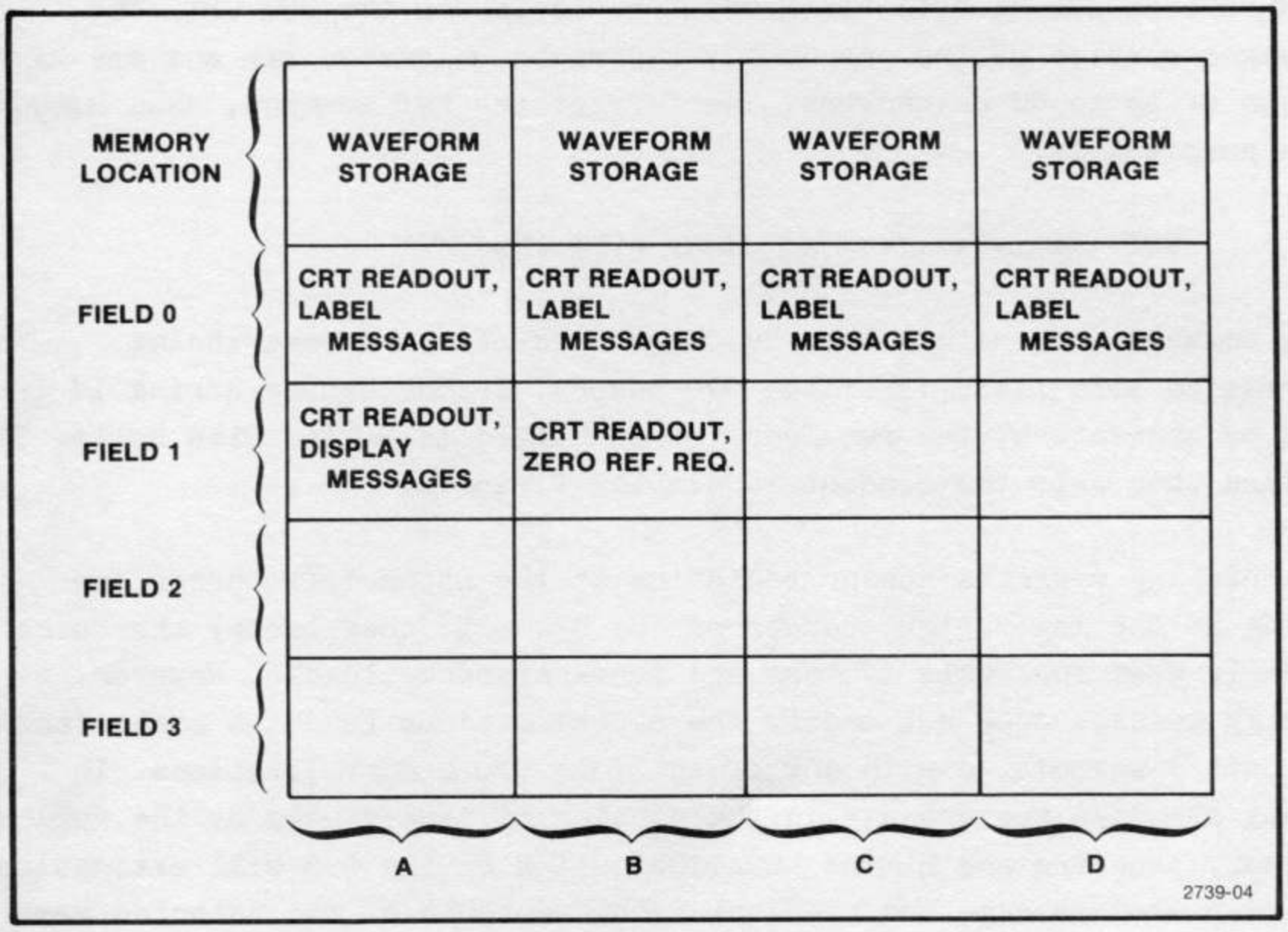


Fig. 3-2. Functional diagram of 4K P7001 memory.

Displayable Characters. The set of characters that can be displayed on the DPO CRT readout is a subset of the full ASCII code set. The characters available are: the digits Ø through 9, all upper-case letters; the lower-case letters c, d, m, n, p, u; and the six characters ., <, /, +, -, and >. There are three additional non-ASCII readout characters available for display by outputting to the display the ASCII characters shown in Table 3-2.

TABLE 3-2

ASCII Characters Used to Display Non-ASCII Characters

ASCII Character	Readout Character
!	↓
=	Δ
@	Ω

2739-05

Displaying Messages on the DPO CRT. The PUT command may be used to send various messages to the P7001 for display on the DPO CRT. The messages consist of the previously described character set and may have a length of up to 8Ø characters. The form of the PUT command, when used for this purpose, is:

PUT string expression **INTO #**ilun,**"DIS"**

This command stores the first 8Ø characters of the source string expression into field 1A of the DPO memory. If the source string is less than 8Ø characters, the remainder of the field is filled with nulls. The command then sets the readout to display field 1A.

Display messages appear beginning at the upper-left corner and ending at the lower-right corner of the DPO CRT; they occupy the space normally used for scale factors and zero-reference levels. However, a display message does <u>not</u> modify the stored readout (such as scale factors and units) associated with any of the four DPO memory locations. It merely displays the message in the portion of screen used by the readout. In fact, pressing any MEMORY LOCATION button on the DPO will extinguish the displayed message and re-display the contents of the selected memory location.

As an example, the following statement will display the message "LASER EXPERIMENTS" in the upper-left corner of the CRT of the DPO ATTACHed as ILUN 1 if the DPO Display Source is set to BOTH or MEMORY:

```
PUT "LASER EXPERIMENTS" INTO #1,"DIS"
```

The message will remain on the CRT until the displayed memory location is changed either manually or under program control. Changing memory locations permanently extinguishes the displayed message; thus to redisplay the message, the PUT command must be re-executed.

Since many computer terminal keyboards have only upper case letters, a special method is used to input the six lower case letters from these keyboards. This method which uses two string function, ASC and CHR, is demonstrated by the following line:

```
PUT CHR(ASC("N")+32) INTO #1,"DIS"
```

This outputs the lower-case letter "n" to the CRT of the DPO ATTACHed as ILUN 1. Similarly the lower-class letters c, d, m, p, and u can be displayed on the CRT by substituting the corresponding upper-case character in the inner pair of parentheses.

Labeling DPO Memory Locations. In some cases, you may want to attach a message to a P7001 waveform that will remain with the waveform for as long as it is stored in a particular memory location. By using a label message, up to 34 characters of information may be written into one of the four memory areas reserved for scale factors and readout. The following form of PUT is used in labeling a memory location:

PUT string expression **INTO #**ilun, **"LAB** { **A** / **B** / **C** / **D** } **"**

This command stores the first 34 characters of the source string expression into one of these selected areas of memory: field ØA (for memory location A), field ØB (for memory location B), field ØC (for memory location C), or field ØD (for memory location D). If the source string is less than 34 characters, the remainder of the locations are filled with nulls. The message appears beginning at the lower-left corner of the DPO CRT and is seen, along with the scale factors and units, whenever the associated memory location is selected for display.

As an example, the following statement outputs the label "EXPERIMENT 1" to the lower-left corner of the CRT of the DPO ATTACHed as ILUN 1:

```
PUT "EXPERIMENT 1" INTO #1,"LABA"
```

The label appears whenever location A is selected, and is not permanently extinguished by switching memory locations.

Acquiring DPO Readout Information. Occasionally a need may arise to input DPO readout information directly from P7001 memory to controller memory. A special form of the GET command allows this. Before discussing this form of GET, however, let's take a brief look at the conventions used in numbering the CRT readout channels.

As seen in Fig. 3-3, there are eight readout channels (each with 1Ø characters) on the DPO CRT. These are numbered Ø through 7 and each one occupies a unique position on the CRT. For example, when a dual-trace vertical plug-in is inserted in the extreme left-hand compartment, the readout for CHANNEL 1 of the plug-in appears in channel Ø on the CRT and the readout for CHANNEL 2 of the plug-in appears in channel 4 on the CRT. On the other hand, when a horizontal (time base) plug-in is inserted in the extreme right-hand compartment, the sweep speed setting appears in channel 3 on the CRT.

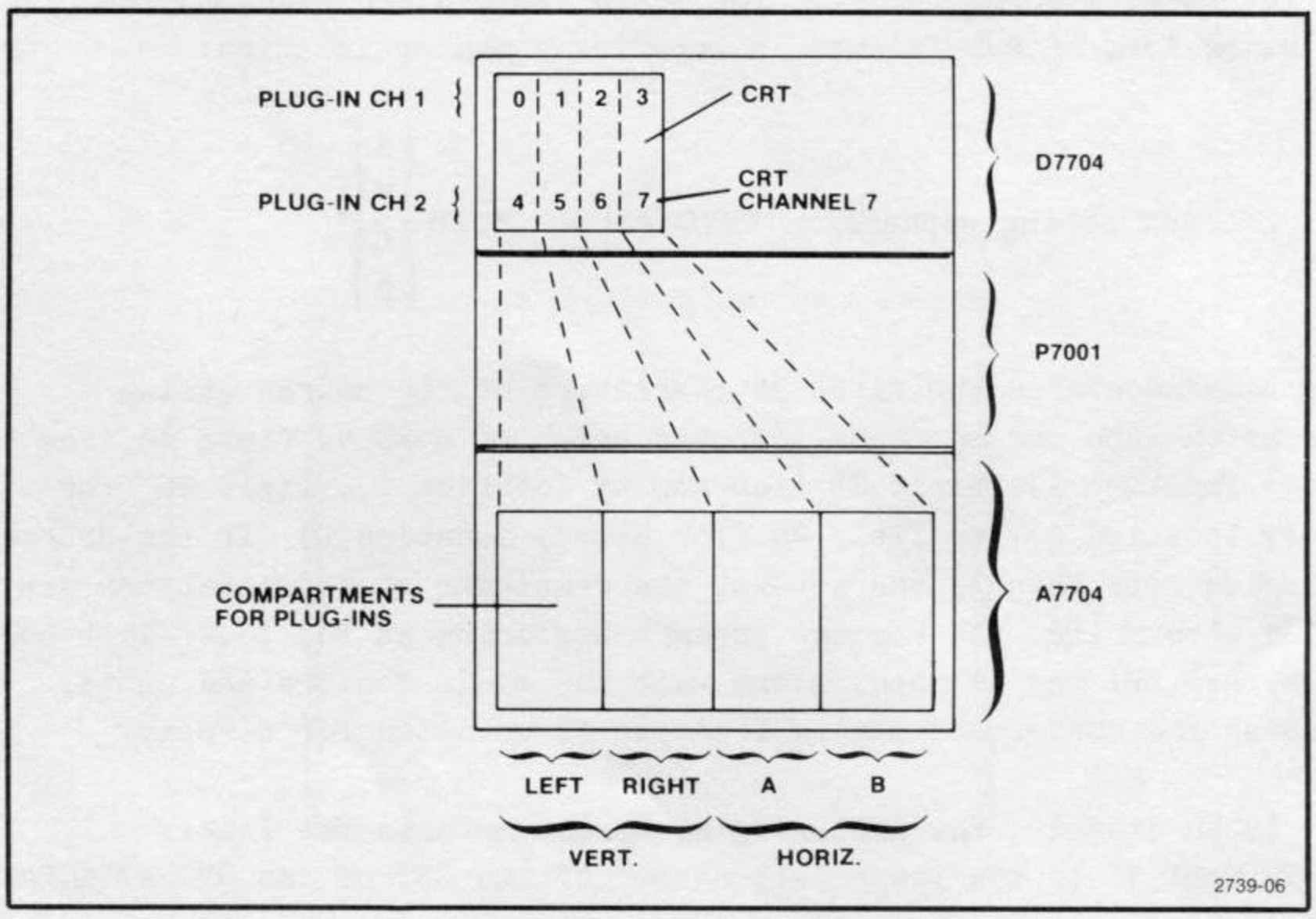


Fig. 3-3. DPO CRT readout conventions.

Each ASCII readout channel of the four DPO waveforms can be read using the GET command as follows:

GET { variable / string variable } **FROM #**ilun,**"CH** { **A** / **B** / **C** / **D** } channel number[,number][**L**]**"**

where the ILUN is the instrument logical unit number of an ATTACHed DPO. The A, B, C, or D following the CH in the string specifies which DPO Memory Location is the source. The channel number specifies which CRT readout channel is read (Ø, 1, 2, 3, 4, 5, 6, or 7). If two channel numbers are specified, the contents of the corresponding readout channels are concatenated. This allows long strings of readout information from digital plug-ins to be processed. When a variable is specified as the target, the numeric portion of the readout is returned. However, if the "L" argument is included, only the four least significant digits that appear in the specified readout are acquired. When a string variable is specified as the target, all characters (except spaces) after the numeric portion of the channel readout are returned.

To illustrate this use of the GET statement, suppose that a waveform in Memory Location A has a vertical scale factor of 5 mV in CRT-channel Ø, a horizontal scale factor of 2Ø ms in CRT-channel 2, and a zero reference of -3 DIV in CRT-channel 7. This is shown in Fig. 3-4.

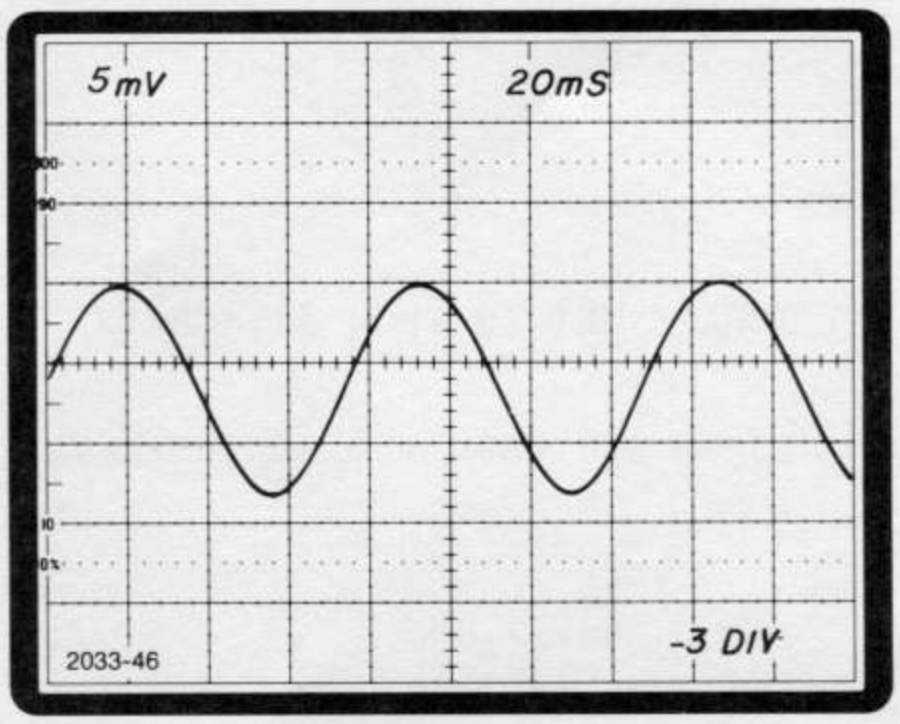


Fig. 3-4. Example of CRT readout on DPO.

Under this condition, it is possible to GET the scale-factor information into controller memory and PRINT it on the terminal. Each of the following three examples illustrates this by showing the statements and the resulting output:

```
GET X FROM #1,"CHAØ"\GET A$ FROM #1,"CHAØ"\PRINT X;A$
5.ØØØØØE-Ø3V

GET Y FROM #1,"CHA2"\GET B$ FROM #1,"CHA2"\PRINT Y;B$
.Ø2S

GET Z FROM #1,"CHA7"\GET C$ FROM #1,"CHA7"\PRINT Z;C$
-3DIV
```

The first example GETs the scale-factor information in CRT-channel Ø, of Memory Location A, storing the floating-point value in variable X and the units string ("V" for volts) in A$. Note that the m (for milli) prefix is inherent in the floating-point value and therefore is not entered in A$. Similarly, the second example selects CRT-channel 2 and stores the floating-point value in Y and the units string ("S" for seconds). The third example GETs the baseline information stored in CRT-channel 7 and stores it in Z with the units string ("DIV" for divisions) in C$.

Consider another example where the readout from a TEKTRONIX 7D13 Digital Multimeter is stored in CRT-channel 1 of Memory Location A. If the readout of the multimeter is:

```
1Ø.15 ohms
```

the the GET statements:

```
GET X FROM #1,"CHA1"\GET A$ FROM #1,"CHA1"
```

would store the readout from it. When the statement:

```
PRINT X,A$
```

is executed, the following output may be seen:

```
10.15@
```

Notice that the ohm symbol was changed to @ as noted in Table 3-2.

YOUR COMMENTS COUNT

The Manual Writers at Tektronix, Inc. are interested in what you think about this manual, how you use it, and changes you might like to see in future manuals. Any queries regarding this manual will be answered personally.

What did you find that was:

interesting? _____

_____

frustrating? _____

_____

helpful? _____

_____

confusing? _____

_____

Is there anything you would like to see added to or deleted from this manual? _____

_____

_____

_____

What is your major application area for this product? _____

_____

_____

_____

Have you found any interesting applications, operating hints, or software routines which you would like to share with us? _____

_____

_____

_____

_____

_____

_____

_____

_____

* * * * *

Name: _____ Position: _____

Company: _____ Department: _____

Street: _____

City: _____ State: _____ Zip: _____

Fold on dotted lines and tape.
Postage will be paid by Tektronix, Inc. if mailed in U.S.A.

FIRST CLASS
PERMIT NO. 61
BEAVERTON, OREGON

BUSINESS REPLY MAIL

No postage necessary if mailed in the United States

Postage will be paid by

TEKTRONIX, INC.
P.O. Box 500
Beaverton, Oregon 97005

ATTN: SPS Documentation Group 157 – 94-384

CHANGE INFORMATION