Tektronix®
COMMITTED TO EXCELLENCE

4050 SERIES
R07
SIGNAL PROCESSING
ROM PACK NO. 1
INSTRUCTION MANUAL

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

MANUAL PART NO.
070-2557-00

First Printing OCT 1978
This Printing DEC 1979

WARRANTY

This product is warranted against defective materials and workmanship, under normal use, for a period of ninety (90) days from date of initial shipment. CRTs found to be defective after the ninety (90) day period and up to twelve (12) months from date of shipment will be exchanged at no charge for the material. Tektronix, Inc. will repair or replace, at its option, those products which Tektronix, Inc. determines to be defective within the warranty period and are returned freight prepaid, to a designated Tektronix Service Center.

Tektronix, Inc. shall be under no obligation to furnish warranty service if:

a. Attempts to repair or service the equipment are made by other than Tektronix personnel.
b. Modifications are made to the hardware or software by other than Tektronix personnel.
c. Damage results from connecting the product to incompatible equipment.

The foregoing warranty shall be enforceable only by the original buyer and not by buyer's customer (end user).

This warranty is in lieu of all other warranties expressed or implied. There is no implied warranty for fitness of purpose. Tektronix, Inc. is not liable for consequential damages.

This instrument, in whole or in part, may be protected by one or more U.S. or foreign patents or patent applications. Information provided on request by Tektronix, Inc., P.O. Box 500, Beaverton, Oregon 97077.

PRODUCT 4051, 4052R07 Signal Processing ROM Pack No. 1

This manual supports the following versions of this product: Level 1 and up

MANUAL REVISION STATUS

REV.	DATE	DESCRIPTION
@	10/78	Original Issue
A	5/79	Revised pages
B	12/79	Revised pages

CONTENTS

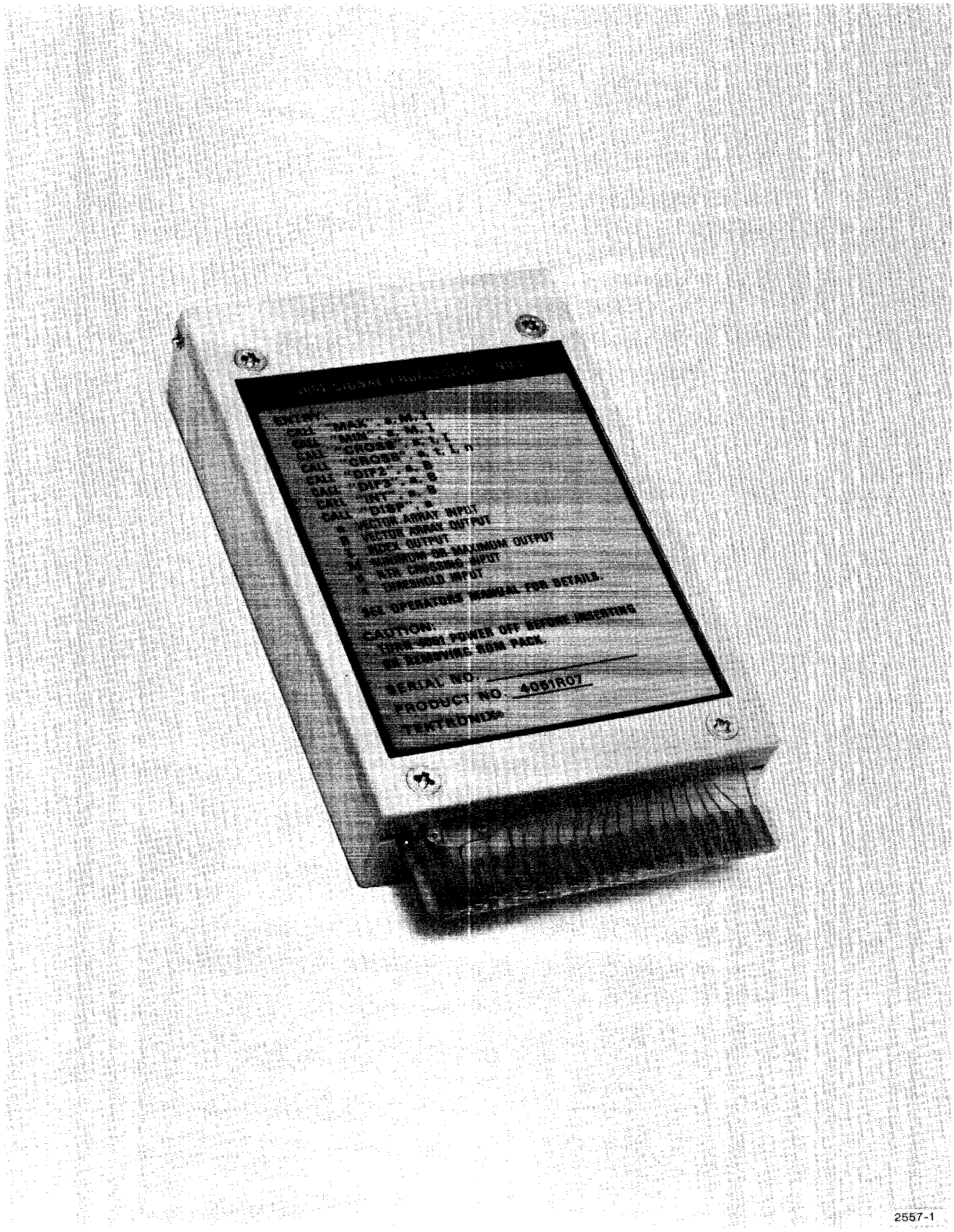


Figure 1-1. Signal Processing ROM Pack No. 1.

Section 1

GENERAL DESCRIPTION

INTRODUCTION

The TEKTRONIX 4050 Series R07 Signal Processing ROM Pack No. 1 (Figure 1-1) is a Read-Only Memory (ROM) device designed for use with TEKTRONIX 4050 Series Graphic Systems. The Signal Processing ROM Pack enables the Graphic System to perform signal processing functions on waveforms or other data stored in single-dimensional arrays. This added capability does not alter the operation of the Graphic System or occupy any of the available RAM (Random Access Memory) space.

The Signal Processing ROM Pack provides seven commands which are essential to signal processing:

- MAX—finds the maximum of an array.
- MIN—finds the minimum of an array.
- CROSS—finds the location of a specified crossing level within an array.
- DIF2—performs a two-point differentiation of an array.
- DIF3—performs a three-point differentiation of an array.
- INT—integrates an array.
- DISP—displays a graph of an array in raw form (without axes).

Commands provided by the Signal Processing ROM Pack execute much faster than BASIC programs that perform the same function. For example, the INT command integrates an array approximately ten times faster than an equivalent BASIC program.

An additional benefit of the ROM pack is that the signal processing routines require no Graphic System memory space. The routines are permanently stored in memory in the ROM Pack, leaving the Graphic System RAM space free for BASIC programs and data.

SPECIFICATIONS

Electrical

Power Requirements

The Signal Processing ROM Pack draws all necessary power from the 4050 Series Graphic System power supplies. Table 1-1 shows the power drawn from the System.

Table 1-1

POWER REQUIREMENTS

Power Supply	Maximum Current	
	Operating	Non-Operating[a]
+5 V	122 mA	28 mA
+12 V	0 mA	0 mA
−12 V	0 mA	0 mA

[a]ROM Pack inserted but no routines being executed.

Environmental

The Signal Processing ROM Pack No. 1 meets the environmental specifications of the Graphic System.

Temperature

Non-operating: −40 to +65 degrees C (−40 to +149 degrees F)
Operating: +10 to +40 degrees C (+50 to +104 degrees F)

Altitude

Non-operating: 15,240 m (50,000 feet) maximum
Operating: 4,572 m (15,000 feet) maximum

Humidity

Storage: 95% non-condensing
Operating: 80% non-condensing

Physical

Dimensions (including edge-board connector)

Length: 11.84 centimeters (4.7 inches)
Width: 6.65 centimeters (2.620 inches)
Depth: 2.22 centimeters (0.875 inches)

Weight

227 grams (8 ounces)

Standard Accessories

4050 Series R07 Signal Processing ROM Pack No. 1 Instruction Manual.

INSTALLING THE ROM PACK

CAUTION

4051R07 ROM Packs having serial numbers less than B02XXXX may not be installed in the 4051E01 ROM Expander.

1. Be sure the power switch is OFF.

CAUTION

Inserting the ROM Pack while the power is ON may cause damage to the ROM Pack. The Graphic System memory contents may also be lost.

2. Insert the Signal Processing ROM Pack into one of the slots in the rear of the Graphic System as shown in Figure 1-2. Press down and gently rock the plastic case until the ROM pack edge connector is firmly seated in the backpack.

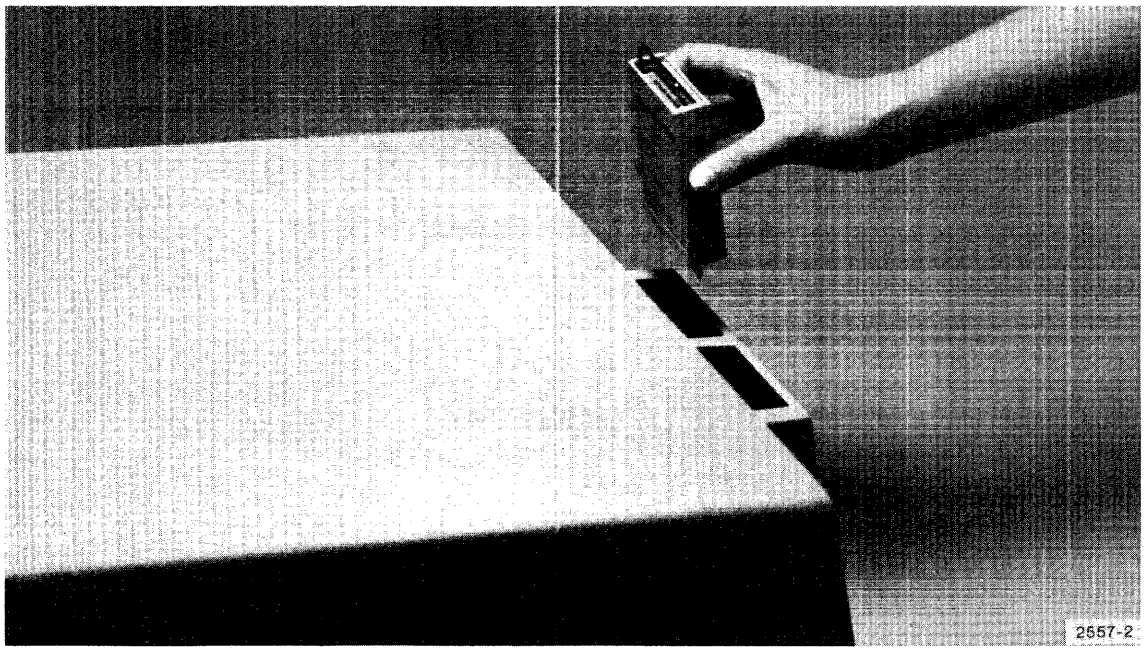


Figure 1-2. Installing the ROM Pack in the Graphic System.

3. Turn the power switch ON and wait a few seconds for the System to warm up.

Section 2

COMMAND DESCRIPTIONS

INTRODUCTION

This section contains a detailed description of each of the seven commands provided by the Signal Processing ROM Pack No. 1. The explanations are intended for users who are familiar with the operation of TEKTRONIX 4050 Series Graphic Systems.

The explanations include basic information about the use of the commands and, where necessary, some background information about the algorithms used. Examples and sample programs are included to help you use the commands, and a brief list of pitfalls to watch for is included at the end of each command description. A Command Summary is also provided in Appendix A for quick reference.

The ROM Pack routines are accessed using the CALL statement in 4050 Series BASIC. The routine name is specified in the CALL statement. Alternately, the routine name can be assigned to a string variable and represented in the CALL statement by that string variable. Refer to the 4050 Series Graphic System Reference Manual for more information on CALL statements.

THE MAX COMMAND

Syntax Form:

[line number] CALL { "MAX" / STRING VARIABLE } ,ARRAY ,NUMERIC VARIABLE ,NUMERIC VARIABLE

Descriptive Form:

[line number] CALL Routine Name, Source Array, Maximum Value, Subscript of Maximum Value

Purpose

The MAX command searches an array for its maximum value.

Examples

```
CALL "MAX",A,V,I
CALL A$,A,V,I
```

Explanation

The MAX command searches the specified array for its maximum value. The maximum value and the location (subscript) within the array of the maximum value are returned in the numeric variables specified in the command. If there is more than one occurrence of the maximum value, the first occurrence is returned.

Sample Program

The following program illustrates how the MAX command can be used to locate the maximum value of an array:

```
100 DELETE A
110 RESTORE
120 DIM A(10)
130 READ A
140 DATA 1,3,11,-3,2,2,7.2,5.4,4,-5.6,-1
150 CALL "MAX",A,V,I
160 PRINT "MAXIMUM = ";V;" AT LOCATION ";I
```

This program searches the source array (A) for its maximum value. Line 140 locates the maximum value and assigns it to the variable V (called the "target" variable). The location of this maximum value is assigned to the variable I. Lines 150 and 160 print the results on the display as shown:

```
MAXIMUM = 11 AT LOCATION 3
```

Things to Watch Out For When Using the Max Command

Undefined Variable	Not dimensioning the source array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the MAX command is executed.
Invalid Command Argument	Defining the source array as a simple numeric or string variable or defining the array to be two-dimensional causes an INVALID COMMAND ARGUMENT error when the MAX command is executed. Likewise, defining a target variable as an array or string variable results in an INVALID COMMAND ARGUMENT error when the MAX command is executed.

THE MIN COMMAND

Syntax Form:

[line number] CALL { "MIN" / STRING VARIABLE } ,ARRAY ,NUMERIC VARIABLE ,NUMERIC VARIABLE

Descriptive Form:

[line number] CALL Routine Name, Source Array, Minimum Value, Subscript of Minimum Value

Purpose

The MIN command searches an array for its minimum value.

Examples

```
CALL "MIN",A,V,I
CALL A$,A,V,I
```

Explanation

The MIN command is similar to the MAX command except that it searches the array for its minimum value. The minimum value and the location (subscript) of that value are returned and assigned to the numeric variables specified in the command. As in the MAX command, if there is more than one occurrence of the minimum value, the first occurrence is returned.

Sample Program

The sample program shown in the explanation of MAX command is easily modified to illustrate the use of the MIN command:

```
100 DELETE A
110 RESTORE
120 DIM A(10)
130 READ A
140 DATA 1,3,11,-3,2,2,7.2,5.4,4,-5.6,-1
150 CALL "MIN",A,V,I
160 PRINT "MINIMUM = "|V|" AT LOCATION "|I
```

This program searches the source array (A) for its minimum value. The results are printed on the screen as shown:

```
MINIMUM = -5.6 AT LOCATION 10
```

Things to Watch Out For When Using the MIN Command

Undefined Variable	Not dimensioning the source array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the MIN command is executed.
Invalid Command Argument	Defining the source array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error when the MIN command is executed. Likewise, defining a target variable as an array or string variable results in an INVALID COMMAND ARGUMENT error when the MIN command is executed.

THE CROSS COMMAND

Syntax Form:

[line number] CALL { "CROSS" / STRING VARIABLE } ,ARRAY,NUMERIC VARIABLE,NUMERIC VARIABLE [,NUMERIC VARIABLE]

Descriptive Form:

[line number] CALL Routine Name, Source Array, Threshold Value, Location of Cross, [Number of Cross]

Purpose

The CROSS command seraches an array for the location of the first element that crosses a specified value (threshold). If the array values cross this threshold more than once, any one of the locations can be selected.

Examples

```
CALL "CROSS",A,V,I,N
CALL "CROSS",A,V,I
CALL A$,A,V,I
```

Explanation

The CROSS command searches the source array for the first element that "crosses" a specified threshold value. If the array values increase as the subscripts increase, the cross location is the first element that is greater than or equal to the threshold. If the array values decrease as the subscripts increase, the cross location is the first element that is less than or equal to the threshold.

If the array values meet or exceed the threshold more than once, the optional fourth argument can be used to specify which crossover location is returned. The first crossover location is returned if the fourth argument is omitted. If the array values do not exceed the threshold the number of times specified by the fourth argument (or once, if the fourth argument is omitted), minus one (−1) is returned in the output variable.

Interpolated Array Locations

When the threshold value is crossed between array elements, the CROSS command interpolates between subscripts and returns a more precise value. The following series of program statements illustrates how the CROSS command returns an interpolated value:

```
100 DELETE A
110 RESTORE
120 DIM A(5)
130 READ A
140 DATA 1,3,5,7,9
150 CALL "CROSS",A,4,I
160 PRINT "CROSS LOCATION = ";I
```

The first four lines of the program generate an array (A) with the elements shown in line 130. Line 140 calls the CROSS routine which searches the array for the location where the threshold value (4) is met or exceeded. Notice that the threshold value is "crossed" between the second and third array elements. As a result, the routine interpolates between the subscripts and returns a value of 2.5. Line 150 prints the results as shown:

```
CROSS LOCATION = 2.5
```

When CROSS returns an interpolated array location, the value is not an integer. If this value is used as a subscript, the Graphic System rounds it to an integer. As a result, the variables B and V in the example below will not have the same values unless I is an integer.

```
CALL "CROSS",A,V,I
B=A(I)
```

The CROSS routine also detects "touchup" or "touchdown" crossover locations; i.e., locations where the array values attain, but do not cross the threshold level. Thus, the following sequence of program statements returns the same values in I1 and I2:

```
CALL "MAX",A,V,I1
CALL"CROSS",A,V,I2
```

Notice that the maximum value of the array A is stored in the variable V when MAX is done. I1 stores the location of this maximum value. The value in V is used as the threshold input variable for the CROSS command, so CROSS finds the same array location as MAX and the values in I1 and I2 are equal.

Things to Watch Out For When Using the CROSS Command

Undefined Variable	Not dimensioning the source array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the CROSS command is executed.
Invalid Command Argument	Defining the source array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error when the CROSS command is executed. Likewise, defining a target variable as an array or string variable results in an INVALID COMMAND ARGUMENT error when the CROSS command is executed.

THE DIF2 COMMAND

Syntax Form:

[line number] CALL { "DIF2" / STRING VARIABLE } ,ARRAY, ARRAY

Descriptive Form:

[line number] CALL Routine Name, Source Array, Destination Array

Purpose

The DIF2 command performs a two-point differentiation on the source array.

Examples

```
CALL "DIF2",A,B
CALL "DIF2",A,A
CALL A$,A,B
```

Explanation

DIF2 performs a two-point differentiation on the source array and returns the result in the destination array. The source and destination arrays must be of the same length and have at least three elements each, or an INVALID COMMAND ARGUMENT error is issued.

DIF2 performs a forward difference calculation on the source array with the results stored in the destination array. The destination array is calculated as follows:

$$B(t)=A(t+1)-A(t) \qquad \text{for } t=1,2,\ldots,N-1$$
$$B(N)=B(N-1)$$

Where:

A is the source array,
B is the destination array (result), and
N is the number of elements in the arrays.

If the same array variable name is used for both the source and destination arrays, the result replaces the source array.

Sample Program

The following program illustrates the use of the DIF2 command:

```
100 DIM A(25),B(25)
110 FOR I=1 TO 25
120 A(I)=SIN(I/5)
130 NEXT I
140 CALL "DIF2",A,B
```

This program sets up two arrays (A and B) of 25 elements each. Lines 110 through 130 generate a sine wave in array A. The DIF2 command in line 140 differentiates the sine wave and stores the result in array B.

Things to Watch Out For When Using the DIF2 Command

Undefined Variable	Not defining the input array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the DIF2 command is executed.
Invalid Command Argument	Defining the input or output array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error when the DIF2 command is executed. Likewise, dimensioning the input or output arrays for DIF2 to less than three elements or dimensioning the arrays to different sizes results in an INVALID COMMAND ARGUMENT error when the DIF2 command is executed.

THE DIF3 COMMAND

Sytnax Form:

[line number] CALL { "DIF3" / STRING VARIABLE } ,ARRAY, ARRAY

Descriptive Form:

[line number] CALL Routine Name, Source Array, Destination Array

Purpose

The DIF3 command performs a three-point differentiation on the source array.

Examples

```
CALL "DIF3",A,B
CALL A$,A,B
```

Explanation

The DIF3 command performs a three-point differentiation on the source array and returns the result in the destination array. As in DIF2, the source and destination arrays must be of the same length and have at least three elements each. If the same name is used for both the source and destination array, the differentiated result replaces the source array.

The three-point differentiation is calculated as follows:

$$B(1)=(-3*A(1)+4*A(2)-A(3))/2$$
$$B(t)=(A(t+1)-A(t-1))/2 \qquad \text{for } t=2,3,...,N-1$$
$$B(N)=(A(N-2)-4*A(N-1)+3*A(N))/2$$

Where:

A is the source array,
B is the destination array (result), and
N is the number of elements in the arrays.

Two-Point vs. Three-Point Differentiation

For arrays where large transitions occur over intervals greater than three elements, the three-point derivative exhibits the least analytic error in estimating the slopes at a given point. Thus, for smoothly varying functions (e.g., a sine wave), the three-point differentiation is the most accurate means of differentiating the function.

When array values exhibit large transitions within intervals approaching a step size of one (an interval of three adjacent array elements corresponds to a step size of one), the two-point derivative may provide a better slope estimate within the interval of transition. Square waves, steps, impulses, and other functions containing large transitions over limited intervals are best differentiated with the two-point algorithm.

Things to Watch Out For When Using the DIF3 Command

Undefined Variable	Not defining the input array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the DIF3 command is executed.
Invalid Command Argument	Defining the input array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error. Likewise, dimensioning the input or output arrays for DIF3 to less than three elements or dimensioning the arrays to different sizes results in an INVALID COMMAND ARGUMENT error.

THE INT COMMAND

Syntax Form:

[line number] CALL { "INT" / STRING VARIABLE } ,ARRAY, ARRAY

Descriptive Form:

[line number] CALL Routine Name, Source Array, Destination Array

Purpose

The INT command performs integration on the source array.

Example

```
CALL "INT",A,B
CALL A$,A,B
```

Explanation

The INT command integrates the source array and leaves the result in the destination array. The source and destination arrays must be of equal length and have at least three elements each. If the source and destination array names are the same, the integrated result replaces the source array. The integration is calculated using the trapezoidal rule for approximating the definite integral as follows:

$$B(1)=0$$
$$B(t)=B(t-1)+.5*(A(t-1)+A(t)) \qquad \text{for } t=2,3,...,N$$

Where:

A is the source array,
B is the destination array (result), and
N is the number of elements in the arrays.

Things to Watch Out For When Using the INT Command

Undefined Variable	Not defining the input array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the INT command is executed.
Invalid Command Argument	Defining the input or output array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error when the INT command is executed. Likewise, dimensioning the input or output arrays for INT to less than three elements or dimensioning the arrays to different sizes results in an INVALID COMMAND ARGUMENT error when the INT command is executed.

THE DISP COMMAND

Syntax Form:

[line number] CALL { "DISP" / STRING VARIABLE } ,ARRAY

Descriptive Form:

[line number] CALL Routine Name, Source Array

Purpose

The DISP command graphs the data in the source array.

Example

```
CALL "DISP",A
```

Explanation

DISP graphs the data in the source array on the Graphic System screen using the current window and viewport parameters. This command is intended for quick visual inspection of data. No labels or axes are plotted by the DISP command. Labels and axes can be generated by separate Graphic System commands.

The DISP command graphs the element values against the element subscripts. DISP terminates when all array elements have been plotted or when the subscript of the next element to be plotted exceeds the horizontal maximum of the current window. The routine clips data so that only those values which fall within the current window are plotted on the screen. An INVALID COMMAND ARGUMENT error message is issued if the horizontal window maximum is less than or equal to zero. If the current window range does not include any input array values, the routine terminates normally without graphing any data.

Setting the Window and Viewport

If the viewport parameters exceed the screen size (i.e., if viewport X-minimum is less than zero, X-maximum is greater than 130, Y-minimum is less than zero, and/or Y-maximum is greater than 100), the result is unpredictable and may be misleading. DISP is intended to give the user a quick preview of the data. The user is responsible for setting the window and viewport parameters before DISP is called.

The following example shows a suggested method for pre-setting the window size. These statements set the window to the exact size needed to display the waveform without leaving unused space or clipping elements.

```
CALL "MAX",A,V2,I
CALL "MIN",A,V1,I
WINDOW 1,M,V1,V2
```

In the window statement, M is the size of the waveform array.

For more information on setting the window and viewport, refer to the explanations of the VIEWPORT and WINDOW commands in the 4050 Graphic System Reference Manual.

Things to Watch Out For When Using the DISP Command

Undefined Variables	Not defining the input array or not assigning numeric values to each array element causes an UNDEFINED VARIABLE error to occur when the DISP command is executed.
Invalid Command Argument	Defining the input array as a simple numeric or string variable or defining the array as two-dimensional causes an INVALID COMMAND ARGUMENT error when the DISP command is executed.
	Defining the maximum X-parameter for the window as a zero or negative causes an INVALID COMMAND ARGUMENT error when DISP is executed.

Section 3
REPLACEABLE PARTS

PARTS ORDERING INFORMATION

Replacement parts are available from or through your local Tektronix, Inc. Field Office or representative.

Changes to Tektronix instruments are sometimes made to accommodate improved components as they become available, and to give you the benefit of the latest circuit improvements developed in our engineering department. It is therefore important, when ordering parts, to include the following information in your order: Part number, instrument type or number, serial number, and modification number if applicable.

If a part you have ordered has been replaced with a new or improved part, your local Tektronix, Inc. Field Office or representative will contact you concerning any change in part number.

Change information, if any, is located at the rear of this manual.

SPECIAL NOTES AND SYMBOLS

X000	Part first added at this serial number
00X	Part removed after this serial number

FIGURE AND INDEX NUMBERS

Items in this section are referenced by figure and index numbers to the illustrations.

INDENTATION SYSTEM

This mechanical parts list is indented to indicate item relationships. Following is an example of the indentation system used in the description column.

1 2 3 4 5 *Name & Description*

Assembly and/or Component
Attaching parts for Assembly and/or Component
- - - * - - -
Detail Part of Assembly and/or Component
Attaching parts for Detail Part
- - - * - - -
Parts of Detail Part
Attaching parts for Parts of Detail Part
- - - * - - -

Attaching Parts always appear in the same indentation as the item it mounts, while the detail parts are indented to the right. Indented items are part of, and included with, the next higher indentation. The separation symbol - - - * - - - indicates the end of attaching parts.

Attaching parts must be purchased separately, unless otherwise specified.

ITEM NAME

In the Parts List, an Item Name is separated from the description by a colon (:). Because of space limitations, an Item Name may sometimes appear as incomplete. For further Item Name identification, the U.S. Federal Cataloging Handbook H6-1 can be utilized where possible.

ABBREVIATIONS

"	INCH
#	NUMBER SIZE
ACTR	ACTUATOR
ADPTR	ADAPTER
ALIGN	ALIGNMENT
AL	ALUMINUM
ASSEM	ASSEMBLED
ASSY	ASSEMBLY
ATTEN	ATTENUATOR
AWG	AMERICAN WIRE GAGE
BD	BOARD
BRKT	BRACKET
BRS	BRASS
BRZ	BRONZE
BSHG	BUSHING
CAB	CABINET
CAP	CAPACITOR
CER	CERAMIC
CHAS	CHASSIS
CKT	CIRCUIT
COMP	COMPOSITION
CONN	CONNECTOR
COV	COVER
CPLG	COUPLING
CRT	CATHODE RAY TUBE
DEG	DEGREE
DWR	DRAWER
ELCTRN	ELECTRON
ELEC	ELECTRICAL
ELCTLT	ELECTROLYTIC
ELEM	ELEMENT
EPL	ELECTRICAL PARTS LIST
EQPT	EQUIPMENT
EXT	EXTERNAL
FIL	FILLISTER HEAD
FLEX	FLEXIBLE
FLH	FLAT HEAD
FLTR	FILTER
FR	FRAME or FRONT
FSTNR	FASTENER
FT	FOOT
FXD	FIXED
GSKT	GASKET
HDL	HANDLE
HEX	HEXAGON
HEX HD	HEXAGONAL HEAD
HEX SOC	HEXAGONAL SOCKET
HLCPS	HELICAL COMPRESSION
HLEXT	HELICAL EXTENSION
HV	HIGH VOLTAGE
IC	INTEGRATED CIRCUIT
ID	INSIDE DIAMETER
IDENT	IDENTIFICATION
IMPLR	IMPELLER
IN	INCH
INCAND	INCANDESCENT
INSUL	INSULATOR
INTL	INTERNAL
LPHLDR	LAMPHOLDER
MACH	MACHINE
MECH	MECHANICAL
MTG	MOUNTING
NIP	NIPPLE
NON WIRE	NOT WIRE WOUND
OBD	ORDER BY DESCRIPTION
OD	OUTSIDE DIAMETER
OVH	OVAL HEAD
PH BRZ	PHOSPHOR BRONZE
PL	PLAIN or PLATE
PLSTC	PLASTIC
PN	PART NUMBER
PNH	PAN HEAD
PWR	POWER
RCPT	RECEPTACLE
RES	RESISTOR
RGD	RIGID
RLF	RELIEF
RTNR	RETAINER
SCH	SOCKET HEAD
SCOPE	OSCILLOSCOPE
SCR	SCREW
SE	SINGLE END
SECT	SECTION
SEMICOND	SEMICONDUCTOR
SHLD	SHIELD
SHLDR	SHOULDERED
SKT	SOCKET
SL	SLIDE
SLFLKG	SELF-LOCKING
SLVG	SLEEVING
SPR	SPRING
SQ	SQUARE
SST	STAINLESS STEEL
STL	STEEL
SW	SWITCH
T	TUBE
TERM	TERMINAL
THD	THREAD
THK	THICK
TNSN	TENSION
TPG	TAPPING
TRH	TRUSS HEAD
V	VOLTAGE
VAR	VARIABLE
W/	WITH
WSHR	WASHER
XFMR	TRANSFORMER
XSTR	TRANSISTOR

Fig. & Index No.	Tektronix Part No.	Serial/Model No. Eff	Dscont	Qty	1 2 3 4 5	Name & Description	Mfr Code	Mfr Part Number
	4051R07			1		SIG PROCESSING:ROM PACK #1		
1-1	380-0384-01			1		. HSG,HALF,RDOUT:LID	80009	380-0384-01
						(ATTACHING PARTS)		
-2	211-0102-00			4		. SCREW,MACHINE:4-40 X 0.500",FLH,STL	83385	OBD
						- - - * - - -		
-3	367-0189-00			1		. HANDLE,BOW:	80009	367-0189-00
-4	334-3482-00			1		. MKR SET,IDENT:	80009	334-3482-00
-5	380-0343-01			1		. HSG,HALF,PTR:PLASTIC	80009	380-0343-01
-6	670-6188-00			1		. CKT BOARD ASSY:ROM PACK	80009	670-6188-00
-7	160-0160-00			1		. . MICROCIRCUIT,DI:2048 X 8 E PROM PRGM(U1)	80009	160-0160-00
-8	160-0161-00			1		. . MICROCIRCUIT,DI:2048 X 8 E PROM PRGM(U2)	80009	160-0161-00
-9	136-0578-00			2		. . SOCKET,PLUG-IN:24 DIP,LOW PROFILE	01295	C952402
-10	283-0010-00			1		. . CAP.,FXD,CER DI:0.05UF,+100-20%,50V(C3)	56289	273C20
-11	156-0469-00			1		. . MICROCIRCUIT,DI:3-LINE TO 8-LINE DCDR(U5)	01295	SN74LS138N
-12	315-0391-00			1		. . RES.,FXD,CMPSN:390 OHM,5%,0.25W(R2)	01121	CB3915
-13	315-0102-00			1		. . RES.,FXD,CMPSN:1K OHM,5%,0.25W(R1)	01121	CB1025
-14	283-0111-00			1		. . CAP.,FXD,CER DI:0.1UF,20%,50V(C2)	72982	8121-N088Z5U104M
-15	290-0106-00			1		. . CAP.,FXD,ELCTLT:10UF,+75-10%,15V(C1)	56289	30D106G015BA9
-16	151-0324-00			1		. . TRANSISTOR:SILICON,PNP(Q1)	80009	151-0324-00
						STANDARD ACCESSORIES		
	070-2557-00			1		MANUAL,TECH:INSTRUCTION	80009	070-2557-00

CROSS INDEX—MFR. CODE NUMBER TO MANUFACTURER

Mfr. Code	Manufacturer	Address	City, State, Zip
01121	ALLEN-BRADLEY COMPANY	1201 2ND STREET SOUTH	MILWAUKEE, WI 53204
01295	TEXAS INSTRUMENTS, INC., SEMICONDUCTOR GROUP	P O BOX 5012, 13500 N CENTRAL EXPRESSWAY	DALLAS, TX 75222
56289	SPRAGUE ELECTRIC CO.		NORTH ADAMS, MA 01247
72982	ERIE TECHNOLOGICAL PRODUCTS, INC.	644 W. 12TH ST.	ERIE, PA 16512
80009	TEKTRONIX, INC.	P O BOX 500	BEAVERTON, OR 97077
83385	CENTRAL SCREW CO.	2530 CRESCENT DR.	BROADVIEW, IL 60153

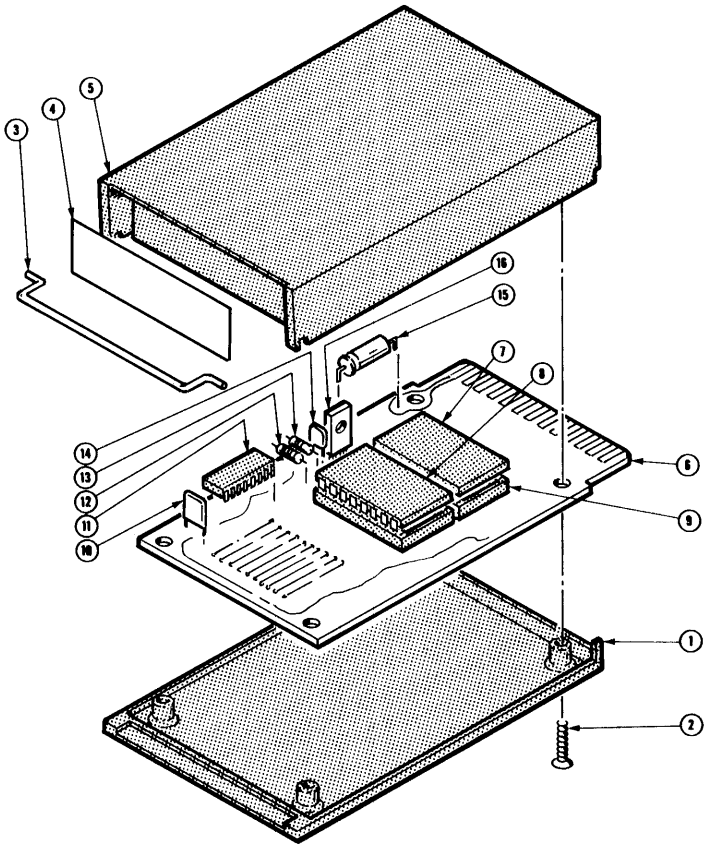


FIG. 1 EXPLODED

Section 4

DIAGRAMS

Symbols and Reference Designators

Electrical components shown on the diagrams are in the following units unless noted otherwise:

Capacitors = Values one or greater are in picofarads (pF). Values less than one are in microfarads (μF).

Resistors = Ohms (Ω).

Graphic symbols and class designation letters are based on ANSI Standard Y32.2-1975.

Logic symbology is based on ANSI Y32.14-1973 in terms of positive logic. Logic symbols depict the logic function performed and may differ from the manufacturer's data.

Abbreviations are based on ANSI Y1.1-1972.

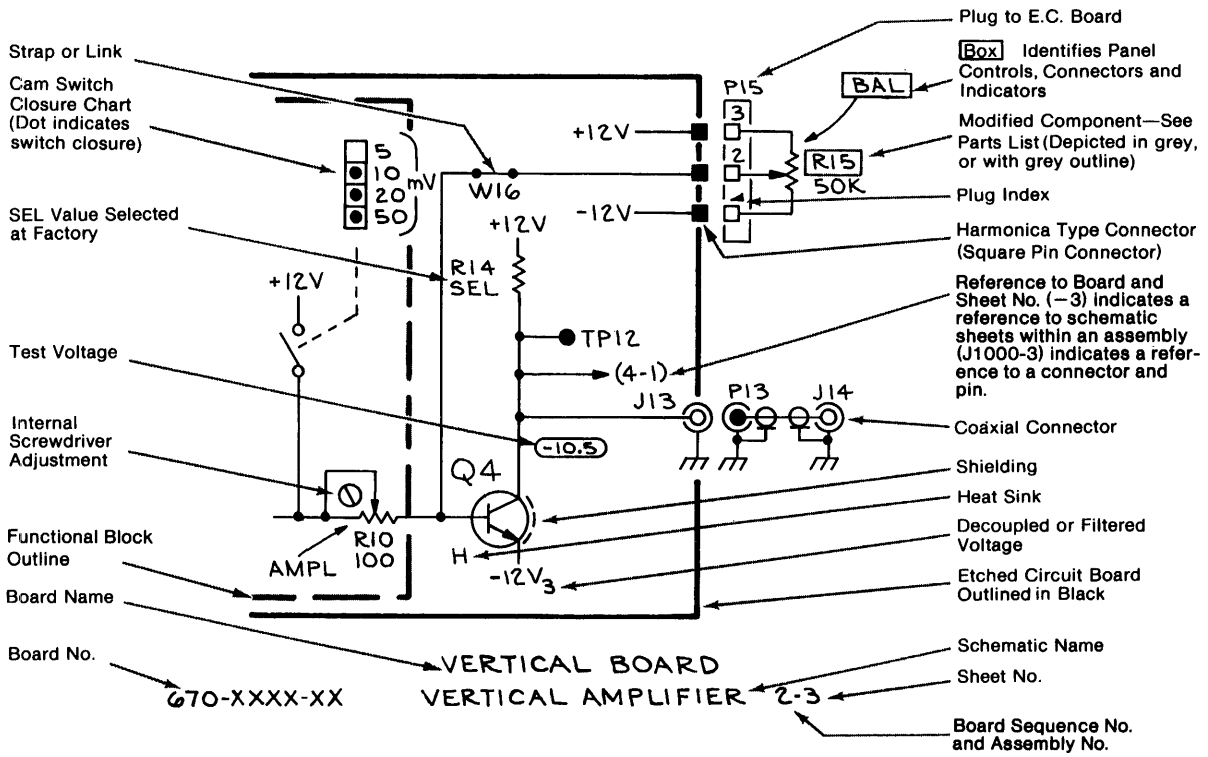
Other ANSI standards that are used in the preparation of diagrams by Tektronix, Inc. are:

Y14.15, 1966	Drafting Practices.
Y14.2, 1973	Line Conventions and Lettering.
Y10.5, 1968	Letter Symbols for Quantities Used in Electrical Science and Electrical Engineering.

The following prefix letters are used as reference designators to identify components or assemblies on the diagrams.

A	Assembly, separable or repairable (circuit board, etc)	H	Heat dissipating device (heat sink, heat radiator, etc)	S	Switch or contactor
AT	Attenuator, fixed or variable	HR	Heater	T	Transformer
B	Motor	HY	Hybrid circuit	TC	Thermocouple
BT	Battery	J	Connector, stationary portion	TP	Test point
C	Capacitor, fixed or variable	K	Relay	U	Assembly, inseparable or non-repairable (integrated circuit, etc.)
CB	Circuit breaker	L	Inductor, fixed or variable	V	Electron tube
CR	Diode, signal or rectifier	M	Meter	VR	Voltage regulator (zener diode, etc.)
DL	Delay line	P	Connector, movable portion	W	Wirestrap or cable
DS	Indicating device (lamp)	Q	Transistor or silicon-controlled rectifier	Y	Crystal
E	Spark Gap, Ferrite bead	R	Resistor, fixed or variable	Z	Phase shifter
F	Fuse	RT	Thermistor		
FL	Filter				

The following special symbols may appear on the diagrams:



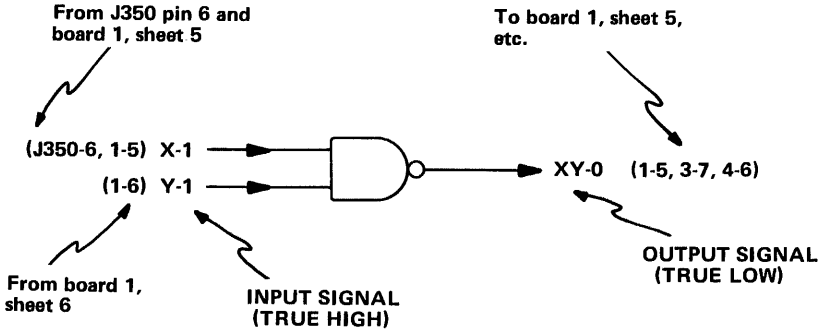
1. TRUE HIGH and TRUE LOW Signals

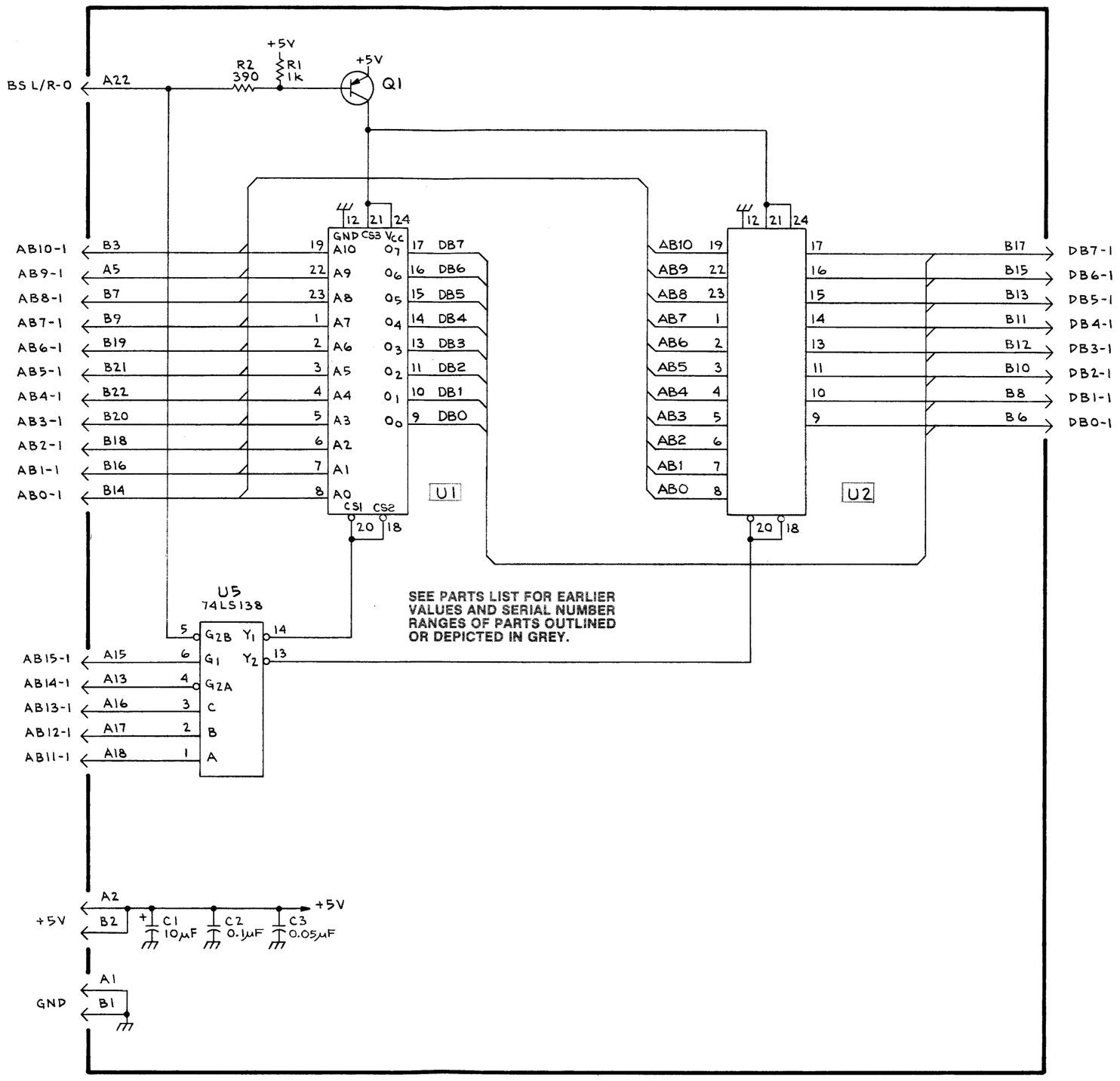
Signal names on the schematics are followed by −1 or −0. A TRUE HIGH signal is indicated by −1, and a TRUE LOW signal is indicated by −0.

SIGNAL−1 = TRUE HIGH
SIGNAL−0 = TRUE LOW

2. Cross-References

Schematic cross-references (from/to information) are included on the schematics. The "from" reference only indicates the signal "source," and the "to" reference lists all loads where the signal is used. All from/to information will be enclosed in parentheses.





4051R07

2557-01
REVA, DEC 1979

670-6488-00&01 SIGNAL PROCESSING ROM PACK NO. 1

Appendix A

COMMAND SUMMARY

Function	Example	Purpose
MAX	CALL "MAX",A,V,I	MAX searches an array for the maximum value and returns the value and its location (subscript) within the array.
MIN	CALL "MIN",A,V,I	MIN searches an array for the minimum value and returns the value and its location (subscript) within the aray.
CROSS	CALL "CROSS",A,V,I,N or CALL "CROSS",A,V,I	CROSS searches an array for the Nth time (or the first, if N is omitted) that the array values cross the specified threshold.
DIF2	CALL "DIF2",A,B	DIF2 performs a two-point differentiation on the source array.
DIF3	CALL "DIF3",A,B	DIF3 performs a three-point differentiation on the source array.
INT	CALL "INT",A,B	Integrates the source array.
DISP	CALL "DISP",A	Graphs the source array without axes.

Appendix B

UNDERSTANDING ERRORS

Error Number	Error Message
12	INVALID COMMAND ARGUMENT IN IMMEDIATE LINE—MESSAGE NUMBER 12 INVALID COMMAND AGRUMENT IN LINE xx—MESSAGE NUMBER 12
	Probable Cause
	a. An array variable is not properly defined. b. An input array is defined as two-dimensional. c. A simple numeric input or output variable is defined as an array or string variable. d. The input or output array for DIF2, DIF3 or INT is dimensioned to have fewer than three elements. e. The input or output arrays for DIF2, DIF3 or INT are not dimensioned to the same size. f. The maximum X-parameter for the window is negative or zero when DISP is called.
32	CALL NAME INVALID IN IMMEDIATE LINE—MESSAGE NUMBER 32 CALL NAME INVALID IN LINE xx—MESSAGE NUMBER 32
	Probable Cause
	a. The correct ROM pack is not installed or incorrectly installed. b. The ROM pack is defective.
36	UNDEFINED VARIABLE IN IMMEDIATE LINE—MESSAGE NUMBER 36 UNDEFINED VARIABLE IN LINE xx—MESSAGE NUMBER 36
	Probable Cause
	An input array is not defined or contains undefined elements.

SIZE ERRORS

In most cases, the routines in the Signal Processing ROM Pack No. 1 do not generate size errors unless the input values exceed the limitations in the Graphic System itself. Any value between −1.0E+308 and +1.0E+308 is accepted by the Graphic System. With FUZZ at its default value, values between −1.0E−64 and +1.0E−64 are treated as equal to zero for comparison purposes. (For more information on the FUZZ parameter, refer to the 4050 Series Graphic System Reference Manual.)

Occasionally size errors may occur during differentiation and integration because the internal algorithms sum and/or multiply input values. A size error can occur if the array values approach the range limitations of the Graphic System or if the absolute value of the difference between adjacent array values approaches size limitations (−1.0E+308 to +1.0E+308). Refer to the 4050 Series Graphic System Reference Manual for more information on size errors.

Appendix C

VERIFICATION PROGRAM

INTRODUCTION

The following sample program exercises the basic functions of the Signal Processing ROM Pack No. 1. This program, written in 4050 Series BASIC, also illustrates the use and syntax of the seven commands provided by the ROM Pack. If you encounter errors while using the ROM Pack, this program can be run to verify that the ROM Pack is functioning properly.

RUNNING THE SAMPLE PROGRAM

To run the program, install the ROM Pack as discussed in Section 1 of this manual. Then enter the program as shown. Refer to the 4050 Series Graphic System Reference Manual if you need more information.

```
100 DIM A(501),B(501)
110 VIEWPORT 0,80,25,80
120 WINDOW 1,501,-2,2
130 J=0
140 FOR I=1 TO 26 STEP 0.05
150 J=J+1
160 A(J)=SIN(I)
170 NEXT I
180 PAGE
190 CALL "MAX",A,V,J
200 PRINT "MAXIMUM VALUE =";V;" AT LOCATION ";J
210 CALL "MIN",A,V,J
220 PRINT "MINIMUM VALUE = ";V;" AT LOCATION ";J
230 CALL "CROSS",A,V,J
240 PRINT "ARRAY A CROSSES ";V;"AT LOCATION ";J
250 CALL "CROSS",A,0,J,2
260 PRINT "ARRAY A CROSSES ZERO SECOND TIME AT LOCATION ";J
270 CALL "DIF2",A,B
280 CALL "MAX",B,V,J1
290 PRINT "AFTER DIF2,MAXIMUM FOUND AT LOCATION ";J1
300 CALL "DIF3",A,B
310 CALL "MAX",B,V,J1
320 PRINT "AFTER DIF3, MAXIMUM FOUND AT LOCATION ";J1
330 CALL "INT",A,B
340 CALL "MAX",B,V,J1
350 PRINT "AFTER INT,MAXIMUM FOUND AT LOCATION ";J1
360 CALL "DISP",A
370 END
```

After the program is entered into the Graphic System, type RUN. If you encounter errors, refer to Appendix B. Find the error number in the appendix and check the probable causes.

As the program runs, the results of each command call are printed on the screen. Figure C-1 shows the output from a successful run of the sample program. Check that the values printed on the screen correspond to those shown in the figure.

```
MAXIMUM VALUE =0.999992073306 AT LOCATION 138
MINIMUM VALUE = -0.999990206551 AT LOCATION 201
ARRAY A CROSSES -0.999990206551AT LOCATION 106.663675692
ARRAY A CROSSES ZERO AT LOCATION 106.663675692
AFTER DIF2,MAXIMUM FOUND AT LOCATION 483
AFTER DIF3, MAXIMUM FOUND AT LOCATION 358
AFTER INT,MAXIMUM FOUND AT LOCATION 295
```

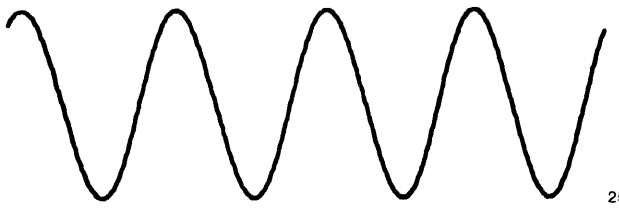


Figure C-1. Output From the Sample Program.

UNDERSTANDING THE PROGRAM

The sample program generates a sine wave in an array and uses this array as input to exercise the ROM pack routines. Line 100 dimensions two arrays—A and B. Lines 110 and 120 set the window and viewport for the graph generated at the end of the program. The next five lines (130—170) load array A with a sine wave. Then the MAX routine is called in line 190 to find the maximum value of the sine-wave array. The maximum value is returned in the variable V and the location (subscript) of this value is returned in J. Line 200 prints the results of the MAX call.

The MIN command is called in line 210. Again the results of the MIN call are printed on the Graphic System screen. Line 230 calls the CROSS routine. The variable V is left with the minimum value of the sine wave from line 210 and this value is used as the threshold input to the CROSS routine. As a result, the CROSS routine returns the location of the minimum value in J. Line 240 prints the results. Then the CROSS command is called again, this time searching for the second zero crossing of the sine wave. Notice that the location returned is not an integer and that the value is interpolated by the CROSS routine. Line 270 performs a two-point differentiation on the sine-wave in array A and stores the result in array B. The MAX command is called again in line 280 to check that the maximum value is in the correct location and line 290 prints the results.

Next, a three-point differentiation of the original array is performed and the results replace the array generated by the DIF2 call. Line 310 locates the maximum value of the result and line 320 prints the results. The INT routine is called in line 330 and a similar message is printed. Finally, the original sine wave is plotted using the DISP routine.