

**Tektronix
4050 Series**

**Peripherals and
ROM Packs**

Tekniques
Tekniques
Tekniques
Tekniques
Tekniques
Tekniques

Reprinted from Tekniques
The 4050 Series Applications Library
Newsletter

Contents

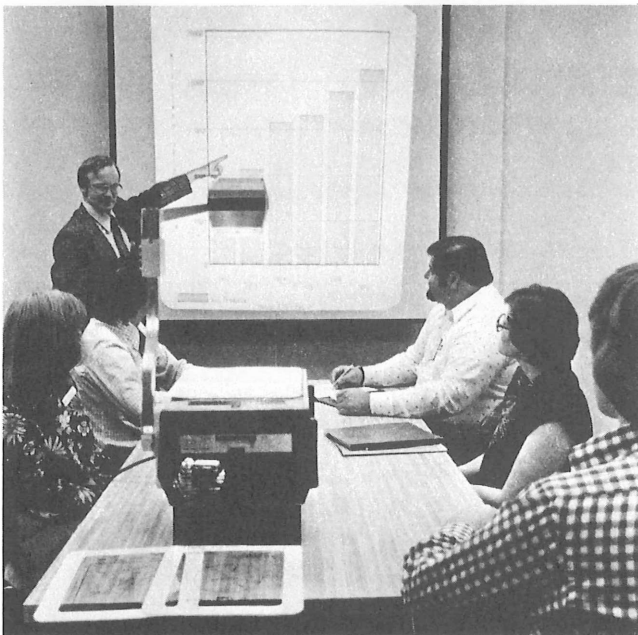
Making Slides with the 4051 and 4662	3
Debut of the 4907, an Intelligent Disc Unit for the 4051	4
The 4051R06 EDITOR: A Text Editor for the 4051	6
4956 Graphic Tablet Provides Fast and Accurate Graphic Input	8
New ROM Pack — Adds Fast Graphing and Data Analysis Functions to the 4051	10
The 4051R06 EDITOR: Searching Operations .	12
The 4051R06 EDITOR: Sorting Operations	15
Signal Processing ROM Pack #1 Speeds Up Numeric Sorting	16
New ROM Pack Enhances Data Analysis Capabilities of 4050 Series	18
Expanding Your Plotting Capabilities: Introducing the 4663	21

Tekniques

REPRINTED FROM TEKNIQUES, VOL 1 NO 8 — 4050 SERIES APPLICATIONS LIBRARY NEWSLETTER

Making Slides with the 4051 and 4662

By Byron Fisher

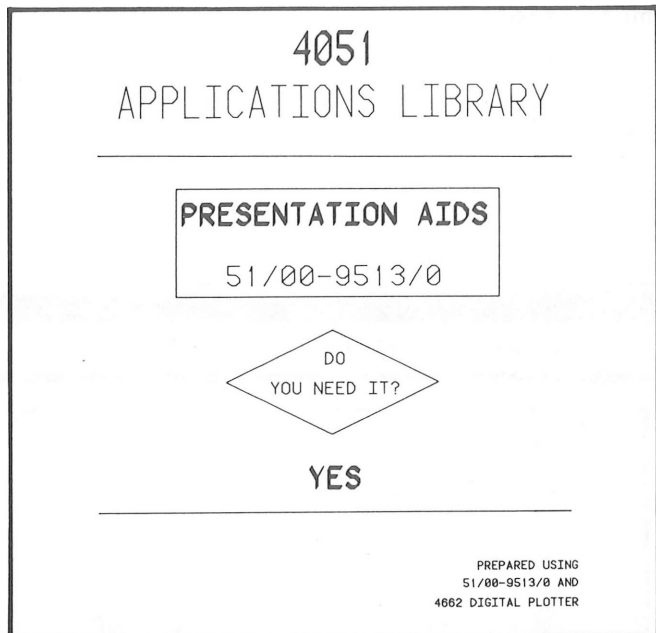


A bar chart is quick and exact with the aid of a 4662 Plotter and 4051.

From time to time, all of us have occasion to produce slides for classes or for presentations to colleagues and management. The 4662 Interactive Digital Plotter, used with a 4051 Graphic System, is a practical way to fulfill such requirements.

The Applications Library contains a program that is useful for making graphs for presentations. This program includes a slide maker and pie chart in Presentation Aids (51/00-9513/0). Using this BASIC program in the 4051 Graphic System with a 4662 Plotter attached, making slides is easy.

Slides can be prepared in two ways. To produce 35mm slides with the 4051/4662 combination, a program is run on the 4051 to generate the desired graphic presentation. The graphic output is directed to the Plotter. The resulting graphics, which can be in several colors, is then photographed to produce 35mm slides. The other method produces slide for an overhead projector. This method uses clear film on the plotting surface to produce the overhead transparency directly on the Plotter. The clear write-on film to use is 3M no. 78-6969-1733-1 or



Demonstrating the 4662 Plotter's versatility, this slide was produced using 4051 Applications Library program #51/00-9513/0.

equivalent. Tektronix offers nine colors of special pens that can be used to write on the clear film. These pens make presentations clear, interesting, and easy to understand. Information on the pens is as follows.

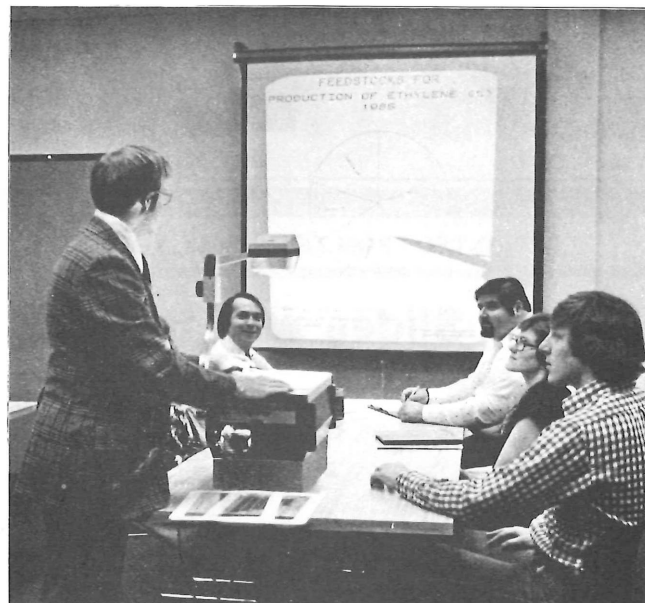
Black	016-0386-00	Package of Three
Brown	016-0386-01	Package of Three
Red	016-0386-02	Package of Three
Orange	016-0386-03	Package of Three
Yellow	016-0386-04	Package of Three
Green	016-0386-05	Package of Three
Blue	016-0386-06	Package of Three
Purple	016-0386-07	Package of Three
Magenta	016-0386-08	Package of Three

Use these pens with the Slide Maker Program to produce professional slides for your presentations.

Consider the mirror-image approach when preparing overhead slides. When using the mirror image, the presentation is drawn backwards on the back side of your clear film. Place the slide on the projector with the written surface down; the projected image will be righted again. You can then use a marking pen to highlight or note items on the top surface during your presentation. These notes can be erased after the presentation. The Plotter's

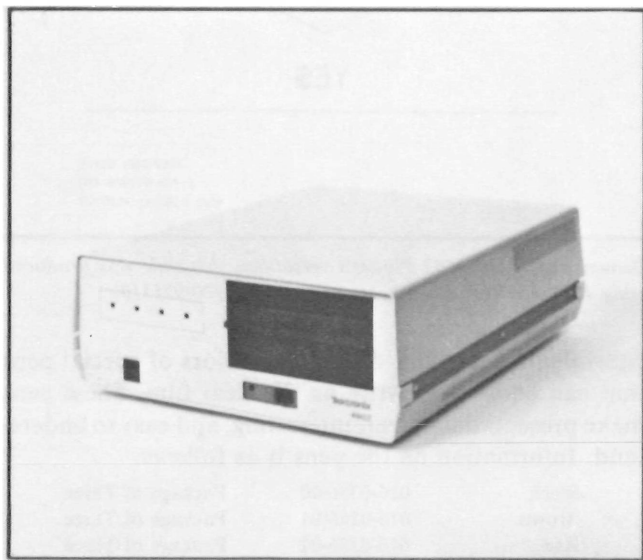
drawing on the other side of the film will remain intact for the next showing.

Creating a mirror image of the plots is easy by using the 4662's front panel pushbutton controls. The 4662 powers up with the page size and orientation set to default conditions (full page, bottom left = bottom left, upper right = upper right). For mirror image, move the pen to lower right position on the desired page area and hold down LOWER LEFT button until the bell rings. Then move the pen to the upper left position on the desired page area and hold down the UPPER RIGHT button until the bell rings. All incoming data will now be plotted as a mirror image.

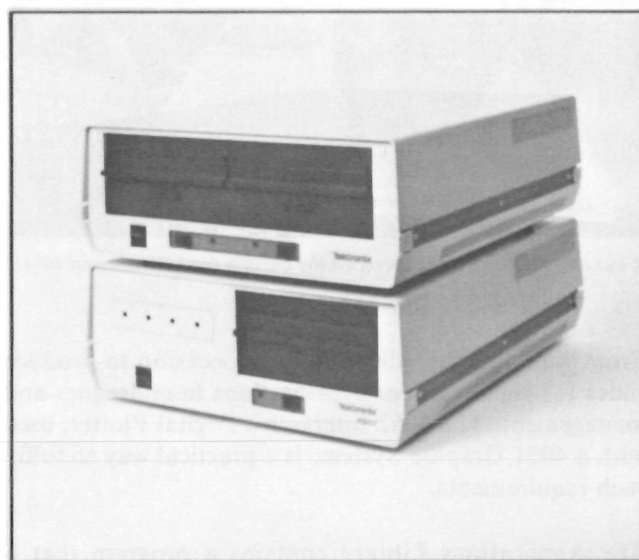


The pie chart and text were drawn directly onto the clear write-on film by the 4662 Plotter driven by the 4051.

REPRINTED FROM TEKNIKES, VOL 1 NO 9 — 4050 SERIES APPLICATION LIBRARY NEWSLETTER



The compact 4907 File Manager brings direct access flexible disc storage to the 4051.



The 4907, Option 31, triple disc drive enables up to 1.89 million bytes of storage.

Debut of the 4907, an Intelligent Disc Unit for the 4051

by Gary P. Laroff

The 4907 FILE MANAGER is a high capability flexible disc mass storage peripheral that makes a 4051-based computation system even more powerful than before. Dual and Triple disc drive systems are available as options. The 4907 is smart. It's built around a microprocessor to handle file management and directory chores. A new ROM Pack, the 4051 FILE MANAGER ROM Pack, is included with the 4907 as a standard accessory. It allows easy access of the 4907's file handling capabilities by the 4051 from a BASIC language program.

Briefly, the 4907 is a direct access, flexible disc device, utilizing a double density read/write feature that enables up to a 630,000 byte capacity per disc. That comes to 1.89 million bytes of storage for the triple drive configuration.

Programming the 4907 is convenient using the new ROM Pack. A major plus is that the file management and directories are handled within the 4907 unit. None of the 4051 read/write memory is required by the 4907 or ROM Pack. In fact, most programs that now do sophisticated

file manipulation with either the 4051 internal tape and/or the 4924 Digital Cartridge Tape Unit can generally be rewritten for the 4907 using a much shorter program. The 4907 can support nine files open simultaneously. Therefore, data transfer, sorts and merges can be accomplished with a minimum of programming. As with the tape drives, status can be monitored on each file by using commands such as "ON EOF (3) THEN", "TYPE (6)" and "CLOSE 4". File access is made easy and quick with an advanced multiple level file-by-name system (each name up to 10 characters plus a 3 character extension). This library structure includes a directory that maintains user files, dates and times of access, passwords and available space.

The 4907 FILE MANAGER offers many features available only on large computer systems in addition to those standard features commonly available on calculators and "mini" systems. A real time clock providing date and time of file activities, multiple file accesses, file security (passwording and execute only) and dynamic allocation of file space are examples of these enhancements.

Many file activities are performed by the 4907 FILE MANAGER which normally require applications programs to perform; file compression, dates and duplication are just a few of these.

The FILE MANAGER and ROM Pack contain a Motorola microprocessor system utilizing 36K of ROM and 8K of RAM. The 4907 instructions are passed from the 4051 via the GPIB in an internal format. These machine level commands are generated by the system firmware interpretation, e.g.: CLOSE, INIT, CREATE, etc., or by the FILE MANAGER ROM Pack subroutines e.g.: CALL "MOUNT", CALL "REWIND", CALL "FORMAT", etc. No system software bootstrapping or special software package is required. As with the internal tape drive, all of the commands may be used within a program and parameters can be generated by the program.

Program overlaying, chaining, and appending are performed in a fraction of the time required when using the tape storage. Directory and file management capabilities allow ease in storing and retrieving programs and data files. The default file structure for programs is binary format. A binary capability similar to the operation of the 4051R05 Binary Program Loader ROM Pack is included in the File Manager ROM Pack.

You'll be seeing a number of 4907 programs for sorting, handling of extensive graphic data bases and data base management here, in TEKniques, in issues to come.

The 4051R06 EDITOR: A Text Editor for the 4051

by Cathy Cramer

This is the first in a series of articles on the Tektronix 4051R06 EDITOR.

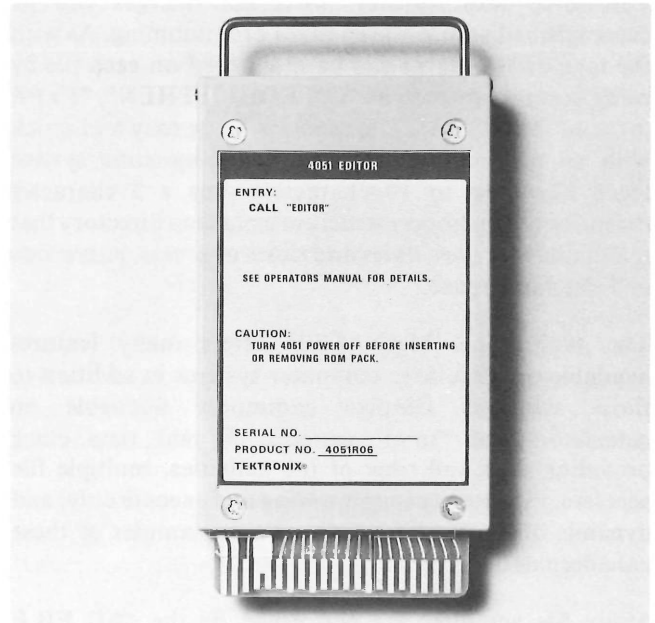
One of the most impressive accessories for the 4051 is the 4051R06 EDITOR. The 4051R06 EDITOR is a ROM Pack, a small Read-Only Memory device that fits into one of the 4051 backpack slots, or into a ROM Expander slot. Inside the ROM Pack is a complete line-oriented and string-oriented text editor that allows you to use the 4051 to create, edit, and store ASCII text of any kind.

The EDITOR'S text editing language consists of 29 commands, including special purpose editing commands, I/O commands, and "environmental" or system status commands. All of the commands become available when you execute the statement CALL "EDITOR" from the 4051 keyboard. And because it's a ROM Pack, the EDITOR takes up only 592 bytes of read/write memory. The remainder of read/write memory can be used to hold text for editing.

You can use the EDITOR to create new files, or to edit already existing files. The commands let you bring lines of text into memory from tape, modify them quickly and easily, and send the edited lines to an internal or external tape storage or printing device. The "text" you work with may be any set of ASCII characters—programs, data, or free text like that in a letter or textbook. If your text is a program, it need not be a BASIC program: you can use the EDITOR to write or edit programs in FORTRAN, COBOL, or any other programming language based on the ASCII character set.

This opens up a whole new world for the 4051. For the first time, you can use the 4051 to write programs in other programming languages without getting syntax errors. You can write and store your programs off-line using the EDITOR, then use the Option 1 Data Communications Interface to send them to a host computer for processing. Or, you can use the same interface to pull files from the host into the 4051 internal magnetic tape, then edit them off-line with the EDITOR. Writing, updating, and documenting programs or data off-line using the EDITOR can save you costly connect time, and spares you from the irritating response delays you can experience while using a text editor that resides on the host.

There are many uses for the EDITOR beside writing programs in other languages. The EDITOR provides fast



TEKTRONIX 4051R06 EDITOR ROM Pack allows you to use the 4051 to create, edit and store ASCII text of any kind.

methods for editing BASIC programs. For example, with one command you can modify an entire program so that it sends all output to a different peripheral device. With another command, you can obtain a listing of every line in the program that contains a particular variable name or keyword. This also means you can easily use the EDITOR to translate programs from other versions of BASIC into 4051 BASIC.

You can use the EDITOR's SORT command to alphabetize long lists of names or to put lists of numbers in increasing or decreasing order. The EDITOR can also play an important role in mass producing high-quality letters. You create and edit the letter at the 4051 keyboard, under EDITOR control, and save it on tape. Then you use a simple BASIC program to read the letter into memory as a string, add the name and address of the recipient of the letter, and send a final copy to a typewriter terminal.

Using the EDITOR

To use the EDITOR, all you have to do is type CALL "EDITOR" and press RETURN. The EDITOR immediately takes control of the Graphic System. Now you enter EDITOR commands, not BASIC commands. During a typical session, you might create and store several new files of text; or bring previously stored files into memory, edit them, and save the corrected versions on tape.



Fig. 1. The EDITOR displays FORTRAN text in two columns on the 4051 screen.

The EDITOR allows you to enter lines up to 396 characters long. (Without the EDITOR, the 4051 accepts lines up to 72 characters long.) The total amount of text you can enter is limited only by the amount of read/write memory available in your 4051 when you call the EDITOR. You can still use the LINE EDITOR keys on the keyboard, to correct one line at a time. And three of the user-definable keys allow you to control how the text appears on the display, and what happens when a PAGE FULL condition occurs. For example, pressing the key labeled MARGIN 2 on the overlay causes text entered from the keyboard or listed on the display to appear in two columns, as shown in Fig. 1. When the screen is full, a blinking F appears, prompting you to press HOME/PAGE.

When you are finished using the EDITOR, just press the RETURN TO BASIC overlay key. Control is immediately returned to BASIC. BASIC commands are available again, as well as any program or variables that were in memory when you called the EDITOR.

EDITOR Commands

The EDITOR is a natural extension to BASIC. If you know BASIC, you won't have any problems learning the EDITOR. In fact, you may find the EDITOR even quicker and easier to use than BASIC. Most EDITOR keywords can be abbreviated to one character: O for OLD and L for LIST, for example. And like BASIC, the EDITOR allows you to omit parameters and rely on default values. For example, you just enter L to list the entire text.

Some of the EDITOR commands are already familiar to you. APPEND, DELETE, FIND, INPUT, MARK,

OLD, PRINT, RENUMBER, SAVE, and WRITE are similar to their BASIC counterparts. In addition, the EDITOR offers many commands you haven't seen before: commands like COPY, INSERT, MOVE, SEARCH, SKIP, SORT, and others. These new commands are specially designed to allow you to swiftly rearrange and modify text.

For example, the COPY command allows you to copy one or more lines, and place the copy anywhere you want. The MOVE command allows you to move specified lines around in the text. The INSERT command lets you insert new lines anywhere in the text (and is the command you use to create new text).

Some examples are shown in Figs. 2 through 4. You'll notice in the examples that EDITOR commands allow you to specify line numbers. The line numbers are called "edit line numbers," and are assigned to the text by the EDITOR command RENUMBER. In the examples, you'll see the edit line numbers to the left of the lines, and separated from the text by a colon.

In Fig. 2, the COPY command is used to duplicate two lines of text. First, a complete listing is obtained by

```

L
1: REAL FUNCTION PSSCALE(M)
2: COMMON /BTEST/ A, B, C, D
3: INTEGER N, B, C, D
4: PSSCALE = FLOAT(N)/256.0
5: RETURN
6: END
7:C
8: REAL FUNCTION RANGLE(N)
9: RANGLE = FLOAT(N)/188.0/32767.0
10: RETURN
11: END
C 2,3,9
L
1: REAL FUNCTION PSSCALE(M)
2: COMMON /BTEST/ A, B, C, D
3: INTEGER N, B, C, D
4: PSSCALE = FLOAT(N)/256.0
5: RETURN
6: END
7:C
8: REAL FUNCTION RANGLE(N)
9: COMMON /BTEST/ A, B, C, D
10: INTEGER N, B, C, D
11: RANGLE = FLOAT(N)/188.0/32767.0
12: RETURN
13: END

```

Fig. 2.

entering L for LIST. Next the command C 2,3,8 is executed to duplicate lines 2 through 3, and place the copy immediately before line 8. A new listing obtained by entering L again shows the results.

In Fig. 3, the MOVE command is used to rearrange a portion of a FORTRAN program. An initial listing is obtained by entering L. Then the command M 6,10,1 is executed to move lines 6 through 10, and place them immediately before line 1. A new listing shows how the lines have been rearranged.

```

L
1:C  DEFINE FORTRAN FUNCTION F(X)
2:  FUNCTION F(X)
3:  F = X**2-5
4:  END
5:  RETURN
6:C  DEFINE FORTRAN FUNCTION DF(X)
7:  FUNCTION DF(X)
8:  DF = 4*X**2-6
9:  END
10: RETURN

M 6,10,1

L
1:C  DEFINE FORTRAN FUNCTION DF(X)
2:  FUNCTION DF(X)
3:  DF = 4*X**2-6
4:  END
5:  RETURN
6:C  DEFINE FORTRAN FUNCTION F(X)
7:  FUNCTION F(X)
8:  F = X**2-5
9:  END
10: RETURN

```

Fig. 3.

Fig. 4 shows the INSERT command being used to add a new line of text. After the initial listing, the command I 3

is given. This tells the EDITOR that new text is about to be entered, and should be placed immediately before line 3. The EDITOR returns the colon (:) as a prompt, and waits. The new line Apt. 5 is entered from the keyboard and the BREAK key is pressed. (To insert more than one line, you'd press RETURN here instead of BREAK.) The final listing shows the newly created line immediately before line 3.

```

L
1:Anne L. Brunner
2:11635 Nobel Ave.
3:Los Angeles, Calif. 97685

I 3
:Apt. 5

L
1:Anne L. Brunner
2:11635 Nobel Ave.
3:Apt. 5
4:Los Angeles, Calif. 97685

```

Fig. 4.

These were very simple examples. In upcoming months, you'll see more articles on the EDITOR, including using the EDITOR to perform searching and sorting operations.

¹ You'll also notice that lines moved by the MOVE command, or newly created by the COPY or INSERT command, do not have edit line numbers. You can assign a complete set of line numbers at any time, by executing a RENUMBER command.

REPRINTED FROM TECHNIQUES, VOL 2 NO 5 — 4050 SERIES APPLICATIONS LIBRARY NEWSLETTER

4956 Graphic Tablet Provides Fast and Accurate Graphic Input

by Miki Tokola

For fast, accurate and convenient graphic data entry into the 4051, consider the TEKTRONIX 4956 Graphic Tablet. It calculates the coordinates of any point you specify on its special work surface, and transfers the coordinates directly to the 4051 through the GPIB. Coordinates can be stored, displayed, or used in calculations. The 4051/4956 system brings exceptional speed and accuracy to any application requiring identification and processing of coordinate information on a two-dimensional surface.

You can use the Graphic Tablet to create a graphic display on the 4051 screen without calculating all of the points and using the necessary MOVE and DRAW commands.

You can generate data files from graphic media, such as strip charts, without tedious data measurement and time-consuming keyboard entry. You can then store these data files on the internal magnetic tape unit, peripheral disc, or other peripheral recording device.

Another way to use the Graphic Tablet is for task and

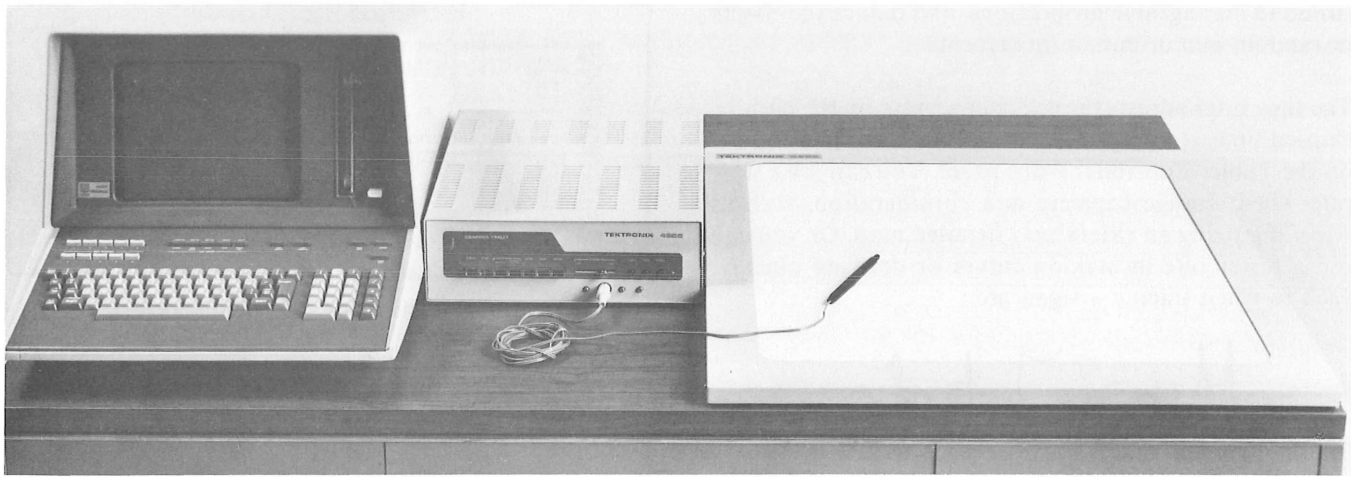
program control, by using the Tablet in a "menu selection" application. When the Tablet is used in this way, coordinate data transferred to the 4051 represents a choice from a "menu" of items. Or it may indicate the location for a desired action, such as the placement on a schematic of a component stored in memory and recalled through "menu selection."

System Parts Work Together

The 4956 consists of a standard 20" by 20" Tablet, a Tablet Controller, GPIB cable, a standard Tablet Writing Pen, and a PLOT 50 Graphic Tablet Support Tape. Options include a 36" by 48" Tablet and a Four-Button Cursor.

The Tablet serves as the working surface on which you place the graphic material to be digitized. This surface is constructed on an extremely flat and stable metal wood sandwich. Special wires which change dimensions depending on their state of magnetization (magnetostrictive wires) are laid under the writing surface in both the X and Y directions. They form a two-layer mesh measuring 4,000 units by 4,000 units and provide identifiable points that are only .005" apart.

A current pulse generates a strain wave which propagates



The TEKTRONIX 4956 Graphic Tablet coupled with the 4051 converts tedious, error-prone tasks into efficient and accurate ones.

down all the wires in one direction simultaneously. A receive coil in the Writing Pen provides an electrical signal from the flux change. The 4956 microprocessor converts time delay distance between the "send" and "receive" coils to digital information by gating clock pulses into a binary counter. The number stored between the start and stop times represents the coordinate value.

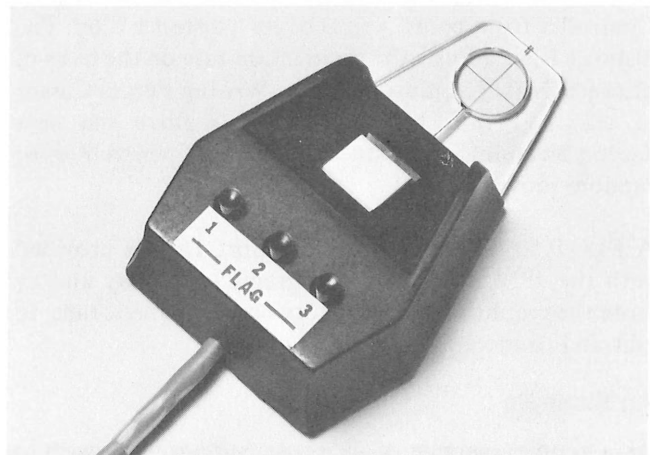
By pulsing from both ends simultaneously, two sets of values are produced which are calculated by the on-board processor resulting in a highly accurate digital coordinate.

The same process is repeated for the Y axis and this data is placed on the GPIB port ready for direct interfacing.

A free-hand Writing Pen is a part of the system, used like an ordinary pen or pencil. It has a built-in switch that is activated by pressing the ball point against the Tablet to start and stop the X and Y count during the digitizing process. The Writing Pen is supplied with an inking ball-point filler. A dry refill is also included, for applications where the digitizing surface must be kept clean (such as digitizing points from a graph in a book).

An alternative to the Writing Pen is a four-button Cursor. This cursor has crosshairs for exacting accuracy and a button to activate the digitizing process. In addition to the main button, the Cursor has three user-definable buttons that can be used for program control. For example, when an appropriate INPUT instruction occurs in a 4051 BASIC program, the Tablet returns ASCII coordinates and an ASCII character status byte through the GPIB. The status byte indicates which combination of Cursor buttons have been depressed. The status byte can be tested to decode the button combination for branching to different parts of the program. Some functions commonly assigned to the Cursor buttons are DRAW, MOVE, delete the last point entered, halt the input process, and rotate the component to be drawn.

The Tablet Controller supplies all operating voltages and



A Four-Button Cursor is an alternative to the Tablet Writing Pen.

currents, provides the controls for choice of operating modes and choice of rate of coordinate pair production, a counter and a microprocessor for system control and self-calibration. The controller output will transfer X and Y data, status, and required operating pulses at TTL levels for control and interface purposes through the GPIB. Buttons on the Tablet Controller control the manner in which points are digitized, along with appropriate indicator lights.

Several Operating Modes

The simplest mode of operation is the POINT mode. You digitize a single point each time you activate the Writing Pen or Cursor. This mode is useful for specifying individual points on the Tablet surface, such as the endpoint of a straight line, or for indicating the location for an object to be drawn.

A **STREAM** mode is also available; it is particularly useful when tracing or drawing freehand. In this mode you continuously generate coordinates and status while the Pen or Cursor is on the Tablet and activated. **STREAM** mode generates a series of points.

The Tablet Controller contains time and distance filters to regulate the number of point coordinates generated in **STREAM** mode. Filters reduce the amount of data

stored to manageable proportions, and reduce the effects of random pen or cursor movements.

The time filter adjusts the digitization rate on the basis of elapsed time; it is implemented by moving a slide control on the Tablet Controller front panel. You can use a slow rate when storage capacity is a consideration, such as when digitizing an extensively detailed map. Or you can use a faster rate in making curves or drawing quickly, such as when tracing a signature.



When you press the INCRE button on the Tablet Controller front panel, you activate a distance filter. The distance filter adjusts the digitization rate on the basis of distance. Now you must move the Writing Pen or Cursor at least .01 inch before you can digitize the next acceptable point. This reduces the effect of pen tremors or random movements.

A PLOT 50: Graphic Tablet Support Tape is provided with the 4956. It contains programs to display and/or store the graphic data in memory or on magnetic tape, to edit and to merge graphic data.

An Example

As a simple example of 4956 use, suppose you wish to generate a display of a given triangle, along with the coordinates for calculations. You would first place the drawing of the triangle on the Tablet surface. Then you would load a simple BASIC program into the 4051 and begin the data entry. You would identify each point of the figure by either touching it with the Writing Pen, or by placing the crosshairs of the Cursor over the point and depressing a Cursor button. A portion of your program calculates the length of one leg of the triangle as follows:

Variable Z is the distance between the two specified points in Tablet units. A conversion to inches is easily performed in statement 420 with a division by 200 (=4000 units/20 inches).

```
300 REM--TRANSFER THE COORDINATES AND STATUS OF A POINT
310 INPUT @B:X1,Y1,A$
320 PRINT USING 330:"X1= ",X1,"Y1= ",Y1,"A$= ",A$
330 IMAGE 4A,4D,5X,4A,4D,5X,4A,2A
340 REM--TRANSFER THE COORDINATES AND STATUS OF ANOTHER POINT
350 INPUT @B:X2,Y2,A$
360 PRINT USING 330:"X2= ",X2,"Y2= ",Y2,"A$= ",A$
370 REM--CALCULATE THE DISTANCE BETWEEN THE POINTS
380 Z=SQR((X2-X1)^2+(Y2-Y1)^2)
390 PRINT USING 400:"Z= ",Z
400 IMAGE 4A,4D
410 REM--CONVERT THE DISTANCE FROM TABLET UNITS TO INCHES
420 I=Z/200
430 PRINT USING 440:"I= ",I
440 IMAGE 4A,2C,2D
```

An example of the output generated by this portion of the code is as follows:

```
X1= 772      Y1= 1276      A$= 1
X2= 2271     Y2= 1848      A$= 1
Z= 1604
I= 8.02
```

By merely touching the three corners of a triangle, you can display the triangle and have the coordinates available in storage for calculations such as perimeter, distance and area.

Using the 4956 Graphic Tablet to identify a series of specified points by coordinate pairs has broad applications in many areas. These areas include:

curve integration	cartography
structural analysis	computer and design
system design (traffic, electrical, etc.)	interactive
schematics	graphics
land analysis	space planning
architecture	medicine
	freehand design

If you are entering data into the 4051 in an application that is inherently graphic, the 4956 Graphic Tablet is a most convenient tool. Using the Tablet can save you time, by allowing data entry without the keyboard. It increases accuracy over hand measurements, and it saves effort because points are merely touched, not measured and interpreted. It converts tedious, error-prone tasks into efficient and accurate ones. There's no POINT in doing it any other way.

REPRINTED FROM TEKNIQUES, VOL 2 NO 7 — 4050 SERIES APPLICATIONS LIBRARY NEWSLETTER

New ROM Pack—Adds Fast Graphing and Data Analysis Functions to the 4051

by Dave Barnard

The 4051R07: Signal Processing ROM Pack #1 adds seven new functions to the 4051. The functions locate minimum, maximum, threshold crossing, determine

derivative (slope), integral (area) or perform fast unlabeling screen graphing of data with just one 4051 BASIC statement per function.

You can execute all seven functions — MIN, MAX, CROSS, DIF2, DIF3, INT, DISP — directly from the 4051 keyboard, or you can incorporate them into a BASIC program. ROM Pack algorithms perform 2 to 10 times faster than equivalent BASIC programs, and the functions consume no memory. The functions are added

by simply plugging the 4051R07 into one of the two slots provided in the 4051.

The data must be able to fit into 4051 memory as a sequence of numeric values. The only special restriction on data is that it must be in a one dimensional array when accessed by the ROM Pack functions. Data may be sales revenue per period, strip chart data or anything in between.

The seven functions are:

- "MIN" (Minimum) performs a fast search of the data and returns both the lowest numeric value and its location in the data array.
- "MAX" (Maximum) is similar to "MIN" but returns the value and location of the largest numeric value in the data array.
- "CROSS" (Crossing) locates points where data meets or exceeds a threshold. The threshold and crossing (1st, 2nd, or nth) are programmable. "CROSS" simplifies comparing large amounts of data to a baseline. Determining when actual data exceeds forecast (sales, for example), is simplified.
- "DIF2" (2 point difference) provides a simple forward two-point derivative (slope) of specified data. Time-series analysis, contour studies, inflection point determination are all much easier using a simple call "DIF2" statement. "DIF2" is designed to make accurate slope estimates of rapidly varying data.
- "DIF3" (3 point difference) resembles "DIF2," but uses a three point algorithm for closer approximation to an ideal derivative. "DIF3" will yield more accurate results than "DIF2" for data that is less varied.
- "INT" (Integral) determines the area under the curve of specified data using a trapezoidal approximation to an ideal integral. The "INT" function can rapidly convert sales-per-period data into annual or total sales, or can rapidly handle area determination for analytical applications in several scientific disciplines.
- "DISP" (Display) simplifies graphing of raw data or results of other commands. Graphing is accomplished in half the time required by an equivalent routine programmed in 4051 BASIC.

Even a minimal use of these added functions can simplify the use of graphics on the 4051. For example, data or process results of unknown numeric range can be graphed with the MIN, MAX, DISP and one other command — WINDOW built into the 4051 BASIC language. Use "MIN" and "MAX" to find the data limits and assign them to variables used by the WINDOW command. Then use

"DISP" to give unclipped graphing of all the data. The ROM Pack permits putting all the data on the screen even if the data range is unknown in just four 4051 BASIC statements:

```
4 CALL "MAX",A,G,I
5 CALL "MIN",A,L,J
6 WINDOW 1,N,L,G
7 CALL "DISP",A
```

Fig. 1. Four line data graphing routine. Line 4 assigns MAXimum values to Y2; Line 5 assigns MINimum values to Y1; Line 6 uses these two values plus total number of points (2N) to set the window. Line 7 graphs the array.

The ability to automatically set the graphics window to fit the data is important whenever the data range is unknown. Even if the data limits are known, the derivative or integral of the data may not be known exactly beforehand. The program listed graphs the N values found in array "A" every time User-Definable key #1 is depressed. All the variables must be defined before using the routine, or an error message results. A summary of the commands and their parameters is included in Figures 2 and 3.

```
CALL "MAX",a,M,I
CALL "MIN",a,M,I
CALL "CROSS",a,t,I
CALL "CROSS",a,t,I,n
CALL "DIF2",a,B
CALL "DIF3",a,B
CALL "INT",a,B
CALL "DISP",a
```

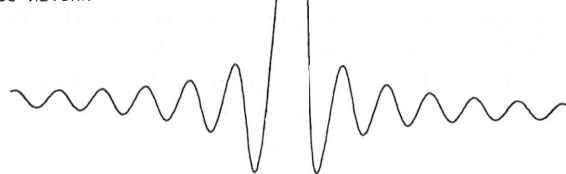
Fig. 2. Feature commands and parameters.

```
a VECTOR ARRAY INPUT
B VECTOR ARRAY OUTPUT
I INDEX OUTPUT
M MINIMUM OR MAXIMUM OUTPUT
n Nth CROSSING INPUT
t THRESHOLD INPUT
```

Fig. 3. Legend for parameters.

Other functions could be assigned to the remaining User-Definable keys. For added information on this ROM Pack, you should contact your nearest Tektronix representative. Ask about 4051R07: Signal Processing ROM Pack #1.

```
4 CALL "MAX",A,G,I
5 CALL "MIN",A,L,J
6 WINDOW 1,2,N,L,G
7 CALL "DISP",A
100 REM SIN(X)/X FUNCTION
110 ON SIZE THEN 220
120 N=200
130 L=-N*0.2
140 DIM A(2*N)
150 FOR K=1 TO N*2
160 L=L+0.2
170 A(K)=SIN(L)/L
200 NEXT K
210 END
220 A(K)=1
230 RETURN
```



A fast graph of SIN(X)/X using three functions of new 4051R07 ROM Pack.

The 4051R06 Editor: Searching Operations

by Cathy Cramer

In the last issue of TEKNiques, we talked about the 4051R06 EDITOR and its applications. This time, we'll describe some special searching operations you can perform using the EDITOR.

The 4051R06 EDITOR is a ROM pack. Installing the ROM pack and typing CALL "EDITOR" turns the 4051 into an ASCII text editor that can handle text of any kind. While the EDITOR is in control, you have 29 commands at your disposal. All are specially designed for easy text handling, but the most powerful are the searching commands.

Searching commands simplify the editing process. They quickly locate occurrences of a particular string of characters in the text, and simultaneously replace, edit, or delete those characters. Searching commands can make one correction, or many corrections at once. They're ideal for major changes to programs and text. But they also make simple everyday editing tasks go a lot faster and easier.

The 4051R06 EDITOR has two searching commands, SEARCH and NLSEARCH (No List SEARCH). Each has several forms that share the same keyword, but differ in syntax and function. SEARCH has four forms: SEARCH and List Line, SEARCH and Replace String, SEARCH and Edit Line, and SEARCH and Delete Line. NLSEARCH has two forms: NLSEARCH and Replace String, and NLSEARCH and Delete Line. The two forms of NLSEARCH are like the equivalent forms of SEARCH, but don't provide a listing of the changed or deleted lines. Table 1 summarizes the forms of the SEARCH and NLSEARCH commands.

Both SEARCH and NLSEARCH search the text, or some portion of the text, for any character string you want. You enter S for SEARCH or NL for NLSEARCH, then the string you want, enclosed in quotation marks. Additional entries tell the EDITOR which form of the command you want. (For example, a SEARCH command that ends with an asterisk is the SEARCH and Delete Line form of the command. A SEARCH command that ends with a semicolon is the SEARCH and Edit Line form of the command.)

A single SEARCH or NLSEARCH command can make major corrections to programs or text in just seconds. The same corrections might take hours to do from the

Command	What it Does	Example
SEARCH and List Line	Lists lines that contain the string. You can send the listing to any tape storage or printing device by specifying a device number.	S "SET"
SEARCH and Replace String	Replaces occurrences of the string with another string, and lists the changed lines. You can send the listing to any tape storage or printing device.	S "PRINT", "PRINT@33:"
SEARCH and Edit Line	Recalls lines containing the string to the line buffer for editing. The lines are recalled and displayed one at a time.	S 1,10 "Mr.";
SEARCH and Delete Line	Deletes lines containing the string, and lists the deleted lines. You can send the listing to any tape storage or printing device.	S @33:"NO REPLY"*
NLSEARCH and Replace String	Replaces occurrences of the string with another string. (Does not list changed lines.)	NL "ON ERROR", "ON SIZE"
NLSEARCH and Delete Line	Deletes lines containing the string. (Does not list deleted lines.)	NL "TRACE"*

TABLE 1

Note: You can include line numbers in any SEARCH or NLSEARCH command, to specify the portion of the text you want to search.

keyboard. For example, suppose you're working on someone else's subroutine, planning to incorporate it into your own BASIC program. You know that both the subroutine and your program use variable name A1, but for different purposes. So you want to change A1 to A2 everywhere it's mentioned in the subroutine. Unless you have an EDITOR ROM pack, you have to check all the lines yourself, and find every mention of A1. For each one, you have to press the RECALL LINE key, position the cursor to A1, and type A2. You then move on to correct another line. If you've been through this procedure before, you know how slow and tedious it is. But with the EDITOR ROM pack, the whole procedure is reduced to just one command. You use the SEARCH and Replace String form of the SEARCH command. All you do is enter S "A1", "A2". (See Fig. 1.)

Another example: Suppose you're writing or debugging a large program, and are having trouble keeping track of how and where variable B\$ is used. You could painstakingly inspect the program yourself, and write down on a piece of paper everything you find out about B\$. But a much easier alternative is to use SEARCH and List Line to obtain a listing of the lines that contain B\$. All you have to do is type S "B\$". The listing immediately appears on the display. (See Fig. 2.)

When you finish debugging the program, you can delete all SET TRACE and SET NORMAL statements using

```

L
: 2625 REM SORT ROUTINE
: 2630 FOR A1=1 TO N
: 2635 X2=A1
: 2640 FOR N2=A1 TO N
: 2645 IF P(N2)=P(X2) THEN 2655
: 2650 X2=N2
: 2655 NEXT N2
: 2660 X0=P(X2)
: 2665 P(X2)=P(A1)
: 2670 P(A1)=X0
: 2675 NEXT A1
: 2680 RETURN

S "A1","A2"
: 2630 FOR A2=1 TO N
: 2635 X2=A2
: 2640 FOR N2=A2 TO N
: 2665 P(X2)=P(A2)
: 2670 P(A2)=X0
: 2675 NEXT A2

L
: 2625 REM SORT ROUTINE
: 2630 FOR A2=1 TO N
: 2635 X2=A2
: 2640 FOR N2=A2 TO N
: 2645 IF P(N2)=P(X2) THEN 2655
: 2650 X2=N2
: 2655 NEXT N2
: 2660 X0=P(X2)
: 2665 P(X2)=P(A2)
: 2670 P(A2)=X0
: 2675 NEXT A2
: 2680 RETURN

```

Fig. 1. The EDITOR command S "A1", "A2" replaces occurrences of A2 with A2, and lists all changed lines. Initial and final listings are obtained by entering L for LIST.

```

L
: 3200 A$=" "
: 3210 FOR N1=1 TO 22
: 3220 READ B$
: 3230 K$=A$
: 3240 A$=K$&B$
: 3250 NEXT N1
: 3260 DELETE B$,K$
: 3270 LI=0
: 3280 DIM F(9),G(9)
: 3290 DIM B$(12),C$(25)

S "B$"
: 3220 READ B$
: 3240 A$=K$&B$
: 3260 DELETE B$,K$
: 3290 DIM B$(12),C$(25)

```

Fig. 2. The EDITOR command S "B\$" lists the lines that contain B\$. The initial listing is obtained by entering L for LIST.

two SEARCH and Delete Line commands, S "TRACE"* and S "NORMAL"*. If you need to correct lines containing PRINT, but don't want to make the same change to each line, you can use SEARCH and Edit Line, and enter S "PRINT";. This successively recalls lines containing PRINT to the line buffer, for you to edit however you like using the keyboard keys. SEARCH and Edit Line saves you time because the EDITOR not only finds all the lines for you, but also automatically recalls each one, just as if you had pressed the RECALL LINE key.

There are literally hundreds of other uses for the EDITOR's searching commands. You can search for any string, and list it, replace it, delete it, or edit it by hand.

And don't forget that the EDITOR lets you work with any kind of text, not just BASIC programs. You can edit

```

L
:Brockway, Marius E.
:Carmichael, David -PAID
:Ellis, Terry L.
:Foster, Alice -PAID
:Gardner, Keith W.
:Harvey, Richard A.
:Hillstrom, A. A. -PAID
:Kearney, John D. -PAID
:Keller, Suzanne -PAID
:Lentz, Robert F.

FIN8
PRI@33:LIST OF THOSE WHO PAID

S@33:"PAID"*

L
:Brockway, Marius E.
:Ellis, Terry L.
:Gardner, Keith W.
:Harvey, Richard A.
:Lentz, Robert F.

FIN8
OLD
L
:LIST OF THOSE WHO PAID
:
:Carmichael, David -PAID
:Foster, Alice -PAID
:Hillstrom, A. A. -PAID
:Kearney, John D. -PAID
:Keller, Suzanne -PAID

```

Fig. 3. The EDITOR command S @33: "PAID"* deletes lines containing PAID, and stores them on file 8 of the internal tape. An initial listing is obtained by entering L for LIST. Then FIN8 and PRI@33: LIST OF THOSE WHO PAID open file 8 for access. L for LIST shows that the lines have been deleted; and FIN8, OLD, and L show that the deleted lines are now stored on the tape.

FORTRAN, COBOL, data, free text—any set of ASCII characters. Fig. 3 shows a sample use for the SEARCH command when the text is a list of names and notations instead of a program. In this example, SEARCH and Delete Line separates out a subset of the text, and stores it on a different file. The command S @33: "PAID"* deletes the clients who've paid their bill, and lists (stores) them on file 8 of the internal tape.

Special Characters

To help you with your searching operations, the EDITOR provides four special characters. All four can be used in SEARCH or NLSEARCH commands, and have special meanings that make searching easier. The four characters are ~, #,], and _ by default, but you can cancel or change these assignments at any time.

The first is a "wildcard" character that allows you to search for any ASCII character. For example, you can search for any character that is immediately followed by \$ (like A\$, B\$, C\$ and so on). The second is an "any digit" character you can use to locate any of the digits 0 through 9. For instance, you can search for the character X immediately followed by a digit (as in X1, X2, X3, and X4). The third is an end-of-record character that allows you to locate, delete, or insert CARRIAGE RETURNS in the text. The fourth is an "all but" character that lets you search for all but one specified character. For example, you can find occurrences of @ that are *not* followed by 3 (like @1 and @A).

As a practical example, suppose you've written a large BASIC program, and used different string variable names A\$, B\$, C\$, D\$, and so on to store YES or NO responses from the user. You realize that one variable name could be used for all responses, so to conserve memory space, you decide to change all string variable names to E\$. You could enter separate NLSEARCH and Replace String commands; one to search for A\$ and change it to E\$, another to search for B\$ and change it to E\$, another to change C\$ to E\$, and so on. A simpler way is to use the "wildcard" character ~, and enter a single command, NL "~\$", "E\$".

```

L
: 1760 PRINT "IS THIS A CONTINUATION OF THE SAME PROBLEM?";
: 1770 INPUT A$
: 1780 IF A$="Y" THEN 2140
: 1790 GOSUB 430
: 1800 PRINT "IS DATA ON PROGRAM TAPE?";
: 1810 INPUT B$
: 1820 IF B$="Y" THEN 1950
: 1840 E(7)=1
: 1850 PRINT "IS DATA TO BE READ FROM EXTERNAL DEVICE?";
: 1860 INPUT C$
: 1870 IF C$="N" THEN 1930
: 1880 PRINT "DO YOU WANT TO ENTER DATA FROM THE KEYBOARD?";
: 1890 INPUT D$

NL "~$", "E$"

L
: 1760 PRINT "IS THIS A CONTINUATION OF THE SAME PROBLEM?";
: 1770 INPUT E$
: 1780 IF E$="Y" THEN 2140
: 1790 GOSUB 430
: 1800 PRINT "IS DATA ON PROGRAM TAPE?";
: 1810 INPUT E$
: 1820 IF E$="Y" THEN 1950
: 1840 E(7)=1
: 1850 PRINT "IS DATA TO BE READ FROM EXTERNAL DEVICE?";
: 1860 INPUT E$
: 1870 IF E$="N" THEN 1930
: 1880 PRINT "DO YOU WANT TO ENTER DATA FROM THE KEYBOARD?";
: 1890 INPUT E$

```

Fig. 4. The EDITOR command NL "~\$", "E\$" changes A\$, B\$, C\$, and D\$ to E\$. Initial and final listings of the text are obtained by entering L.

Fig. 4 shows a simple example of how this works. The EDITOR understands ~ to mean "any character," and looks for any character followed by \$. It finds A\$, B\$, C\$ and D\$ and changes them all in one step to E\$.

Now suppose there's a bug somewhere in your program, and you want to list and examine all numeric initializa-

tion statements like A=0, E=1.25, and X=1.0. If you've used a lot of variables, you don't want to have to search for each one individually. And a command like S "A=" finds extra statements you don't want to look at yet, such as A=A+V0. So you use the special "digit" character # and enter S "=#". The EDITOR understands the symbol # to mean "any digit 0 through 9," and looks for an equals sign followed by a digit. It finds A=0, E=1.25, X=1.0, and so on. (See Fig. 5.)

```

L
: 1220 PRINT "ENTER # OF VALUES:";
: 1230 INPUT X
: 1240 GOSUB 1660
: 1250 A=0
: 1260 E=1.25
: 1270 X=1.0
: 1280 GO TO 1030
: 1290 Y=2
: 1300 GOSUB 2630
: 1310 GO TO 1050
: 1320 B(4)=B(4)+1

S "=#"
: 1250 A=0
: 1260 E=1.25
: 1270 X=1.0
: 1290 Y=2

```

Fig. 5. The EDITOR command S "=#" locates and lists lines containing an equals sign immediately followed by a digit. The initial listing is obtained by entering L for LIST.

These are only a few of the ways you can use the EDITOR's searching commands and special characters. If you have an EDITOR ROM pack, you've probably found many other uses; ones that are tailored to your own specific needs. Next month, look for a brief overview of how to perform sorting operations using the EDITOR.

The 4051R06 EDITOR: Sorting Operations

by Cathy Cramer

(This is the third in a series of articles on the 4051R06 EDITOR ROM pack.)

The 4051R06 EDITOR ROM pack provides two sorting commands, SORT and REVSORT (REVERSE SORT). Both allow you to sort data or text of any kind. They're a lot faster than BASIC programs that perform the same sorting operations. And because they're ROM pack commands, SORT and REVSORT don't take up any 4051 memory space.

SORT and REVSORT rearrange lines according to the ASCII values of characters in certain positions within each line. You specify the positions when you enter the command. For example, SORT 100,500:1,2,3 rearranges lines 100 through 500 according to the ASCII values of the first three characters in each line.

You can specify as many character positions as you want, and in any order you want. Sorting begins with the last position you specify, and ends with the first position. For example, REVSORT 100,500:4,3,1,2 rearranges lines 100 through 500, first according to the second character in each line; then according to the first character; then according to the third; and finally, according to the fourth character.

SORT arranges the lines so the ASCII values increase from line to line. You can use it to alphabetize names, and arrange lists of numbers in increasing order. REVSORT arranges the lines so the ASCII values **decrease** from line to line. You can use it to "reverse alphabetize" names, and arrange lists of numbers in decreasing order.

In the example, an initial listing obtained by entering L for LIST shows that the text consists of fifteen names. The command SORT 1,15:1,2,3 is given to sort lines 1 through 15 according to the first three characters in each line. (For this particular list, no two names are identical in the first three or more positions; so you need only specify positions 1, 2, and 3 in the SORT command to correctly alphabetize the list.)

After the SORT command, a new listing shows that the names are completely alphabetized. The line numbers to the left of each line have been removed by the SORT command. You can obtain new line numbers by executing a RENUMBER command.

The following example shows the SORT command being used to alphabetize a list of names:

```
L
1:Zimmerman, Neil D.
2:Millstrom, A. A.
3:Carlmichael, David
4:Brockway, Marius E.
5:Harvey, Richard A.
6:Taylor, Owen
7:Gardner, Keith W.
8:Foster, Alice
9:Siebold, William B.
10:Lentz, John F.
11:Sloan, Irene
12:Pollock, Robert
13:Kearney, John D.
14:Ellis, Terry L.
15:Keller, Suzanne
```

SORT 1,15:1,2,3

```
L
:Brockway, Marius E.
:Carlmichael, David
:Ellis, Terry L.
:Foster, Alice
:Gardner, Keith W.
:Harvey, Richard A.
:Millstrom, A. A.
:Kearney, John D.
:Keller, Suzanne
:Lentz, John F.
:Pollock, Robert
:Siebold, William B.
:Sloan, Irene
:Taylor, Owen
:Zimmerman, Neil D.
```

In the following example, the REVSORT command is used to arrange numbers in decreasing order:

```
L
1:SALESMAN
2:-----
3:CLIFFORD          $ 920,000
4:ETHERIDGE        $ 150,990
5:GLINES            $ 249,000
6:KINTZ             $ 820,000
7:LENZ              $ 950,300
8:NOAKES            $ 433,600
9:PARKER            $ 1,320,500
10:SAUER             $ 1,570,000
11:THOMPSON          $ 720,400
12:VAN DYKE          $ 565,300
13:VERMIERE          $ 450,000
14:WENTWORTH        $ 1,102,000
```

REVSORT 3,14:33,34,35,36,37,38,39,40,41

```
L
1:SALESMAN
2:-----
:SAUER              $ 1,570,000
:PARKER             $ 1,320,500
:WENTWORTH          $ 1,102,000
:LENZ               $ 950,300
:CLIFFORD           $ 920,000
:KINTZ              $ 820,000
:THOMPSON           $ 720,400
:VAN DYKE           $ 565,300
:VERMIERE           $ 450,000
:NOAKES             $ 433,600
:GLINES             $ 249,000
:ETHERIDGE          $ 150,990
```

The initial listing obtained by entering L shows a list of names and amounts sold. The REVSORT command is used to rearrange the lines so that the numbers in the AMOUNT SOLD column decrease.

The numbers appear in positions 33 through 41 of lines 3 through 14, so the command given is REVSORT 3,14:33,34,35,36,37,38,39,40,41. After the REVSORT command is executed, a new listing shows that the lines have been correctly arranged so that the numbers in the AMOUNT SOLD column decrease.

These articles on the Editor ROM have provided a basic understanding of how the EDITOR works, and what it can do for you. We haven't attempted to cover all of the EDITOR's commands and features. If you've found features or have applications that you'd like to share with other readers, write and let us know.

REPRINTED FROM TEKNiques, VOL 3 NO 4 – 4050 SERIES APPLICATIONS LIBRARY NEWSLETTER

Signal Processing ROM Pack #1 Speeds Up Numeric Sorting

by Dave Barnard

How could the Signal Processing ROM Pack #1 (4051R07 or 4052R07) possibly improve numeric value sorting, or even relate to such a task? Someone really did ask that question. The answer is that it relates very well, and significantly improves the performance of even the simplest sorting program. Although Signal Processing ROM Pack #1 was designed for waveform, signal, and time-series analysis,* its functions can be applied to **any** one-dimensional array of numeric values. It can also operate on numbers stored in 4050 Series Graphic System memory, for sorting operations.

With a few changes a bubble sort program can take advantage of the power and speed of the Signal Processing ROM Pack #1. The changes are easy to implement and easy to understand. Although more powerful sorting algorithms exist, they won't outrun the simple ROM Pack-enhanced bubble sort until the number of values sorted exceeds 1500 for the 4051, or 2400 for the 4052.

Bubble Sort Program

Let's look at a simple bubble sort program (Figure 1). Each time an inner loop consisting of lines 1110 through 1210 is executed, the numbers in the array are only partially ordered. For badly scrambled numbers, line 1090 will command entry of the loop almost N times, or as many times as there are numbers in the array. And the FOR...NEXT loop must examine N-1 values for each repetition. Therefore, sorting time could be proportional

to N^2 (N = the number of values to be sorted). Even if all the numbers are in sequence, the FOR...NEXT loop must be executed once; that's N-1 executions of some of the lines in the loop. But, when the quantity of numbers to be sorted is small, the bubble sort is an easy solution that is often fast enough.

However, when sorting only 80 numbers, this routine takes approximately two minutes. The lengthy execution occurs because line 1130 simply compares two neighboring values to determine if they're out of sequence. The limited information provided by this comparison only allows partial re-ordering of the values during each run of the FOR...NEXT loop.

```

L101000,1230
1000 REM
1010 REM VARIABLES: *****STAPT OF BUBBLE SORT*****
1020 REM *****INPUT*****
1030 REM A - THE LINEAR ARRAY TO BE SORTED INTO
1040 REM INCREASING ORDER
1050 REM N0 - THE NUMBER OF ELEMENTS OF A
1060 REM *****OUTPUT*****
1070 REM A - THE SORTED ARRAY
1080 A2=INT(N0/2)+1
1090 IF A2<3 THEN 1220
1100 A3=0
1110 FOR A1=1 TO A2-2
1120 REM CHANGE / TO / FOR SORTING INTO DECREASING ORDER
1130 IF A(A1)<A(A1+1) THEN 1190
1140 A3=A1+1
1150 REM SWITCH ELEMENTS
1160 A4=A(A1)
1170 A(A1)=A(A3)
1180 A(A3)=A4
1190 NEXT A1
1200 A2=A3
1210 GO TO 1090
1220 RETURN
1230 REM *****END OF BUBBLE SORT*****

```

Fig. 1. Simple Bubble Sort program.

But suppose that the entire array could be searched, instead of comparing just two points in the array. The approach would be a more direct, logical series of steps:

1. Examine all the numbers until you find the biggest.

*A previous TEKNiques article (Vol. 2 No. 7) described the seven functions that the 4051R07 performs.

- Put the biggest number at the end of the array (and save the old end number in the place where you found the biggest one).
- Narrow your search as though the array is shorter by one number (don't again look at numbers you have sorted).
- Go back to 1 and repeat until you have to search only one number—you're done.

That's closer to the way people sort numbers. For a collection of numbers (N of them) the process involves searching for the largest (or smallest) number only N-1 times.

Signal Processing ROM Pack #1-Enhanced Bubble Sort

With the "MIN" or "MAX" commands of the Signal Processing ROM Pack #1, you can sort numbers in the way just described. The sort program (Figure 2) replaces the one in Figure 1. Line 1110 determines whether the "MAX" or "MIN" command is used for ascending or descending sorting. The variable A1 always points to the last number in the array of unsorted numbers. The CALL Z\$ A,V,I statement finds the largest (or smallest when Z\$ = "MIN") number and its location, providing much stronger direction (more information) for the sorting operation. The limits of the search ("MAX" or "MIN" command) are decreased by one each time through the FOR...NEXT loop, by re-dimensioning the array. Only N-1 times through the loop sorts all N of the numbers.

The execution time is still proportional to N^2 but only N (approximately) loops occur in BASIC. Searching for the largest number among N numeric values is delegated to the ROM Pack. That is, the loop contained in statements 1110 to 1190 in Figure 1 is replaced by the firmware in Figure 2 (statement 1140) with the resultant increase in speed. The 80 number numeric sort that took two minutes without the ROM Pack Function takes only *five seconds* using the "MAX" command of the ROM Pack.

Enhanced Bubble Sort vs. Quick Sort

For larger sizes of sorting problems a different routine is useful: the Quick sort. Such a routine is contained in the General Utilities Program Vol. 1 (4050A08). The Quick sort conquers the sorting problem by subdividing it which makes the execution time proportional to $N \log_{10} N$ instead of N^2 . However, for smaller sorting tasks, the ROM Pack-assisted bubble sort is faster than even the Quick sort. Eventually its proportionality to N^2 slows it down and the Quick sort algorithm becomes faster.

The table in Figure 3 compares the measured times for performing sorts on various numbers of values in the 4051. When approximately 1500 numbers are sorted, the Quick sort program from the General Utilities Vol. 1 pulls out in front.

```

L1S1000,1210
1000 REM *****START OF BUBBLE SORT*****
1010 REM MODIFIED SORT IS DIRECTED BY SPS ROM PACK
1020 REM VARIABLES:
1030 REM **INPUT**
1040 REM A - THE LINEAR ARRAY TO BE SORTED INTO
1050 REM INCREASING ORDER
1060 REM N0 - THE NUMBER OF ELEMENTS OF A
1070 REM **OUTPUT**
1080 REM A - THE SORTED ARRAY
1090 A2=INT(A0)
1100 REM CHANGE "MAX" TO "MIN" FOR SORTING INTO DECREASING ORDER
1110 Z$="MAX"
1120 FOR A1=A2 TO 2 STEP -1
1130 DIM A(A1)
1140 CALL Z$,A,V,I
1150 A2=A(A1)
1160 A(A1)=U
1170 A(I)=A2
1180 NEXT A1
1190 DIM A(A2)
1200 RETURN
1210 REM *****END OF SORT ROUTINE USING SPS #1*****

```

Fig. 2. Bubble Sort program modified to use 4051R07 ROM Pack.

For larger applications a 4052R07 Signal Processing ROM Pack #1 combined with the 4052 or 4054 provides a powerful tool. With the greater speed and larger memory nearly intractable problems become solvable. Figure 3 also compares a 4052R07-assisted bubble sort in the 4052 to a Quick sort in the same machine. The point at which the Quick sort method becomes faster than the simpler sorting program has moved to a new plateau (approximately 2400 points).

N (Number of Points)	4051 Bubble Sort (seconds)	4051+4051R07 Bubble Sort (seconds)	4052 Bubble Sort (seconds)	4052+4052R07 Bubble Sort (seconds)	4051 Quick Sort (seconds)	4052 Quick Sort (seconds)
50	45	3.1	10	too small	15	2.5
80	120	5.0	24	-	22	4.6
500	too long	90.0	995	14	202	45.0
1000	-	365.0	too long	46	450	85.0
2000	-	1199.0	-	165	990	192.0

Quick Sort Time = 4051R07 Time at N = 1500
Quick Sort Time = 4052R07 Time at N = 2400

Fig. 3. Execution time for sorting methods.

New ROM Pack Enhances Data Analysis Capabilities of 4050 Series

by Dave Barnard and Cliff Morgan

Have you ever had a set of evenly spaced data values and wanted to:

- Determine the frequency and amplitude of the cyclic components; or
- Highlight the cyclic content of the data and at the same time suppress most of the random ups and downs; or
- Measure the degree of association between that set and another data set even though one may lead or follow the other, and also determine the number of points of lead or lag that *maximizes* the amount of association; or
- Apply discrete time or digital filtering to the set?

If you have answered "yes" to any of the above, you may find the newest 4050 Series ROM Pack will solve your problem more simply and more quickly than you imagined.

The newest product, the Signal Processing ROM Pack #2 (FFT), is actually a pair. One, 4051R08, is compatible with the 4051. The other, 4052R08, is compatible with the 4052 and 4054.

The new ROM Pack lets you:

- Perform the Fourier Transform of an array of data containing from 16 to 1024 points.
- Calculate the autocorrelation of a data array of 8 to 512 points.
- Perform cross-correlation of two sets of data each containing from 8 to 512 points each (for every position of one relative to the other).
- Calculate the convolution of two sets of data points. One set represents the digital filter or any other discrete time system, the other set is a set of data to be processed.

Signal Processing ROM Pack #2 (FFT) begins where Signal Processing ROM Pack #1 left off. Its eight new commands include FFT, IFT, Convolution, Correlation plus four others for converting data to alternate formats or for pre-processing the data for easier display and analysis.

If you already use any of the functions mentioned you will find them easier than ever to use in their ROM Pack form. Each of them may be executed directly from the Graphic System keyboard or from a 4050 BASIC Program.

As with other ROM Packs, adding the new functions by plugging the new ROM Pack into one of the slots

provided, consumes no read write memory. All of the new functions are performed with just one statement per function. The FFT command, for example, can reduce a typical FFT program in 4050 BASIC (50 lines or more long) to a single statement; a memory saving.

With more space freed for data and analysis programs you would probably like to set your sight higher toward newer and tougher problems. If you were just thinking it would help if the ROM Packs were faster than their 4050 BASIC equivalents, you're right. We really did it this time!

FFTs in a Flash

Their speed and speed range is astounding. Installed in a 4051, the 4051R08 increases the speed of Fast Fourier Transform (FFT) operations by a factor of 7 over an equivalent 4050 BASIC program. On the 4051, FFT and related operations are slightly faster than a 4052 running equivalent BASIC programs. The already-fast 4052 and 4054 experience even greater enhancements of FFT calculation speeds due to their bit-slice architecture and micro-coded floating point operations. FFT or IFT transformation in a 4052 or 4054 equipped with a 4052R08 jumps to 22 times the speed of equivalent 4050 BASIC programs running on either system. And, when comparing the 4052/4052R08 system to a 4051 using the fastest FFT algorithm we could find in BASIC, the factor takes a quantum leap: over 100 to 1!

Table 1 illustrates speed increases when processing random arrays of data employing the new ROM Pack #2 FFT and IFT functions.

Table 1. Fourier Transform Computation Time				
Function	Computation Time In Seconds			
	4051 In BASIC	4051 + 4051R08	4052 In BASIC	4052 + 4052R08
512 points FFT	205	26	40	2
1024 points FFT	446	65	88	4
512 points IFT	207	27	42	2
1024 points IFT	446	64	90	4

The 4050 BASIC FFT algorithm is extremely fast: it contains no sine or cosine calculations, but rather uses a stored table of sine values only. Most of the FFT, or IFT, calculation times were devoted to indexing through various arrays and subroutines of the very fast algorithm. Because of this, the factor of only approximately 5 for the 4052 versus the 4051 resulted.

Repertoire of Waveform Analysis Functions

The advanced signal processing and data analysis commands are:

"FFT" (Fast Fourier Transform)—transforms a one-dimensional array into frequency domain data consisting of an interleaved array of its cosine (real) and sine (imaginary) spectral components.

"IFT" (Inverse Fourier Transform)—converts an array of interleaved frequency components (spectral) into its equivalent time series.

"UNLEAV" (un-interleave)—separates the interleaved complex results of the "FFT" command into two separate arrays to simplify display or manipulation of frequency data.

"INLEAV" (Interleave)—prepares real (cosine) and imaginary (sine) spectra contained in separate arrays into a format compatible with the "IFT" function.

"POLAR" (Rectangular to Polar)—converts and separates the intermixed complex output of the "FFT" command into separate arrays containing magnitude (distance) and phase (bearing) data.

"TAPER" (Programmable Cosine Window)—multiplies any chosen one-dimensional array by a cosine window whose tapering width is programmable. When the width is selected to be 0.5 (50%), the well known Hanning Window results.

"CORR" (Correlation)—performs the cross-correlation of two data arrays. If an array is cross-correlated with itself, the autocorrelation is obtained.

"CONVL" (Convolution)—forms the convolution series from two arrays of data. This is the discrete time (sampled data) analogue of the convolution integral.

Some Working Environments

An engineer might easily put the new ROM pack's capabilities to work analyzing signals or waveforms for laser fusion studies, component testing, power supply design, or audio-through-radio frequency design and analysis. Source signals might be captured by a Digital Processing Oscilloscope, a 7912AD or other digitizers including GPIB-compatible multimeters.

Potential uses include a chemist's analysis of strip chart data in the areas of biomedical and chemical data analysis. Or general time series analysis, for autocorrelation or cross-correlation of financial, business, or other data. (It will be particularly useful in this area, since the autocorrelation is more revealing than the single coefficient of correlation statistic provided by most statistics packages.) And, in the area of general linear system modeling, the ROM Pack can be used for discrete time (sampled) systems using convolution of the model (time series) by the input in economics, operations research, physics, and engineering.

An FFT Application

Since the FFT transforms a signal from the time domain

to the frequency domain, use this routine to study the frequency spectrum of a signal. For example, in harmonic distortion analysis, you compute the FFT of the signal, then derive the magnitude spectrum via the POLAR routine. Next, the amplitude of each harmonic is determined and the total harmonic distortion (THD) is computed. Measurement of total harmonic distortion is a common measurement in amplifiers and sound systems.

An IFT Application

The IFT command does the inverse operation of the FFT command. That is, it transforms a signal from the frequency domain to the time domain. When used in conjunction with the FFT command, IFT can be used to perform very selective digital filtering on a waveform sample. As an example, suppose that the magnitude spectrum of a signal has been obtained and that you wish to attenuate all harmonics of a signal to a certain specified level. This could be done by selectively multiplying all harmonics by a certain value. Following this procedure, you could then convert the signal back to the time domain by computing the IFT of the resulting spectrum.

A CORR Application

In a practical sense, correlation can be thought of as successively shifting (by some horizontal increment), multiplying, and integrating the two signals to be correlated. From a mathematical standpoint, correlation is achieved by computing the FFT of each signal to be correlated, then performing a complex conjugate multiplication on these FFTs, and finally taking the IFT of the resulting product. When the two signals being correlated are the same, the process is referred to as autocorrelation; when they are different, it is called cross-correlation.

A common application for autocorrelation is detecting the presence of signals buried in noise. When a noisy signal is autocorrelated, the result appears as a periodic waveform modulated with a triangular envelope. This technique of signal detection is used in biomedical studies, astronomy, tone control systems, and numerous other applications.

Cross-correlation is commonly applied in the detection and ranging of radar, sonar, and other pulsed waveforms, whereby the transmitted and received signals are cross-correlated to determine the delay (and hence distance) to a target. Cross-correlation is also useful in certain business applications for determining the association between sets of data (such as sales and pricing).

A CONVL Application

Like correlation, convolution can be thought of as successively shifting, multiplying, and integrating the two arrays (waveforms) to be convolved. However, in the case of convolution, one of the waveforms is reversed in time

before performing the shifting-multiplication-integration process. Mathematically, convolution is performed by computing the FFT of each signal to be convolved, multiplying these two FFTs, and then computing the IFT of the resulting product.

A common engineering application for convolution is determining the output of a linear, time invariant system (such as a passive filter or network). Given the input signal, $x(t)$, and the impulse response, $h(t)$, the output $y(t)$ can be predicted simply by convolving $x(t)$ with $h(t)$.

Getting Started

The SPS ROM Pack #2 command formats are shown in Figure 1.

```
CALL"FFT",a
CALL"IFT",a
CALL"POLAR",a,x,y,v]
CALL"CONVL",b,c,x
CALL"CORP",b,c,x
CALL"TAPER",aE,p]
CALL"INLEAU",d,e,x
CALL"UNLEAU",d,x,y
```

Fig. 1. Signal Processing ROM Pack #2 (FFT) Command Formats.

The variable legend is shown below:

```
a Vector array input/output
b,c Vector array inputs (Overwritten)
d,e Vector array inputs
x,y Vector array outputs
v Phase delay (Optional)
p Taper fraction (.5 is 50% width
tapering for the Hanning window)
```

The program contained in Figures 2 and 3 computes and displays the FFT of an array of data.

```
LIS100,195
100 N2=N-2+1
110 DIM X(N),A(N2),B(N2)
120 GOSUB 200
130 INPUT Z#
140 CALL "FFT",X
150 CALL "UNLEAU",X,A,B
160 GOSUB 300
170 INPUT Z#
180 CALL "POLAR",X,A,B
190 GOSUB 300
195 END
```

Fig. 2. FFT Main Program.

```
LIS200,600
200 REM DRAW THE N POINTS OF DATA IN FUNCTION FORM
210 CALL "MAX",X,Y2,I
220 CALL "MIN",X,Y1,I
230 WINDOW 1,N,Y1,Y2
240 PAGE
250 VIEWPORT 10,110,10,100
260 CALL "DISP",X
270 RETURN
300 REM THE UPPER GRAPH IS OF A(I), THE LOWER IS B(I)
305 PAGE
310 CALL "MAX",A,A2,I
320 CALL "MIN",A,A1,I
330 WINDOW 1,N2,A1 MIN 0,A2 MAX 0
340 VIEWPORT 10,110,50,90
350 FOR I=1 TO N2
360 MOVE I,0
370 DRAW I,A(I)
380 NEXT I
390 VIEWPORT 10,110,10,45
400 FOR I=1 TO N2
410 MOVE I,0
420 DRAW I,B(I)
430 NEXT I
440 RETURN
```

Fig. 3. Graphic Subroutines.

Lines 100 through 120 perform dimensioning and housekeeping.* The subroutine at line 200 graphs the waveform (Fig. 4.)

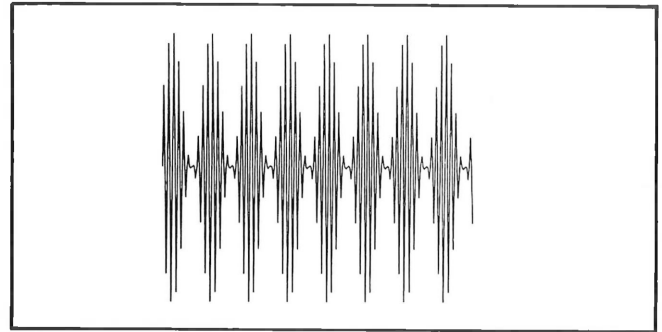


Fig. 4. A waveform to be transformed—256 points. This waveform was retrieved from a tape file and stored in array X.

Notice the calls to Signal Processing ROM Pack #1 in this graphic subroutine which extensively reduces the amount of coding and execution time.

Line 150 invokes the FFT. Line 160 separates the complex results (in array X) into separate real (A) imaginary (B) arrays. The subroutine at line 400 graphs the results in spectral form, shown in Figure 5.

A few more lines provide and graph the results in magnitude and phase (Polar) form as shown in Figure 6. Since the original waveform was a sine wave amplitude modulated by a sine function, the results are simple to interpret, and agree with the expected result. As anticipated, the frequencies consist of the center frequency (carrier) and sidebands resulting from the modulation process.

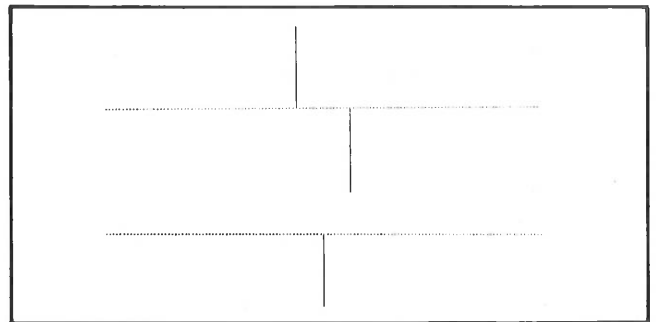


Fig. 5. Real (top) and Imaginary Transform Data for Figure 4.

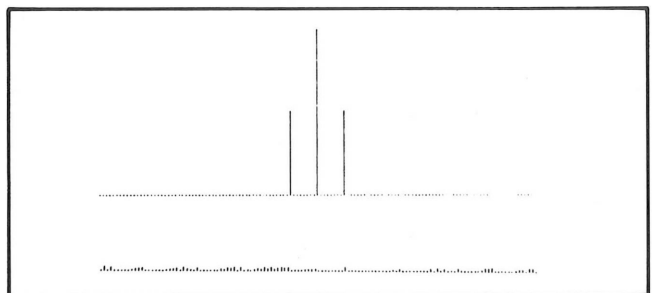
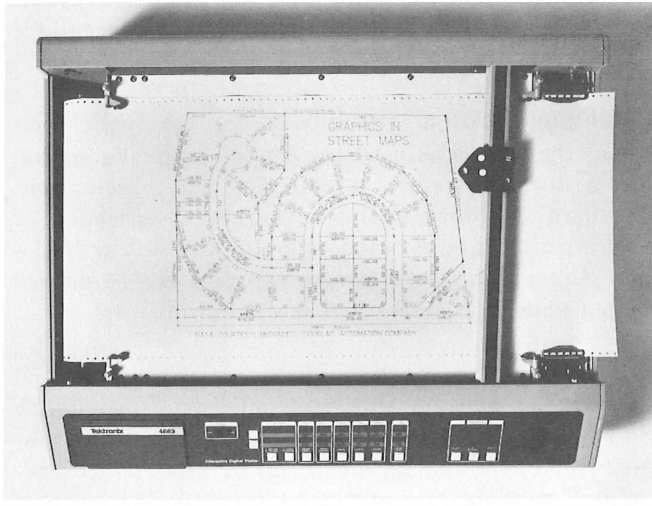


Fig. 6. Magnitude (top) and Phase Data—Figure 4 Transformed.

*Whenever FFT, IFT, convolution or correlation are used, array sizes must be a power of two, in keeping with the Sande-Tukey (decimation in frequency) algorithm the ROM Pack uses.

Expanding Your Plotting Capabilities: Introducing the 4663



The TEKTRONIX 4663 Interactive C-size Digital Plotter is an intelligent, powerful companion to the 4050 Graphic Systems.

by Terry Davis
TEKnikues Staff

Graphics are the heart of many applications for 4050 Series systems, graphics that show us complex data through an easy to grasp picture. Because graphics are so important, many systems today include a plotter to neatly record graphic output in a form that can be shared as well as saved. But the more pictures you plot, the more chances you get to bump into the boundaries of a small, local plotting device. Maybe you'd like a larger picture, or automated production of multiple plot copies, or two colors to highlight a graph. Now meet the 4663: an Interactive Digital Plotter that pushes back the boundaries of both size and usability in a local plotter.

The TEKTRONIX 4663 is a smart C-size plotter that represents an engineering first—the versatility of large plotters combined with the convenience and operating ease of a small plotter. The 4663 gives you a choice of plotting media; its electrostatic hold-down platen will accommodate any standard media size up to U.S. C-size (17 x 22") or European A2-size (420 x 594 mm). Larger plotting needs can be accommodated: The plotter will hold media up to 21 x 28-1/2" and draw plots up to 17.8 x 23.5" using one pen only. For automated production of multiple plot copies, an optional programmable paper advance for roll stock is available. And, like the 4662, plotting can also be done on acetate film for preparing transparencies, and the like.

The media choice is complemented by an equally wide choice of pens; fiber tip pens, ball point pens, and wet ink pens are all available in a variety of colors. The 4663 features a dual-pen carriage with programmable pen

selection for fast two-color plotting. And for each pen station, the operator can assign optimum pen pressure and pen velocity.

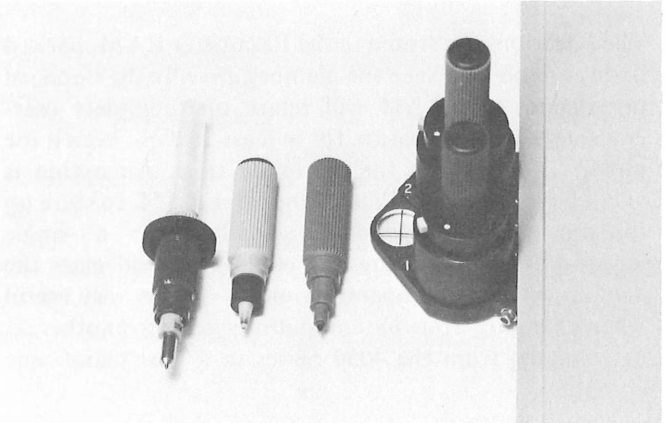


Fig. 1. The 4663 features a dual-pen carriage with programmable pen selection for fast two-color plotting.

Versatile but Simple

One design goal for the 4663 was increased flexibility—to provide a plotter that would be equally at home in diverse applications like printed circuit board layout, numerical control, mapping, drafting, report generation, and a host of others. But an equally important part of the design goal was that the versatility not bring a like increase in operating complexity. Ease of use is reflected in the plotter's simplified control panel, as well as in its unique parameter entry device (Fig. 2).

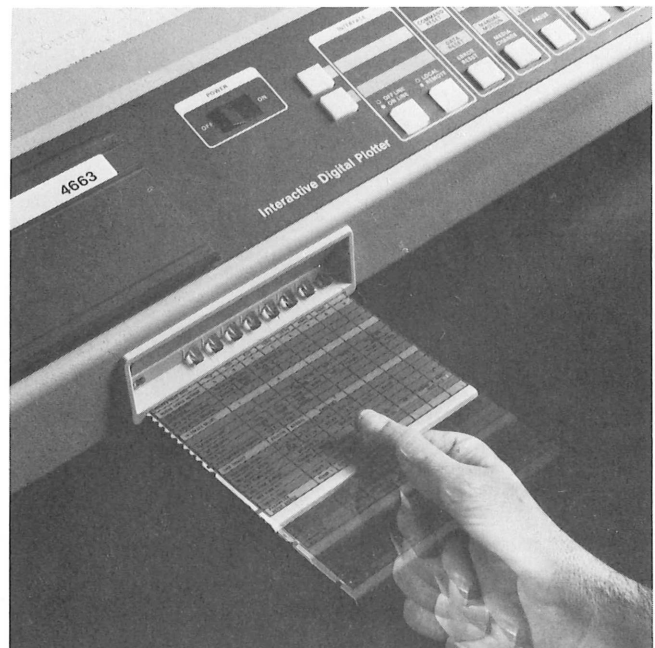


Fig. 2

The parameter entry device replaces the profusion of switches, jumper straps, and other rear panel arrangements common to most plotters and similar peripheral equipment. About a dozen system parameters, and a like number of interface parameters, can be preset using the parameter entry card. As the card is moved in or out, one line at a time, the lighting of the eight LED-lit pushbuttons indicate the parameter status. All that's required to change a parameter is to press its associated pushbutton.

The selections are stored in the Parameter RAM, backed up by a battery to keep the memory alive. In the standard instrument, this RAM will retain one complete user-environment specification for at least 90 days, even if the plotter is turned off for the entire time. An option is available to add additional Parameter RAM, to store up to four user-environment specifications; a single keystroke will recall any one of the four and place the 4663 in the selected operating mode. This is very useful when changing from one computing system to another, as in changing from the 4050 Series to a host mainframe computer.

Easy Front Panel

The front panel controls provide manual selection of a variety of commands, in addition to the selections available through the parameter entry device. Eight major control groups, in conjunction with two color-coded shift keys, allow control of over twenty functions from an uncluttered, easy-to-understand front panel (see Fig. 3). The "PLOT CONTROL" group, for instance, enables a pause in plotting with continued data buffering, a solid outline of the viewport, or tic marks indicating viewport corners.

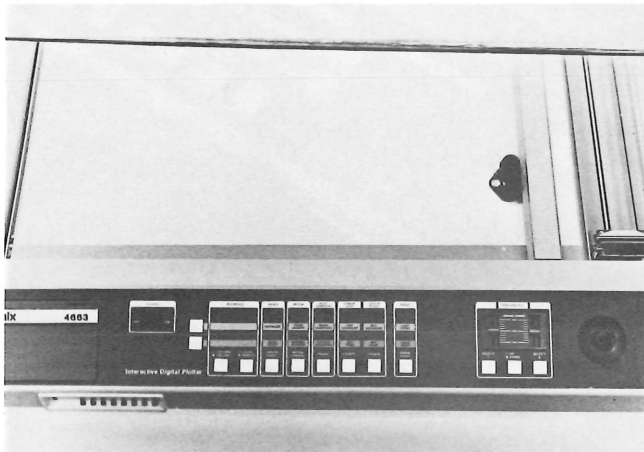


Fig. 3. The front panel controls over 20 functions.

The control provided by the front panel and the parameter entry device afford great flexibility in how the plotter is to be used. But few, if any, selections are required to "get a plot". They simply provide convenient local control over plot production without the bother of

program changes. For instance, graphics that have just been previewed at high speed on the display screen can be replotted as a C-size drawing, at maximum speed, using a ball-point pen and the fast coarse alpha characters. The same graphics can then be plotted again as an A-size drawing (in vertical format, as for a notebook) using wet ink pens and enhanced (high-precision) alpha characters. Both plots can be drawn without making changes to the source program.

Local configuration capability is ideal for applications where the plotter is used for output involving several media and pen types, several different presentations (standard drawings, text illustrations, overheads), or several different hosts or local storage devices. The 4663 is one plotter that's smart enough to provide the desired output under this wide range of conditions.

In many instances it's desirable to use only a portion of the page for plotting; this is accomplished through the SET VIEWPORT function. A special, often-used viewport (such as a mailing label) can be saved as a "user-defined" page via the parameter entry card. The viewport can also be set by the host computer via the VIEWPORT command.

The front-panel PAGE function can be used to locally calibrate the plotter to the absolute media dimensions, which will exhibit minute size variations as a function of humidity and printing variations. Other front-panel controls let you operate the plotter on- or off-line and in a local or remote mode.

Versatility in Graphics

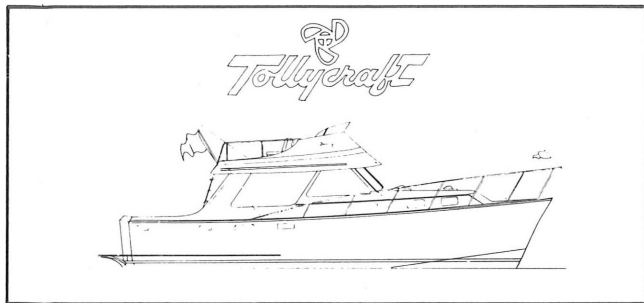
A comprehensive set of graphic plot commands allows a choice of dimensional coordinate units, the type of line to be drawn (solid, dashed, etc.), the coordinate type (absolute or relative), and the choice of which pen will be drawing.

The choice of dimensional coordinate units (graphic units) includes World Units and Device Units. (World Units are the units defined by the current window.) When Device Units are chosen you have a further selection of Addressable Device Units (ADUs), Graphic Device Units (GDUs), or millimeters.

ADUs provide device-dependent device addressing, with a numeric addressable range of from 0 to 4096 on the longest axis of the viewport. The range of the shortest axis is determined by the current aspect ratio.

GDUs are the standard unit for the 4050 Series; they provide device-independent addressing with a numeric addressable range of from 0 to 100 on the shortest axis of the viewport. Using GDUs ensures that any graphics containing only coordinates between 0 and 100 will be

plotted without clipping, no matter what the current aspect ratio is. For millimeters, the numeric range is the actual axis length in millimeters. These units give a plot of the same physical size, regardless of the current viewport size.

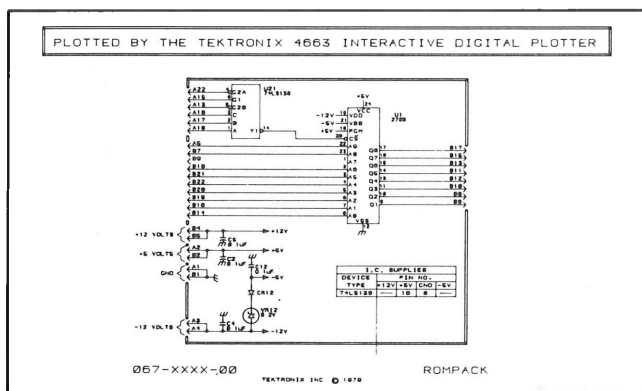


A circular interpolation option is also available to further enhance the graphics. This permits drawing circles or arcs with a single command. For circles, you need only to specify the radius, and the circle is drawn around the current pen location. Arcs are defined by specifying two points on the arc, which begins from its current pen location. You can choose the smoothness of the arc or circle, to meet the need for a quick overview or a high resolution finished plot.

Versatility in Alphanumerics

Graphics versatility in the 4663 is complemented by equal versatility in alphanumerics. Included are a self-contained character generator, provision for up to 15 character fonts (nine of which are resident in the 4663), and a host of commands that operate on the alphanumerics. One font can be designated as the standard font, and another as the alternate. The characters can be rotated, scaled up or down, or slanted, independent of the graphics and all under program control.

There are commands to set the character size and spacing, if you desire settings other than the default. You can also print centered characters, for applications such as identifying a line on a graph. And the pen can be moved specified X and Y distances in fractions of the current character size, to facilitate drawing superscripts and subscripts.



Sometimes you want to draw alphanumerics in a paragraph format. A Set Margin Separation command activates the right margin and allows you to set the number of character spaces that separates the left and right alpha margins. Carriage Return and Line Feed functions will be automatically performed to draw a string of alphanumeric characters within the margins established.

The alphanumerics capability is enhanced with options that allow down-loadable character sets, programmable macros, and circular interpolation.

Digitizing Capabilities

The 4663 can also be used to digitize a plot and transmit the data to a host or to the 4050 Series System. The joystick, the crosshair on the pen carriage, and the front-panel POINT switches, which allow the point to be designated as a DRAW, MOVE, or LAST point, are used in this function. Digitizing in one axis only is greatly simplified by using the Joystick Axis Disable command. This command allows you disable either the X or Y axis, or both.

That's quite a set of capabilities for a local plotter. And whether you consider these performance capabilities (and the technological advances behind them), or the simplified user interface, you'll see that the 4663 expands the horizons of local plotting abilities. Its capabilities make it an ideal recording companion for a 4050 Series Graphic Computing System, for even the most complex of plotting needs. And, any programs that will drive the 4662 will drive the 4663—just plug the plotter in and it goes. If you'd like to know more about the 4663, just ask your local Tektronix Sales Engineer.

Options

4663 INTERACTIVE DIGITAL PLOTTER STD is RS232

- | | |
|--------|---|
| OPT 1 | GPIB |
| OPT 4 | GPIB ONLY |
| OPT 31 | CIRCULAR INTERPOLATION & PROGRAMMABLE MACROS |
| OPT 32 | MATH CHARACTER SET & DOWN LOADABLE CHARACTERS |
| OPT 36 | PAPER ADVANCE |
| OPT 37 | ADDED DEFAULT PARAMETERS |

