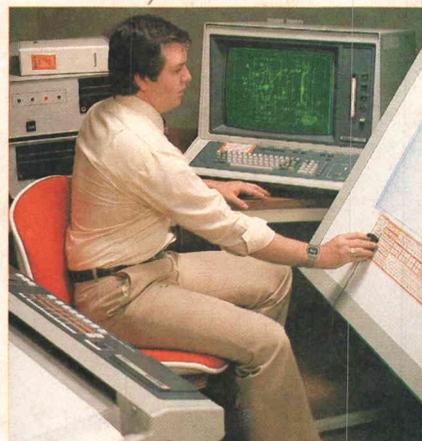
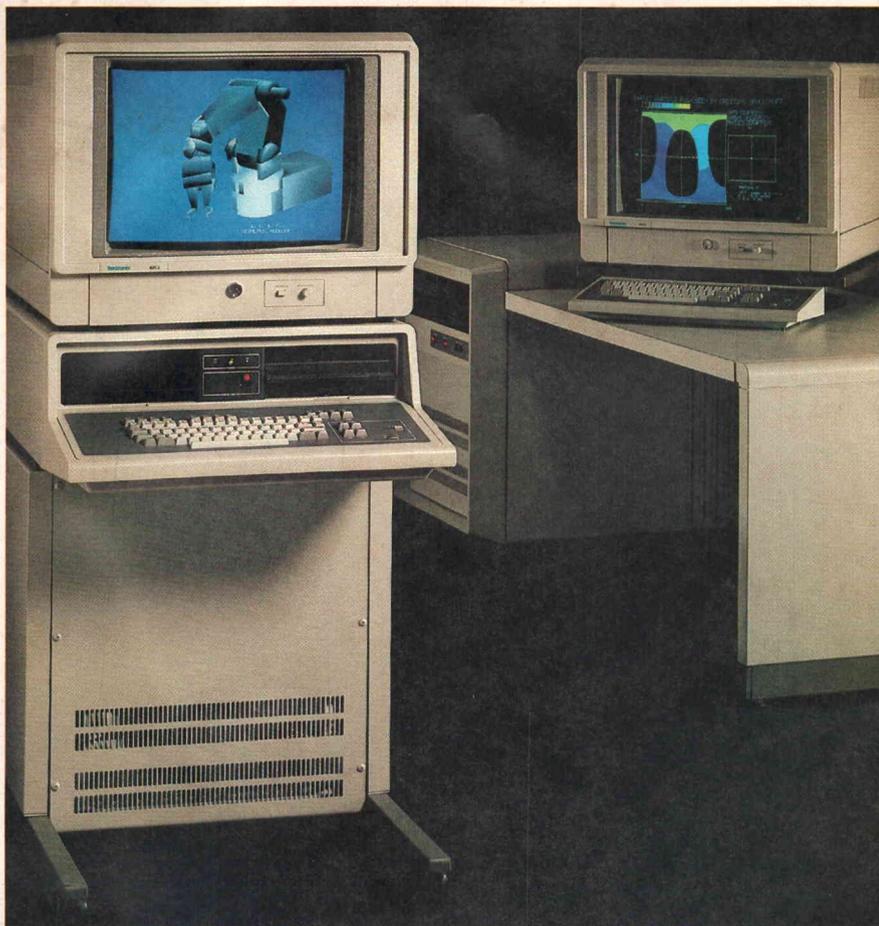


Tekniques



TEKniques Expands Its View

**PLOT 50
2-D Drafting: A Low
Cost Alternative**

SOLUTIONS

Software Information through SOLUTION VENDOR PROGRAM

Tekniques

In This Issue

<i>TEKniques</i> Expands Its View of Tektronix Graphics	2
Display Technology	4
4110 Series: An Evolution in Graphics Display	5
Ford Motor Slices Time from Lab to Designers Using 4052	10
Editor's Note	11
4050 Series Graphics Systems Workshops Scheduled	11
PLOT 50 2-D Drafting: High-Valued Features in a Low-Cost Package	12
Solution Vendor Program Informs Customers of Third Party Software	17
BASIC Language Enhancements Highlight New Models	18
Extended Memory File Manager Expands High Speed Memory in the 4052/54 Series	20
Built-In Diagnostics Enhance Extended Memory File Manager	20
Price/Performance Advantage in Tektronix 4663S Plotter	22
Plotter Manuals Updated	22
Programmable Spectrum Analyzers and 4050 Desktop Computers Automate Signal Monitoring	23
Input/Output	24
Programming Tips	25
BASIC Bits	30
New Abstracts	32
Program Updates	38
Library Addresses	40

TEKniques, the IDD Applications Newsletter, is published by the Information Display Division of Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, Oregon 97077. It is distributed to TEKTRONIX computer and terminal users.

Publishing Manager	Ken Cramer
Managing Editor	Patricia Kelley
Technical Editor	Dan Taylor
Graphic Design	John Ellis
Circulation	Rory Gugliotta
Typesetting	Jean Bunker

Copyright © 1982, Tektronix, Inc.
All rights reserved.

To submit articles to *TEKniques* or for information on reprinting articles, write to the above address. Changes of address should be sent to the IDD Library serving your area (see Library addresses).

TEKniques Expands Its View of Tektronix Graphics

TEKniques has been bringing news of applications, products, programming tips and other things of interest to 4050 graphics system users. With this issue, however, it expands its scope. In addition to the programmable 4050 Series, future *TEKniques* will include information about Tektronix graphics terminals.

To provide a better understanding of the variety of graphics terminals Tektronix produces and their respective applications, let's take a look at the graphics display history of Tektronix.

In The Beginning

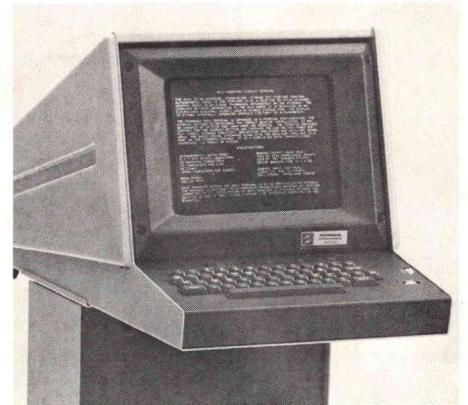
From the day (pre-1960) Tektronix produced its first 5-inch direct view storage tube (DVST) oscilloscope display, it has been offering graphics displays to the scientific and engineering communities. But Tektronix was primarily a manufacturer of test and measurement instruments and the initial DVST was mainly used for waveform displays which didn't require a fine writing beam or large screen.

After several years of engineering work, Tek produced an 11-inch DVST optimized for information display. It could write uniformly over the entire screen, could write small characters, and could display pictures. Unlike other computer graphics displays at that time, the Tektronix 611 didn't need refreshing; it could store a lot of information and remain flicker-free at a substantial savings in data transmission costs.

Building on the 611 display technology, Tektronix introduced its first DVST computer terminal in 1968. Called the T4002, it was the only graphics display of any consequence selling for under \$10,000. The 4002A followed closely on its heels.

4010 Series

Computer graphics was just emerging as a practical technique and the small, crisp writing beam and high resolution of the Tektronix DVST made it superior for graphics displays. When the 4010 terminal was introduced in 1971, with about twice the performance and at half the cost of the 4002, Tektronix was squarely in the information display business. A new business entity was born to forward the display technology: the Information Display Division (IDD) of Tektronix.



TEKTRONIX 4010 11" DVST Display Terminal.

The introduction of the 19-inch 4014 and 4015 terminals provided DVST graphics display devices with unequalled resolution. Local intelligence was offered along with plug-in firmware enhancements and other features users desired. These terminals provided the mapping, design, process layout and other diverse disciplines with the display capabilities they required. The 4014 soon became the standard in the industry for graphics display.

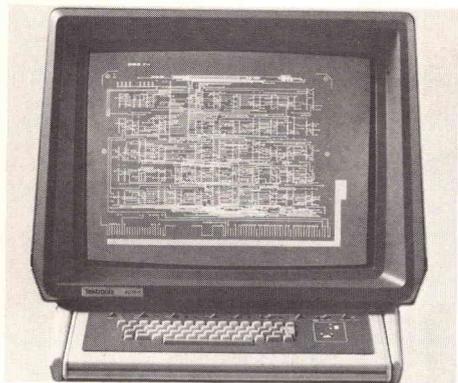


TEKTRONIX 4014 19" DVST Display Terminal with Optional Refresh.

The large screen 4016 (25-inch with more than 12 million displayable points) completed the Tektronix 4010 Series of DVST graphics display terminals. Designers of electronic circuit boards, utility networks, automotive components, street maps and similar applications who needed to work with fine detail while maintaining the total picture perspective chose the 4016.

4020 Series

With the advent of microprocessors and low cost memory, another form of display — raster scan (video) — became viable. Using



TEKTRONIX 4016 25" DVST Display Terminal with Optional Refresh.

this technology the Tektronix 4025 provided fast, straightforward interaction with the host computer and maximum efficiency in the manipulation of alphanumeric and graphics in a raster scan terminal.

Color graphics merged with alphanumeric to produce the 4027. Mapping, graphic arts, process-control and other applications which were enhanced by color found the display quality and local firmware capability of the 4027 fit their requirements.

Aiming to improve and enhance existing products, the 4020 Series was recently upgraded with the 4025A and 4027A to provide faster display and higher sustained data rate.



TEKTRONIX 4027A 11" Color Raster Scan Display Terminal.

4100 Series

In 1981 the 4110 Series of terminals was unwrapped which included both raster scan display and DVST display, in the 4112 and 4114 respectively. Local graphics functions in both terminals improved interactivity and significantly reduced host computer overhead. But the 4110 Series maintained compatibility with the 4010 family of terminals and the software developed for them.

Slipping into the line just before the end of the year, the 4113 brought all the features

and options of the 4112 plus the added dimension of high quality color.

The 4110 family commonality allows interdependent applications to share a host computer data base and user interface while using the correct graphics capabilities. The DVST 4114, for example, can display the most intricate three-dimensional, cross-sectional model, while its raster scan companion, the 4112, can use up to eight shades of gray for analyzing properties such as temperature or pressure variations. Combining the extra dimension of color with raster scan in the 4113 delivers the display clarity demanded for analyzing stress contour plots or designing multilayer circuit boards, buildings, or ships.

The power of the 4110 Series is beginning to make its impact. *TEKniques* will keep you informed of the unique and creative ways in which users employ these graphics terminals to solve problems.

PLOT 10™

So reams of alphanumeric computer print-out could be converted to graphics displays in the form of graphs, charts, isometric drawings and other graphic representations, PLOT 10™ ANSI Standard FORTRAN software was developed in 1971.

The modular design of subroutines in PLOT 10™ enabled users to start with only the code needed, then expand as applications become more sophisticated or specialized.

Over the years the PLOT 10™ Terminal Control System (TCS) became an industry standard for displaying graphics on DVST terminals. Built around TCS, PLOT 10™ has grown to include several software products including the Interactive Graphics Library (IGL) package. This is a modular system of I/O, device drivers, primary commands and advanced feature support that lets users move at will among Tektronix display devices or technologies. In 1978 PLOT 10™ IGL became the first commercially available package supporting the ANSI/SIGGRAPH "Core Graphics Standard."

4050 Series

Alongside developing its graphics display terminals, the Information Display Division married computing power with its DVST display technology and in 1975 produced the 4051 Desktop Computer. It was an instant success. And its programmability created a demand for a users' library. Thus, accompanying the 4051 was the 4050 Applications Library to help 4050 users exchange programs. *TEKniques* was established as the periodical of the Applications Library.

In 1979 the faster 4052 and larger screen, refreshed 4054 expanded the 4050 Series line. A bit later color was added to enhance the refreshed graphics of the 4054.

Just recently the 4052A and 4054A have brought more technological developments to 4050 users. And the expanded memory provides state-of-the-art storage and retrieval for these desktop computers.



TEKTRONIX 19" DVST 4054 with Optional Color-Enhanced Refresh

PLOT 50

Accompanying the 4050 Series has been a set of software supporting the graphics capability of the 4050 systems. PLOT 50 continued the methodology established by PLOT 10™ for fundamental computer graphics commands which meant users familiar with the FORTRAN PLOT 10™ TCS software could easily understand PLOT 50 graphics terminology.

Written in BASIC, PLOT 50 products included utilities, statistics, mathematics, graphing and business planning and analysis packages to interpret raw data. Recently PLOT 50 digitizing, picture composition and drafting packages have been added to aid engineering, mapping, design and other graphics analysis disciplines.

Solution Software

Rounding out the software support is the Solution Vendor Program providing application-specific software solutions to our customers.

Just as one individual can't hope to excel in all things, one company can't hope to solve all problems. Tektronix, therefore, believed that for certain applications it made sense to combine Tektronix Graphics with software from other sources. Consequently the Solution Vendor Program brings the expertise of others together with Tek's graphics display technology to provide solutions for certain customer application problems.

The Big Picture

With this very brief sketch of IDD graphics

over the last 15 years, you can see that the 4050 Series has plenty of company. *TEKniques* has kept 4050 users informed of 4050 applications, products and peripherals. But to give our readers a broader view of com-

puter graphics in engineering, science and related disciplines, *TEKniques* will be covering applications of other IDD products as well as the 4050 Series.

The Applications Library will also be expanded to include user developed software for other Tektronix terminals.

Display Technologies

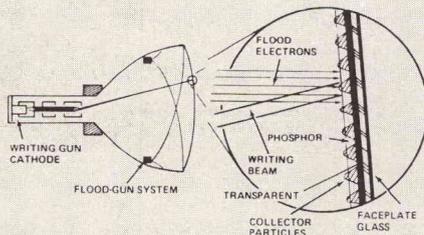
Computer graphics displays can best be separated into two definitive categories — those which use an electron beam, and those which do not, for instance, plasma displays. Displays using an electron beam are called cathode ray tubes and it is this technology discussed in the following paragraphs.

Cathode ray tube technology can be divided into the direct view storage tube which doesn't require local display memory to store an image, and the refreshed tube which requires information to be stored in some type of local display memory. The refreshed tube technology is further subdivided by vector generation, whether it's done by raster scan or calligraphic display.

At Tektronix the direct view storage tube and the raster scan refreshed tube have been the mainstays of our graphics display terminals and desktop computers.

Direct View Storage Tube

The direct view storage tube (DVST) consists of a single electron writing beam, a phosphor-coated display surface, silicon targets mounted just behind the phosphor coating, and special flood guns.



The writing gun of the cathode ray tube strokes out or traces an image on the silicon targets. The high energy of the writing beam is passed through the positively charged silicon target onto the display phosphor. Once the image is etched into the target area, the flood guns mounted on the rear of the cathode ray tube emit a low energy flood of electrons to maintain the image on the display surface. Because of the energy from the flood guns, there is no need to continually redraw the image. The graphics are essentially stored on the phosphor of the screen.

Pictures displayed on the direct view storage tube can be highly complex and detailed. All lines are continuous and crisp which results in superb resolution. Because flood guns maintain the image, the continuous retracing of an image doesn't occur and flicker isn't present. Since the DVST doesn't require local display processing and memory to continually refresh the picture, display system cost is lower.

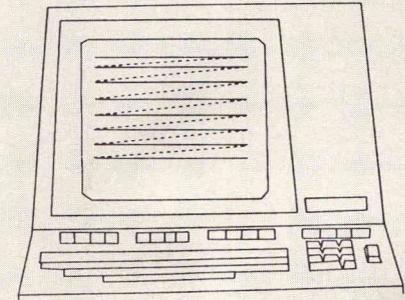
In more sophisticated (and more expensive) systems, a modified refresh capability overcomes the inherent inability of the DVST to selectively erase, i.e., erase a portion of the display without having to redraw the entire picture. A separate local display memory allows objects to be created and repeatedly retraced producing a nonstored object on the screen. The object can be deleted from the screen, or stored on the screen while remaining in display memory to be called up again. This feature allows objects to be moved around the screen before storing, or allows menus or prompts to be periodically displayed without storing. Since this memory is essentially independent of the circuitry required to produce a stored image, both stored and nonstored information can be displayed simultaneously.

The superior resolution, high information density and relative low cost make the direct view storage tube ideal for complex line drawing applications.

Refreshed Raster Scan Tube

Raster scan technology consists of a single electron writing beam and a short persistence phosphor coating the display surface which must continually be excited or else the image will decay and cease to glow. Therefore, a writing beam sweeps across the screen surface constantly giving energy to (or refreshing) selected phosphor dots (picture elements or pixels). The writing beam sweeps the display surface from left to right, top to bottom, up to 60 times per second.

The interlaced or noninterlaced method may be selected for scanning the display surface. An interlaced display will scan every other line of the display (1, 3, 5 . . . etc.) in the first 60th of a second, then



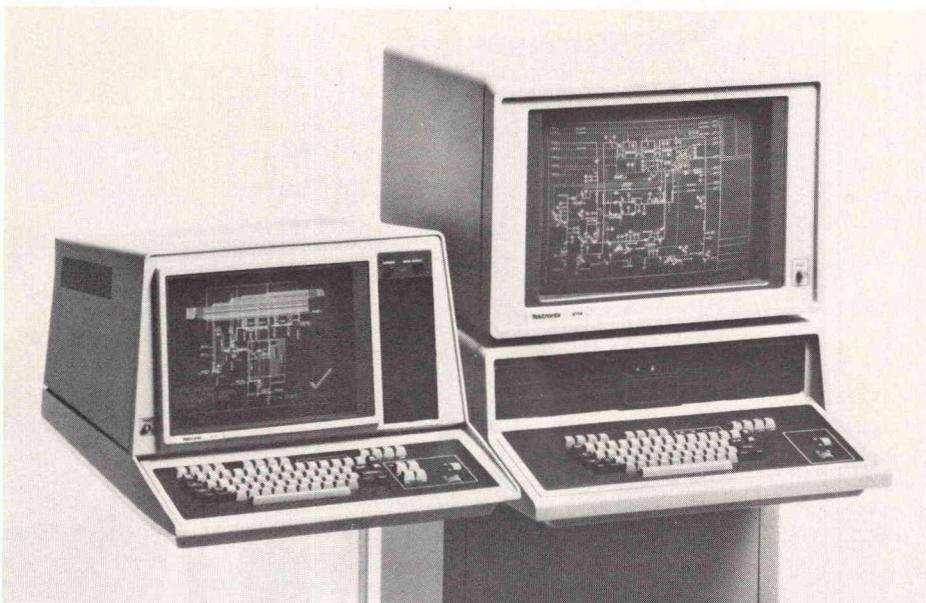
return to the top of the screen and scan the intermediate lines (2, 4, 6 . . . etc.) in the next 60th of a second. Interlaced displays require a longer persistence phosphor which also reduces flicker. The noninterlaced display is just the opposite of the interlaced, scanning every line every 60th of a second, and results in a very bright display.

A computer cannot supply enough information fast enough to continually paint an entire complex picture on a display surface 60 times per second. To compensate, the raster scan terminal contains an internal memory plane (more than one if it's a color terminal) which retains the picture information sent from the computer. The terminal reads the information stored and causes the writing beam to scan sequentially from left to right, top to bottom, turning on only the selected pixels.

Because a picture can be updated between scans, raster scan technology accommodates selective erase. And, as with a picture in a magazine or newspaper which is made up of many dots, shading and dot patterns can be achieved by varying dot densities to create a picture that resembles a photograph with continuous toning.

Because the raster scan area is fixed and lines will consist of individual dots, raster scan display devices serve as excellent graphics terminals where high resolution is not a requirement. Architectural modeling, molecular modeling, process control, LANDSAT pictures, imagery and alphanumeric forms fill-out are just a few examples of applications which are efficient users of raster scan display technology. 

4110 Series: An Evolution in Graphics Display



Because the new TEKTRONIX 4110 Series is based on a range of needs rather than on a single technology, it provides users with a choice: the high dynamics monochrome raster display in the 4112 (left), the high-resolution storage tube with local refresh capabilities in the 4114 (right), or the superb clarity of raster color graphics in the 4113 (shown on the cover).

An extension of the Tektronix 4010 family of terminals, the 4110 Series of host-based terminals combine the power of host computing with the flexibility of local intelligence and interactivity in displays of DVST, monochrome or color raster scan (see Display Technology article). They are compatible with the 4010 members and support programs developed for the 4010 Series.

Each 4110 Series terminal has certain unique strengths supported by a powerful set of common capabilities.

Local Interactivity

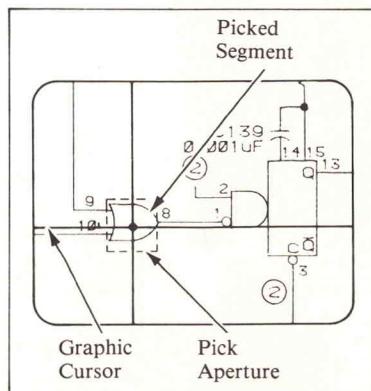
All 4110 terminals are designed for friendly local interaction. Keyboard keys provide access to frequently used terminal features. English-like commands set the operating environment, inquire about the status, drive peripherals, and program "macros" into keys. Commands may be typed in at the terminal keyboard and be executed as if coming from the host.

Graphic Interactivity

Graphic input (GIN) provides a means for the operator at the 4110 terminal to interact with a graphics application program running on the host. But local processing power and memory allow 4110 terminals to shoulder more of the work that was formerly performed by the host computer.

In addition to notifying the host of the location of the graphics cursor in picture space, 4110 terminals can also identify the picture segment (see Segments) picked and which part of the segment was picked. A segment may also replace the thumbwheel crosshairs as the GIN cursor. When using the graphics tablet cursor, the terminals can send the host a stream of continuous reports on the cursor's location.

A variety of features enhance 4110 GIN functions including rubberbanding, inking, gridding, user-defined GIN cursors and multiple GIN reports to the host.

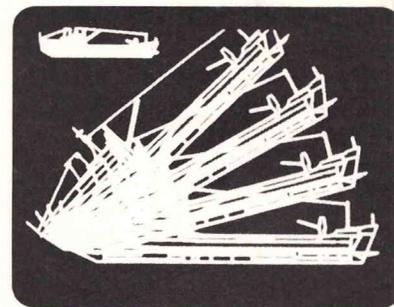


The 4110 terminals can identify to the host which picture segment was picked and which part of the segment.

Segments

A major contribution of the 4110 family to faster, more efficient graphics is local picture segments. That is, the graphics data defining a portion of a picture may be sent from the host and stored in the 4110 Series terminal as a segment under an assigned number. Thereafter, when the host sends instructions for a picture, it can recall each segment independently by its number from local storage, and doesn't have to retransmit the entire sequence of graphics data for the segment.

2D image transformations allow the user to modify the image (how the segment is displayed on the screen) of each segment. A segment can thus be repositioned on the screen (translated), scaled up or down, or rotated. When the host transforms a segment, it need send only a few transformation commands, rather than retransmitting the data defining each segment. Transformations may also be done locally by the terminal operator.



The ship shown on the upper left corner of the 4112 display is stored as a local picture segment in the 4110 terminal's memory. It is then translated to the lower half of the screen, scaled to enlarge the image, then rotated three times in the counterclockwise direction.

Segments can be grouped into one or more classes and a selected class highlighted, erased, or transformed without the host having to reference each segment involved.

Additional commands make a segment visible or invisible, change its writing mode, or change its detectability for graphic input operations.

A locally defined segment can be copied into other segments. Segments displayed in refresh (always the case in the 4112 and 4113) can be selectively erased — removed

from the screen without erasing the entire display. A segment can be made to blink on and off, highlighting it.

When the 4110 Series terminal is equipped with the optional flexible disk drives, locally retained picture segments can be saved to disk. This allows the host to command the 4110 to load the segment into memory directly from disk which is faster and less costly than retransmitting all the graphics data from the host.

User-Defined Fonts

Storing user-defined fonts in memory (and optionally disk) further decreases the amount of data that must be transmitted to 4110 terminals by the host. When a character in a user-defined font is subsequently referenced, the vectors which define the character are drawn from the 4110 terminal's memory rather than having to be retransmitted by the host.

Programmable Keys

Most of a 4110 terminal's keys are programmable. This means a sequence of host or terminal commands (called a macro) can be assigned to one of the keyboard keys. When that key is pressed, or when the host calls for the macro, the sequence is called from 4110 memory rather than being sent from the host. Macro definitions can also be stored locally on disk, then loaded by a single command from the host.

Dialog Area

The display of nongraphics, such as terminal-host messages, may be restricted to a specific portion of the 4110 terminal's screen. When



A specific portion of the 4110 terminal's screen may be designated as the dialog area, and repositioned at any time.

messages fill the 4110's dialog area, the text is automatically scrolled. As a line of text is scrolled up and out of the dialog area, it's stored locally in memory. By using the thumbwheels, the operator may scroll back to reference previously displayed text. This prevents writing over picture information or having to erase the screen (and the picture) to capture a message. The dialog area may be turned off allowing messages to come through and to be stored in memory for accessing later without cluttering the screen. It may also be repositioned or redefined at any time.

Data Communications

For all 4110 terminals, an optional block mode of communication is available. This enables hosts to use compact data formats even if those hosts cannot normally send or receive control characters, lowercase, etc. Full eight-bit binary files can be sent and received. Error checking is provided even with hosts that do not provide parity bits. And long strings of data such as graphic input strokes, can be entered without regard for overrunning the host input buffer.

Memory and Storage

Over 700K bytes of memory are available for the 4110 terminals to support the local capabilities. Single and dual integral flexible disk drives with 494K bytes capacity each may be specified for local storage of segments, fonts, macros or completed graphics displays.

4112 and 4113 Display Features

In addition to the capabilities just discussed, the 15-inch monochrome 4112 and 19-inch color 4113 terminals provide graphics tools unique to their raster-scan technology displays.

Surfaces, Bit Planes, Gray Levels and Colors

The 4112 user may choose from a palette of 15 levels (shades) of gray, the 4113 user from a palette of 4096 colors. How these shades or colors may be displayed in a picture depends on the writing surfaces used and those surfaces' bit planes.

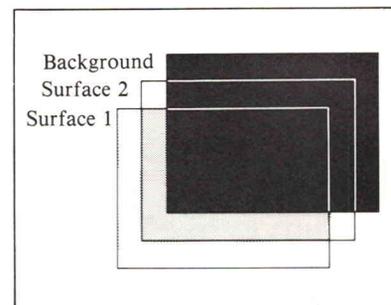
The 4112 may have one to three writing surfaces and bit planes, the 4113 from one to four.

A writing surface is a logical entity separate from the physical display screen. It might be thought of as a piece of transparent mylar covered with tiny transparent dots (called pixels) laid out in a grid of 480 rows, 640 to a row. All the transparent surfaces are overlaid on a common background.

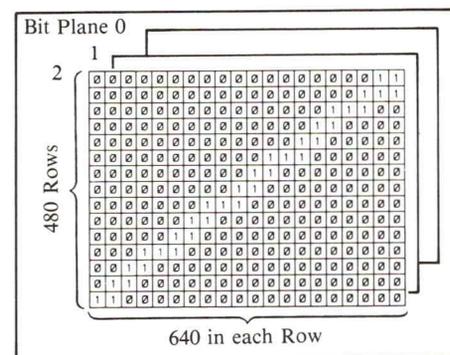
Each bit plane may be considered as a grid

(480 by 640) of switches corresponding to the pixels in the surfaces. Turning on a switch in a bit plane assigns a gray level or color to its corresponding pixel in a surface.

A user can employ one or all of the surfaces for displaying a drawing. For a segment drawn on a surface to show, however, one or more bit planes must be assigned to that surface so the appropriate switches in the bit plane(s) can be turned on to assign a gray level or color to the segment's pixels.



In this example, surface 1 is displayed in front of surface 2. Behind the surfaces is a background of "0% lightness" (black).



The number of bit planes assigned to a writing surface determine how many gray levels or colors can be displayed on that surface.

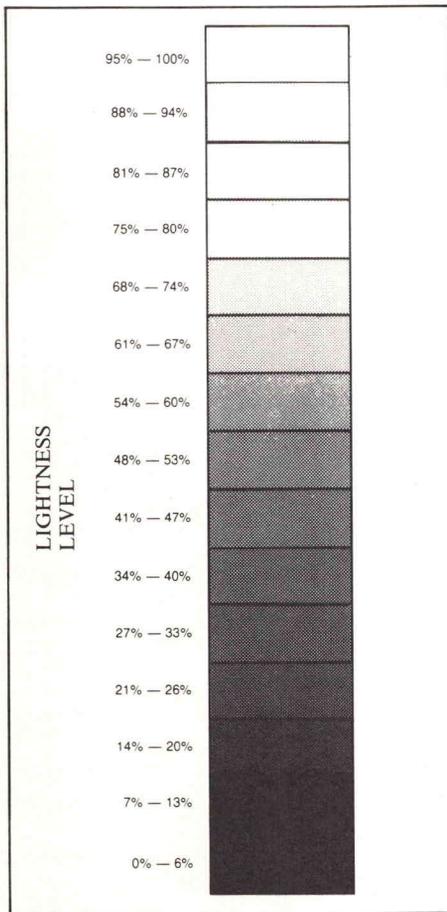
How Bit Planes Assign Gray Levels or Colors to Pixels

All switches turned on in a single bit plane point to (or index) the same gray level or color which is to be assigned to their corresponding surface pixels. Switches turned on in another bit plane can index a different gray level or color. But two bit planes working together can index a third color by turning on switches from both planes for the same surface pixel.

Thus, with one bit plane assigned to a surface, two gray levels or colors can be displayed — the bit plane's indexed color and the background. To illustrate, 1) where a bit plane switch is off, the corresponding pixel on the surface remains transparent and the background gray level or color shows through; 2) where a bit plane switch is on, the pixel is assigned the gray level or color indexed by that bit plane's switches.

When two bit planes are stacked together and assigned to one writing surface, three shades or colors may be displayed plus the background: 1) where both bit plane switches for a pixel are off, no color is assigned leaving the background to show through; 2) where a switch in only the first bit plane is on, the gray level or color indexed by that switch is assigned to the pixel; 3) where a switch in only the second bit plane is on, the gray level or color indexed by that switch is assigned to the pixel; 4) but when switches from both planes are on, the gray level or color indexed by both would be assigned to the pixel.

Three bit planes assigned to one writing surface could provide seven shades or colors plus the background; four bit planes assigned to a surface could provide 15 gray levels or colors plus the background.

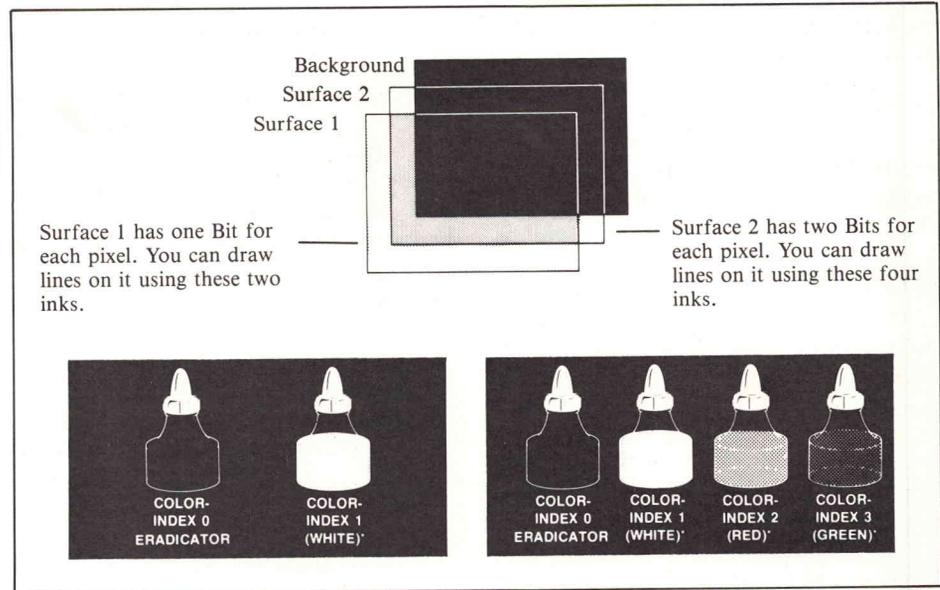
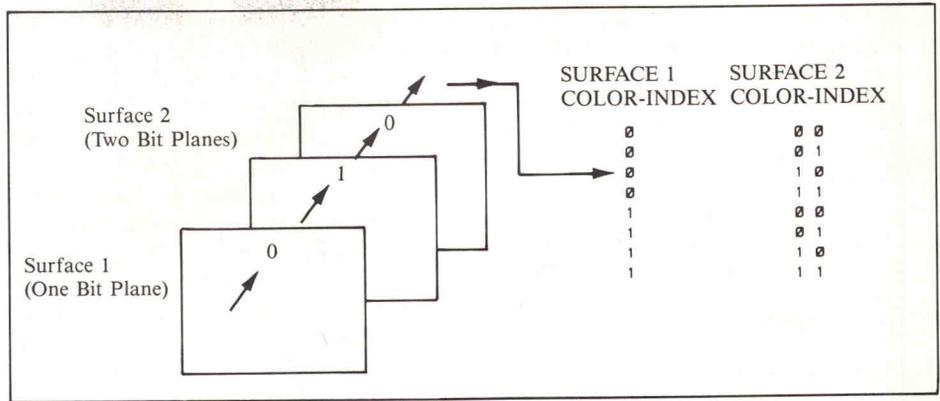


A user may choose from 15 gray levels.

With one bit plane assigned to surface one, a pixel can display one gray level or color when its bit plane switch is turned on (index 1) or let the background show through (index 0).

With two bit planes, surface two can display three colors plus background:

Bit Plane 2	Bit Plane 1	Index
off	off	0
off	on	1
on	off	2
on	on	3



Think of the indices as ink bottles into which inks of different colors or gray levels may be loaded. Colors are arbitrary except for index 0. Surface pixels set to color index 0 allow the surface behind it to show.

Choosing Gray Levels or Colors

The gray levels or colors in the "ink bottles" indexed by the bit planes are easily changed. The background can be assigned any one of the gray levels or colors available, and each ink bottle (except 0) can be assigned any gray level or color.

To change gray levels or colors, the user simply reassigns gray levels or colors to a surface's "ink bottles" (indices).

Assigned gray levels may range from a lightness level of 0% (black) to 100% (white). Colors are selected from a universe of 4096 possible colors.

Displaying Surfaces

Thus, a picture requiring many colors or shades could be drawn on one surface with the maximum number of bit planes assigned to that surface. The rest of the surfaces would remain undefined (and transparent).

An alternative method of display is to use several writing surfaces. If each of the bit planes is assigned to a different surface, the

surfaces can be viewed separately, or overlaid and viewed as an entity. Pixels in the surface(s) whose bit plane switches are off are transparent and allow the subsequent surface(s) to show through. Pixels on surfaces whose bit plane switches are on can obscure subsequent surfaces, let them show through, or combine with them for different shades or hues (user-defined).

Any surface can be turned on or off, can be highlighted (blinked), or can be reorganized by viewing priority in any order. Thus it would be easy and fast to look at only the level of the drawing desired or to bring the bottom writing surface forward as the first overlay.

This method of display would be handy for separating a drawing such as a multilayered circuit board and displaying individual levels or viewing it as a complete picture.

A surface can also be defined and have picture segments assigned to it without having any bit planes assigned to it. When a bit plane (or planes) is assigned to that surface,

it immediately becomes visible; any surface whose bit planes are removed would immediately become transparent. By transferring bit planes among defined surfaces, even greater manipulative power is available. But, of course, the number of colors per surface is restricted because fewer bit planes can be assigned to a surface.

Panels and Fill Patterns

The 4112 and 4113 terminals can draw panels (closed regions) and fill the interior of the panels with fill patterns. A fill pattern can be a gray shade or a color (4113), one of 16 pre-defined patterns, or a pattern defined by the user. The fill pattern may include the panel's boundary, or the boundary may be distinct. Panels may be included as part of a segment.

Zoom and Pan

A picture is defined in the 4110 terminal's 4096 by 4096 addressable space. The 4112 and 4113 displays, however, are comprised of 640 by 480 pixels. Consequently, the screen of the terminal doesn't display the full resolution of its picture space. Zoom and pan compensate for the difference in resolution.

The 4110 zoom is a true zoom. That is, it doesn't just expand a drawing portion, but actually displays more discrete points in the picture as it zooms in for a closer look. This allows a complete picture entity to be defined using the larger picture space and to be zoomed in to display more detail rather than having to define it in magnified segments to capture the detail.

Pan locates the area in the picture space to be displayed. Pan allows the operator to view adjacent areas by moving through and around the picture space.

Both zoom and pan are controlled locally from the thumbwheels and keyboard viewing keys.

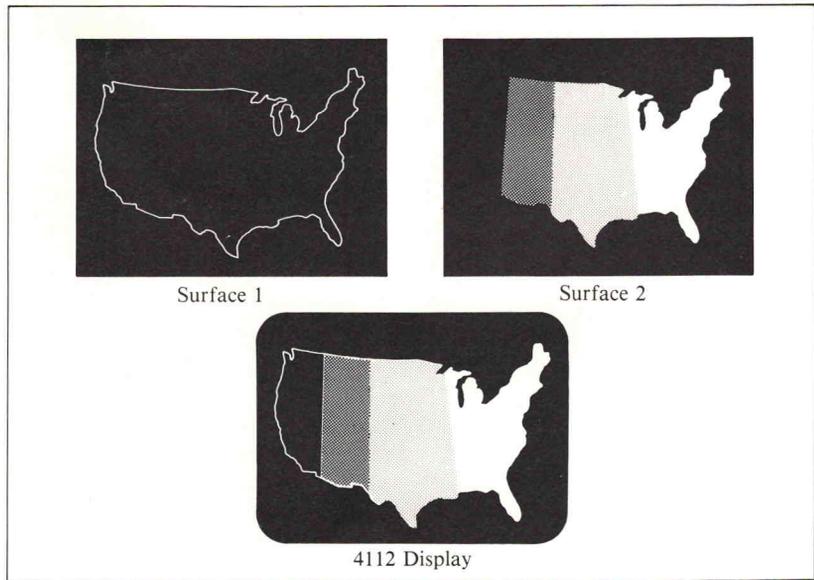
Multiple Views

Up to 64 separate views can be defined and displayed. A view has:

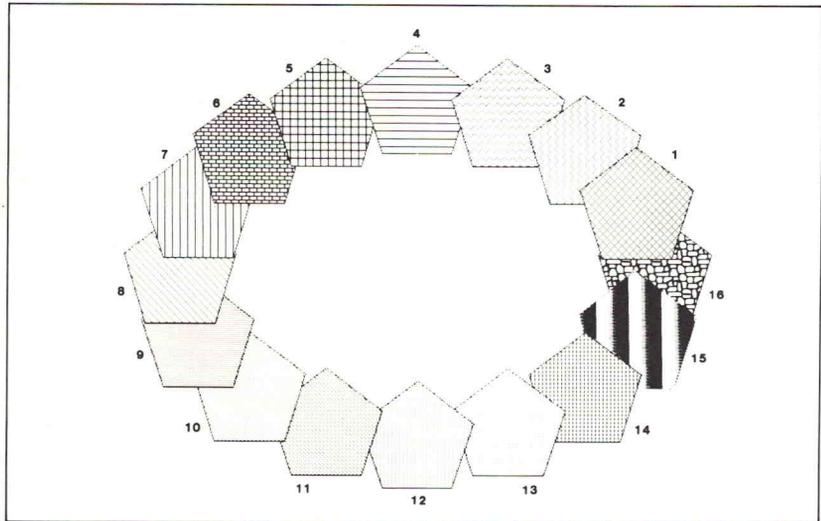
- a view number
- a viewport definition
- a window definition
- a surface number
- a border

Although many views may be displayed at one time, only one view is current at any given time, that is, only one view can be operated on at a time (zoom, pan and so on).

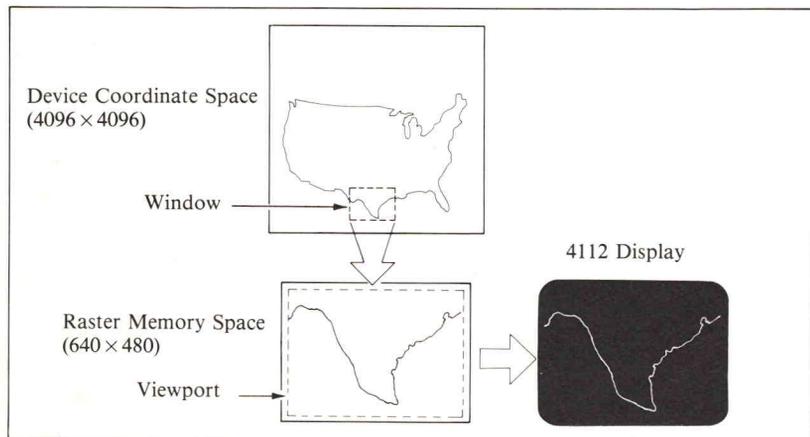
Each view is assigned a number. The window determines the rectangular area in the



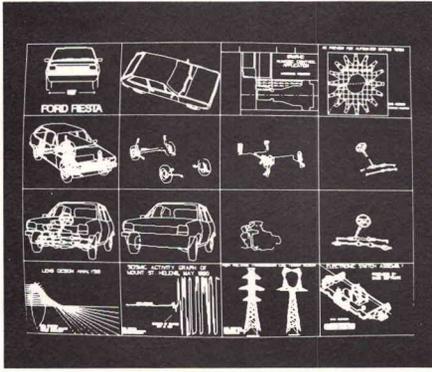
Surface 1 has one bit plane assigned to it, and the borders of the map are drawn in index-1 color. Surface 2 has two bit planes assigned to it allowing three gray levels plus the background. Overlaying the two surfaces against the background produces a map of five gray levels.



Fill patterns may be chosen from gray levels, solid colors, 16 predefined patterns or may be user-defined.



During zoom and pan operations, the 4112 and 4113 actually display more discrete points from device coordinate space as they zoom in for a closer look.



Dividing this 4112 display into 16 viewports facilitates a side-by-side comparison of related graphics information.

terminal's 4096×4096 space which will be in that view. A viewport specifies the rectangle in 640 by 480 display space where the window contents appear. The viewport exists on one of the writing surfaces. A picture in a view is defined by the segments which are visible when that view is current.

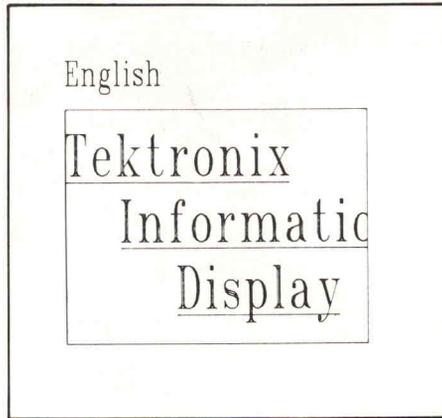
Gray level or color indices assigned to segment primitives (lines, text, etc.), fill patterns, text, and dialog area correspond directly to those assigned to the surface on which they are displayed.

A toggle key makes the border around the current view's viewport visible or invisible.

More of the concepts and capabilities of the 4112 and 4113 terminals will be reviewed in future issues of *TEKniques*.

4114 Display Features

Coupled with the local intelligence common to all 4110 terminals, the 19-inch 4114 DVST display meets current and evolving needs in high density graphics. The 4114's 4096 by 4096 addressable points with 4096 by 3120 displayable provide resolution high enough for the most complex engineering and scientific graphics.



Stroke-generated characters on the 4114 are producible in 16 sizes for greater variety of titles, text and labels.

Refresh

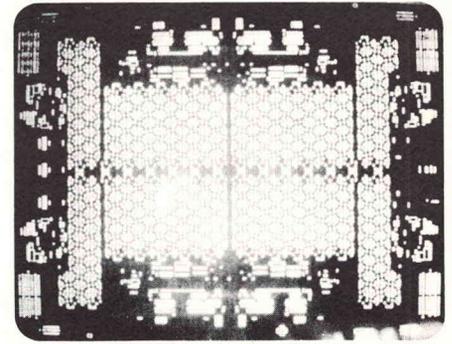
The refresh capability of the 4114 allows more than 1500 centimeters (approximately 3000 short vectors) to be displayed in refresh mode. The size of refresh memory permits many segments to be displayed in refresh for powerful user interaction. Refreshed segments may also be highlighted (blinked).

In addition, characters in the user-definable dialog area are shown in scrolling refresh, eliminating clutter from the drawing area.

An optional color enhanced refresh displays all refreshed vectors in an amber color. The clear contrast with stored vectors is ideal for high density applications and those with a great deal of refresh manipulation.

FAST Redraw

Because displays are stored in local memory, redrawing of the picture is not dependent upon data communications baud rate. And the redraw time of the 4114 is extremely fast. For example, 26,000 short vectors may be redrawn from 4114 memory in one-half second.



Once stored in the 4114's memory, this integrated circuit mask containing 26,000 short vectors can be redisplayed on the screen in less than half a second.

Speedy Communications

Graphics data bases are usually quite large. Higher data communications rates in the 4114 up to 19,200 baud allow information to be transmitted at rates that will move a lot of information in a short time.

The 4110 Series expands and extends graphics terminals into new levels of interactivity and productivity. By having intelligence and speed in the terminal, Tektronix 4110 Series terminals reduce communications traffic and increase data communications rates resulting in faster displays and more interactive programs. 

Ford Motor Slices Time from Lab to Designers Using 4052

by Patricia Kelley
TEKniques Staff

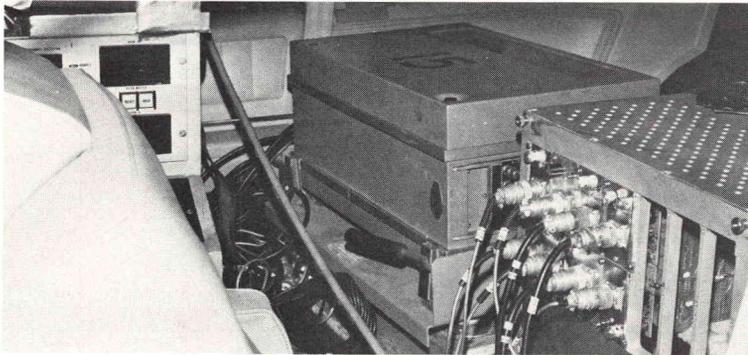
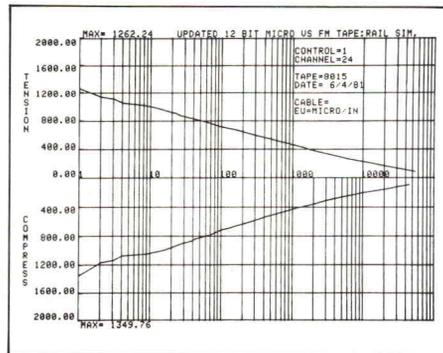
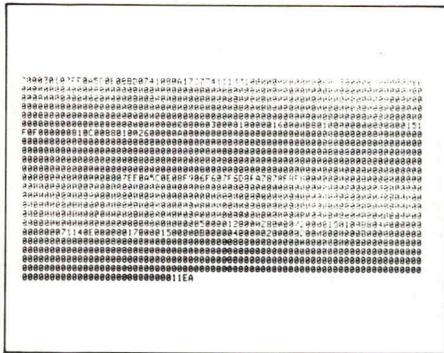


Figure 1. A 24-channel data acquisition unit picks up incoming signals from sensors attached to the vehicle's components. These are converted, analyzed and plotted for the design engineers at Ford Motor Company.



Figures 2a and 2b. Statistically reduced and analyzed data (figure 2a) presented graphically (figure 2b) gives Ford's design engineers a picture of component reaction to fatigue tests.

Data captured from load cycle testing on a vehicle in Ford Motor Company's vehicle laboratory or on a test car driven over proving ground routes must be converted, formatted, previewed and plotted before design engineers can analyze the test outcome.

Wanting to speed the results from the laboratory to the hands of the design engineers, one of the stress testing units at Ford in Dearborn, MI, settled on the Tektronix 4052 Desktop Computer as the solution. It has served them well.

Jerry Blake, principal engineer at the lab, and Duncan Preece, research engineer, briefly explained their work.

Previously, data acquisition functions were accomplished by playing FM tape recordings of analog road test data into a host computer. Today, mobile microcomputer-based data acquisition systems handle these tasks locally. It's Jerry Blake's responsibility to implement the data acquisition requirements in a mobile system. Duncan Preece programs the 24-channel unit.

On a vehicle, sensors attached to the structure under study, e.g., cross member, suspension system, frame, and so forth, are connected to the mobile test unit (figure 1). Incoming data is processed through computer controlled signal conditioning, digitized, statistically formatted and stored in RAM memory. At the conclusion of each test, the data is transferred to a small built-in cassette recorder.

When test series (a number of individual tests) is completed, the cassette is played into the 4052 for immediate engineering unit conversion, analysis and plotting. At this point, the design engineers can have their data in listing or plot form. With this system, these initial plots can be generated at the rate of one set every 20 minutes. (A set contains the data and plots for 24 data channels.)

Lori Frank, programmer, writes the code for the 4052 to generate the immediate plots of the road load data, as well as to provide for the option of transferring the data to Ford's host computer. When the laboratory

engineers are satisfied with the presentation, they can command the 4052 to transfer the data to the host computer for file storage and high resolution CalComp plotting.

The comparison graphs produced by the 4052 system give the design engineers timely, easy-to-understand information on component reaction to fatigue tests. The graphics enable them to rapidly determine whether additional tests on that vehicle are needed. Or, if a design engineer wishes to see the data with different parameters, Lori can make code changes on the 4052 program and produce the desired plot immediately. This would be less practical on a host-based system.

The 4052 and its plotter also are taken to remote sites for on-location data reduction. This has represented a major efficiency for certain product development programs.

So successful was the first system that the testing laboratory staff packaged the sensors, testing boards, software and 4052 and shipped them to the Ford's Engineering and Research Center in Dunton, England. It's working very well and that group has extended the system's capability to run damage analysis routines — a program usually running on a host computer which can predict what a part will do under fatigue based on completed stress tests.

Blake commented that the versatility and high resolution of the 4052 was a major factor in their choice. He also credited the superior support of Tektronix Sales Engineers and Systems Analysts for a large part of the reason they have chosen Tektronix gear — including many oscilloscopes — over the years.



Figure 3. Jerry Blake, principal engineer at the testing lab, relates to Gregg Worth, Tektronix, (foreground) the nature of the stress tests.

Editor's Note: The state-of-the-art nature of this laboratory's work was impressive. Our observations indicated that there wasn't much spare time, so we are very grateful to Jerry, Lori and Duncan for interrupting their activities to visit with us. Greg Worth, Technical Support Manager at Tek's Detroit Office, arranged the interview and helped gather information.

* Editor's Note:

IDD Applications Library Name Denotes Expanded Role

You'll note the Applications Library has changed its name to reflect its expanded role. Programs for Tektronix IDD terminals will be included in the library in addition to the 4050 Desktop Computer programs. More information will be forthcoming in subsequent issues of *TEKniques*.

Back Issues and Reprints from *TEKniques*

TEKniques is in its sixth year of publication. Issues from the first three years (Volumes 1-3) have all been distributed. However, most of the articles from those issues have been assembled by application area and are available in the following reprints:

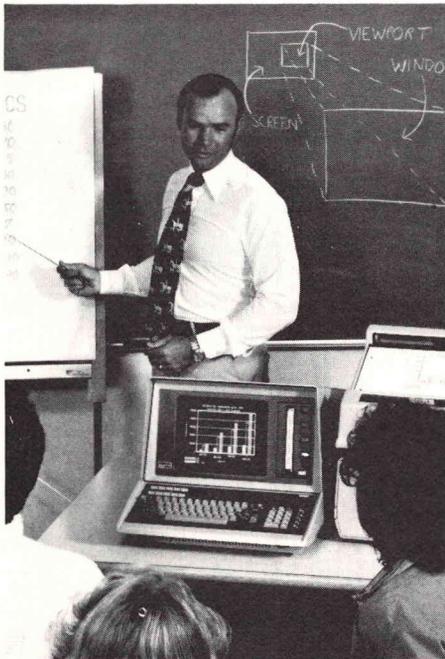
- Engineering and Design AX-4449
- Mapping AX-4460
- Data Acquisition and Analysis . . . AX-4450
- Business Graphing and Reporting AX-4451
- Peripherals and ROM Packs AX-4452

If you need an article from one of these previous volumes, and don't have your copy, one of the reprint sets will likely fill your needs. To obtain a copy of one of the reprint volumes, just contact your local Tektronix office or the IDD Applications Library office serving you.

And, of course, back issues of *TEKniques* Vol. 4 (1980), Vol. 5 (1981) and Vol. 6 (1982) will continue to be available from the IDD Applications Library office that serves your area.

Programming Tips Handbook

The programming tips from the first three years have been collected into a handy booklet which is included in the Programming Aids T2 tape documentation (part #062-5972-00). 



4050 Series Graphic Systems Workshops Scheduled

The 4050 Series Graphic Computing Systems workshops are designed to help you get the most out of your graphic system. The week-long workshops combine classroom lecture with extensive laboratory sessions.

The 1982-1983 schedule is:

Santa Clara, CA	Rockville, MD
September 13-17	September 20-24
October 11-15	October 18-22
November 1-5	November 15-19
December 6-10	December 6-10
January 3-7	January 10-14
January 31-February 4	February 7-11
February 28-March 4	March 7-11
March 21-25	April 4-8
April 18-22	May 2-6
May 2-6	May 23-27
May 23-27	

For additional information, contact Customer Training at (503) 642-8660. 

PLOT 50 2-D Drafting: High-Valued Features in a Low-Cost Package

by Jeff Carter
with John Harms
and John Tangney
Tektronix, Inc.
Wilsonville, OR

Tektronix has recently introduced PLOT 50 2-D Drafting, a cost-effective alternative to high-priced automated drafting systems. Planned for office and drafting room operation, the 4054 Desktop Computer-based system doesn't require special electrical or environmental considerations, unlike previous large systems. Nor does it require expensive connect time to a host computer. It's a standalone, two-dimensional drafting system for electrical, mechanical or other engineering disciplines. Drawings from two of these disciplines are shown in figures 1 and 2.

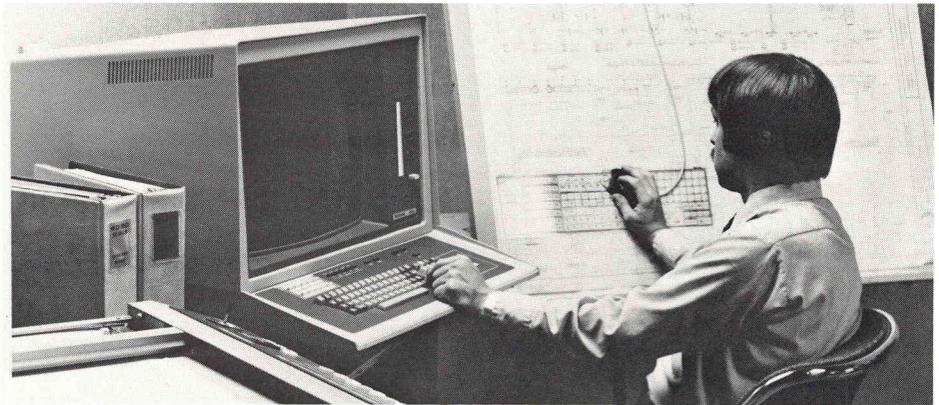
PLOT 50 2-D Drafting makes short work of some of the slowest and least productive manual drafting tasks, like revisions. It also permits drawings to be manipulated in ways that would be difficult and time consuming with hand drafting methods, and expensive with previous automated systems.

Automating the drafting tasks, however, doesn't require sacrificing familiar methods or flexibility. PLOT 50 2-D Drafting allows existing company drafting standards and practices to be maintained while adhering to national (ANSI Y14) and many international (ISO) drafting standards. Furthermore, it communicates in the language of the drafter and proceeds in customary drafting steps; therefore, computer knowledge isn't necessary.

PLOT 50 2-D Drafting is a general purpose drafting system that helps eliminate drafting bottlenecks to increase drawing output at a reasonable cost.

An Intelligent Tool Communicates Effectively

Before gains in output can be achieved with an automated system, the user must be able to interact effectively with it. PLOT 50 2-D Drafting methods comfortably approximate manual methods. Drawings are created and revised on the 19" screen of the 4054 Desktop Computer which replaces the sheet of paper and drafting table. Program operation centers around a tablet menu of familiar drafting functions (shown in figure 3). The drafter selects a function from the



The TEKTRONIX 4054-based PLOT 50 2-D Drafting system is a cost-effective alternative to high priced drafting systems. Cutting across disciplines, PLOT 50 2-D Drafting is a general purpose drafting system for a variety of engineering fields.

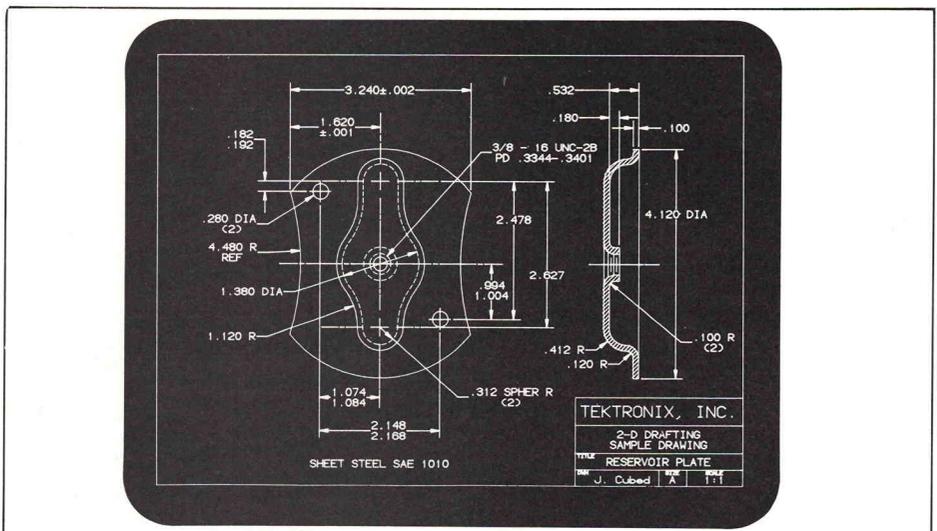


Figure 1. A drafter has a wide choice of dimensioning styles, line styles, and shading with PLOT 50 2-D Drafting.

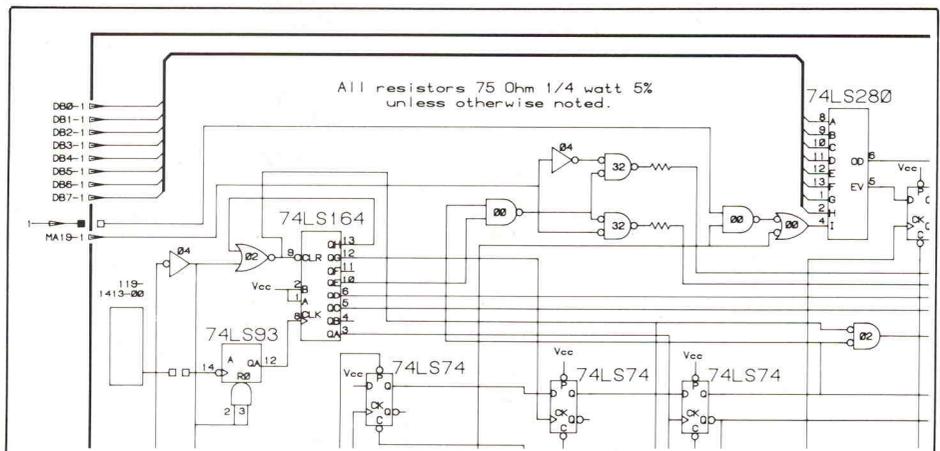


Figure 2. PLOT 50 2-D Drafting eliminates repetitive manual drawing. Symbols or portions of a drawing can be created once then automatically copied to another portion of the drawing.

menu, then responds to simple prompts or questions displayed on the 4054 screen (figure 4). Users don't have to learn a special set of terms to invoke a command, nor are

they restricted to a menu hierarchy.

For example, to manually draw a circle, the drafter would reach for the appropriate

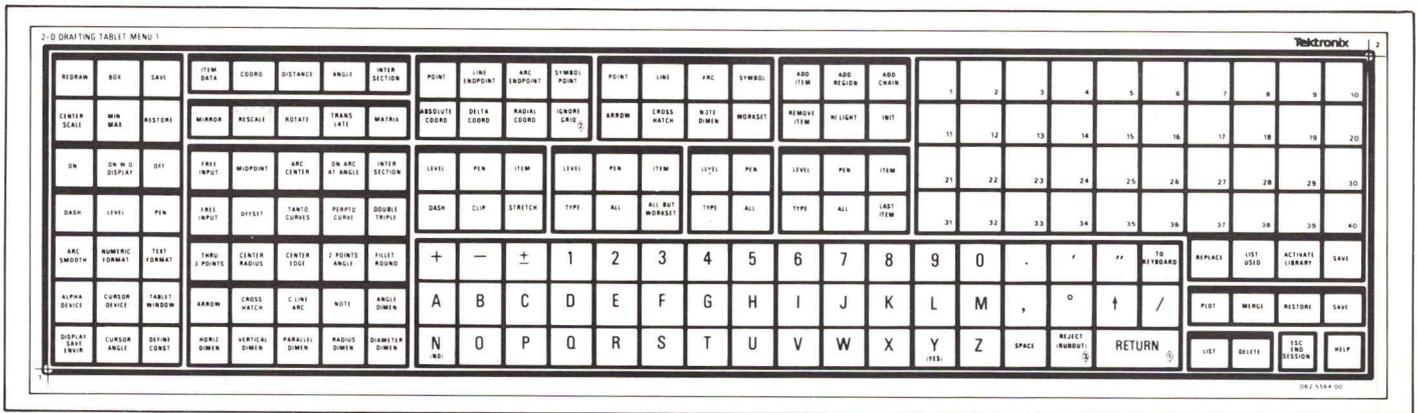


Figure 3. By selecting a function from the PLOT 50 2-D Drafting menu, the drafter commands the system.

template or compass, place it on the sheet and draw the circle. With PLOT 50 2-D Drafting, the drafter would pick up the tablet cursor and touch the *center radius* square or *center edge* square on the tablet menu (located under *ARC/CIRCLE* in figure 5), then follow screen prompts to locate the points.

In other automated drafting systems that are command based, the drafter would be required to remember and to type in the appropriate commands to construct the circle. In automated systems with a menu hierarchy, the drafter would have to remember in which family of operations the circle-drawing method resides, would have to pick the appropriate master menu, then would have to walk through submenus to reach the "circle of certain radius."

Positioning graphic elements or annotation in PLOT 50 2-D Drafting is similar to manual methods. Just as either axis of a drafting machine may be locked so it moves only in the X or only in the Y axis, either the X or Y thumbwheel of the 4054 may be used to move along the X or Y axis. Or, like the free movement of the machine, both thumbwheels may be used at once for movement in any direction.

The manual drafter slides the template around on the drawing until it is aligned, and then draws the symbol. The dynamic graphics feature of the 4054 Desktop Computer allows symbols and text to be moved around on the screen before positioning, minimizing redraws.

But communication methods are not limited to the tablet menu and 4054 thumbwheels. The tablet cursor may be used to trace a sketch placed on the tablet surface. Text and numerics for annotation or positioning may be entered from the 4054 keyboard or from the tablet menu character pad. The drafter may alternate among input methods, using whichever is most comfortable and convenient.

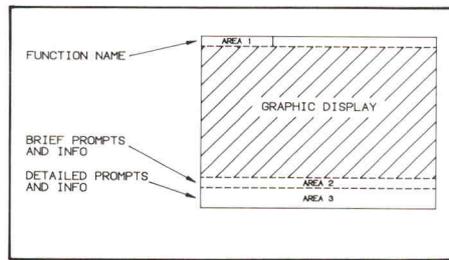


Figure 4. The 4054 screen is divided into the graphic display and three message areas. The function name indicates which function is in current use. Prompts, instructions, and help and error messages are displayed at the bottom.

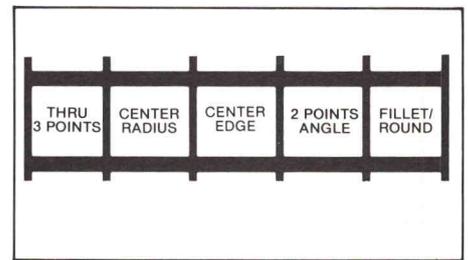


Figure 5. Drafting steps, such as drawing circles, are chosen from familiar function squares on the PLOT 50 2-D Drafting tablet menu.

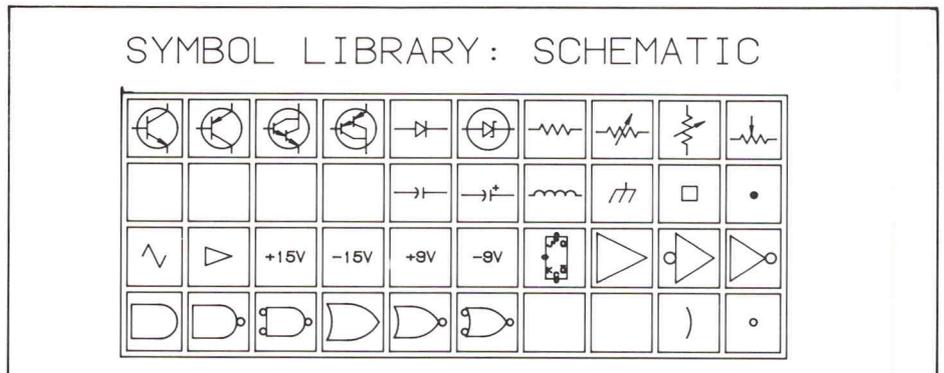


Figure 6. Symbols may be created and then stored in libraries to be called back for insertion in drawings.

Thus, communicating with PLOT 50 2-D Drafting is flexible and natural. A drawing is created or revised in logical drafting steps by choosing the appropriate function from the tablet menu. The screen displays the current step and makes clear to the drafter the options available and the valid responses. Positioning through the thumbwheels and dynamic graphics is simple and straightforward. Computer experience or programming knowledge is not required, reducing learning time and speeding productivity.

Computer Power Accelerates Drawing Creation and Revision

Every engineering drawing contains at least small areas of duplication — a portion of

the drawing itself, patterns, notations — and most make extensive use of symbols. PLOT 50 2-D Drafting ensures that a drafter never draws anything twice.

Commonly used symbols, objects or subassemblies may be drafted once, then repeatedly copied in the drawing or stored in a library for use in other drawings (figure 6). These electronic templates reproduce items in a fraction of the time it would take manually, while ensuring accuracy and completeness.

Aside from symbols, however, the power of this feature may not initially be recognized. Many areas of duplication in engineering drawings go unnoticed since seeing these relationships is of no advantage for the manu-

al drafter. Patterns in the lower left-hand area of a drawing must be hand drawn again if they occur in the upper right. A mirror image must be copied again. PLOT 50 2-D Drafting replaces manual drawing repetition with computer copying power (illustrated in figure 7).

“Snap-to” capabilities are an important advantage of the automated drafting system. Like graph paper and a T-square, an electronic grid guides the drafter. Similar to moving the drafting machine or templates, the automated drafter uses the 4054 thumb-wheels or tablet cursor to position an element or symbol upon the grid. But the manual drafter must precisely align the symbol template or scales on the drafting machine before drawing, whereas PLOT 50 2-D Drafting allows the drafter to just get close to the grid point and the system will “snap” the items to perfect alignment. Lines may be snapped into true horizontal or vertical alignment; symbols may be snapped to a position on an existing drawing element.

Locations may also be entered without the grid, by absolute coordinates, by delta values, by radius values, or by snapping to existing items in the drawing. Thus, precisely connecting or placing symbols, elements or notation is efficient and quick.

PLOT 50 2-D Drafting allows complex drawings to be separated into logical layers, similar to the multiple floors of a building (illustrated in figure 8). This especially speeds drawing revision. Common drawing elements such as geometry or dimensions can be easily assigned to different levels. A level can also form a base for subsequent drawings.

With 127 levels available in a PLOT 50 2-D Drafting drawing, an engineering office could optimize this feature to enhance the ability of drafters to edit and revise drawings. Office standards could be established such as: certain components of electrical work on one level, other components on another level, and mechanical work on still another. Revision affecting only one level could be accomplished without impacting other levels.

Since the levels can be turned on or off in any combination, only those levels containing the information of interest need be displayed, avoiding the confusing clutter of unneeded detail.

Blanking also reduces drawing complexity or density by removing items from the display but not from the drawing. This allows the drafter to work with a less complex display on the screen. A touch of the pen returns the drawing to its original complexity, permitting the drafter to see the impact

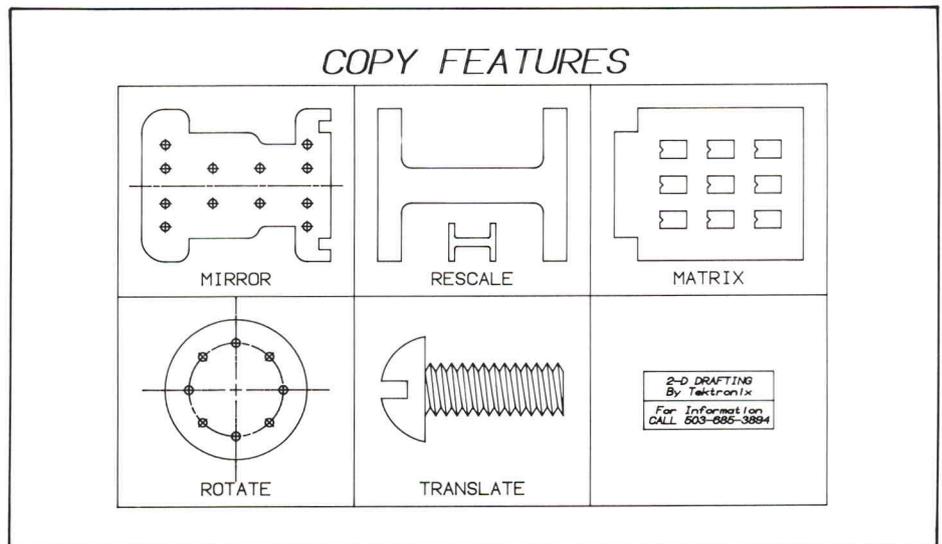


Figure 7. Drawing elements may be created once then copied to other areas as mirrored images, at a different scale, as reproductions in a regular matrix, or at different angles.

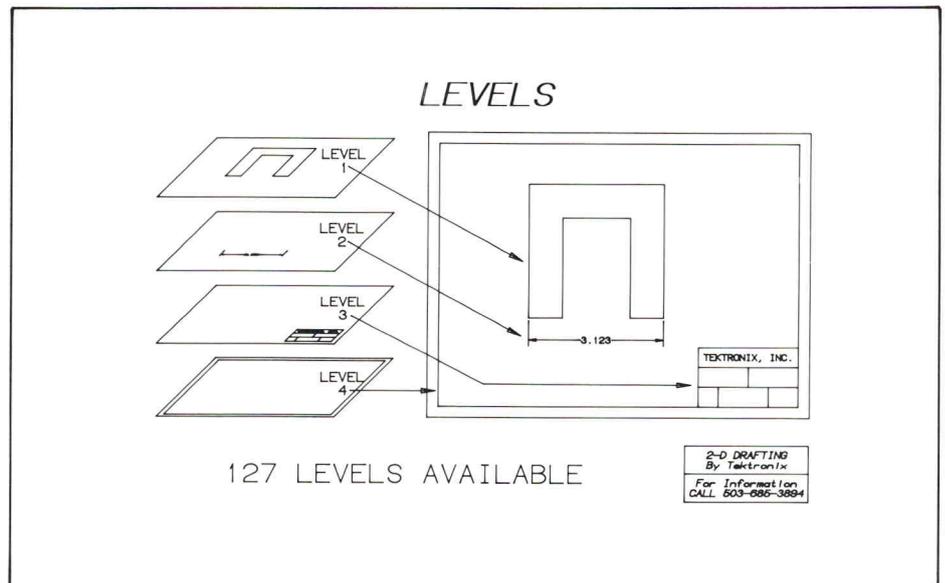


Figure 8. Up to 127 levels can separate complex drawings into easy-to-edit components, facilitating drawing control and intra-office communication.

of the edited portions in the context of the total drawing. This would be difficult to do manually.

PLOT 50 2-D Drafting will zoom in on a portion of the drawing to display a smaller area but with greater detail. Thus, the smallest item or a group of items may be quickly accessed for precise revision.

The power of the automated system eliminates erasers. Pressing a key can delete a symbol or part of a picture. But before actually eliminating anything from a drawing, the drafter can remove it from the screen to visually verify the decision. Converting solid lines to dashed lines can be done in seconds. Not only is electronic revision much faster, but smudges and torn paper are a thing of the past. So often in manual methods, an engineering drawing that began life as a

beautiful ink on mylar, or pencil on paper drawing becomes an illegible, thin, smudged and smeared sepia through the revision process. With PLOT 50 2-D Drafting, whenever revisions or changes are made, an original plot can be generated, quickly and cleanly. The user is spared interpreting illegible drawings.

Once a drawing is completed and stored, it needn't be redrafted if used as part of another drawing. A merging function eliminates unnecessary repetition by allowing a stored drawing to be merged at any location in a current drawing, at any scale and rotation.

Whether a drafter is incorporating duplicate elements into a drawing, aligning items, or revising, PLOT 50 2-D Drafting's power

speeds the operation and extends capabilities.

Flexibility Accommodates Differences

PLOT 50 2-D Drafting, however, doesn't sacrifice flexibility to achieve speed. Graphic elements — arcs, symbols, lines or annotation (shown in figure 9) — may be designed in many different ways. *Free Input* gives the drafter the liberty to specify locations by any of nine different methods (see figure 10). Other geometric designations place the elements in precise relationships (figure 11).

A variety of dimensioning (figure 12), lettering styles, and drawing scales permit several industry-recognized methods. Flexible dimensioning automatically generates the dimension value (with manual override) of an item, extension lines, and dimension lines with arrowheads. Whether the requirement is horizontal, vertical, parallel, radius, diameter or angle dimensions, PLOT 50 2-D Drafting can handle it. And the values may be placed just about anywhere.

Formats for numeric input will satisfy almost any application. Units of measurement fit the drafter's need.

Even communicating with the system is flexible. PLOT 50 2-D Drafting allows the drafter to respond to prompts or questions through the 4054 keyboard, the tablet, or both.

Automating Provides Quality Engineering Drawings

Precise, clean drawings may be plotted in a matter of minutes. Even though several drafters may have worked on a drawing, its elements will be consistent, its letters uniform, its dimensions accurate. And the original quality is always there — no smudges or erasure dust.

PLOT 50 2-D Drafting drawings may be sent to the Tektronix 4662 B-size plotters, to the 4663 C-size plotter, or to selected D- or E-size plotters.

Cost-Effectiveness Goes Beyond Drafting

PLOT 50 2-D Drafting's feature set and performance is a cost-effective alternative to large, high priced CAD systems. Depending on the drafting task, productivity gains of two to 20 times are possible over manual methods — without expanding staff or facilities. And the professional computerized output will satisfy the most critical client.

Furthermore, PLOT 50 2-D Drafting conforms to the PLOT 50 Graphic Model Exchange File standard*. This means the graphic files produced by PLOT 50 2-D Drafting may be transferred to other programs which observe this standard.

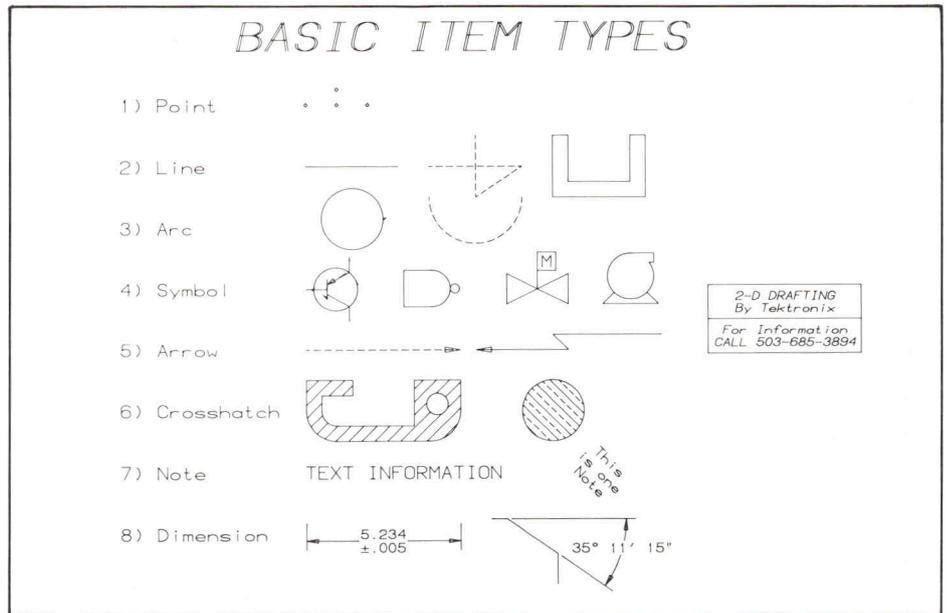


Figure 9. Eight basic elements of PLOT 50 2-D Drafting are used to create or revise a drawing.

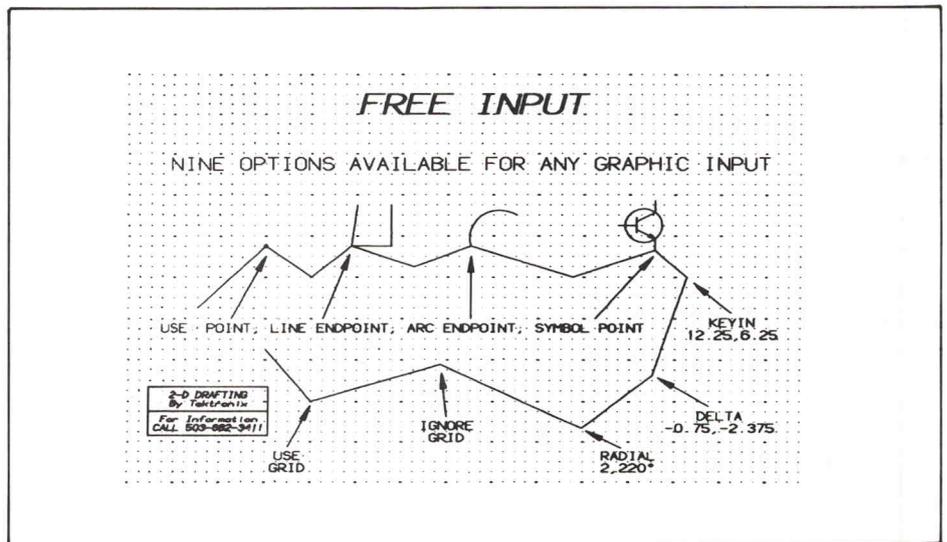


Figure 10. Free input is more than just moving the cursor around the drawing area. Entries may be aligned to a grid, to existing points, to line endpoints, to arc endpoints, or to symbol connection points. The drafter may specify graphic locations as absolute drawing coordinates, as delta-x and delta-y from the previous free input location, or as radial coordinates (radius and angle).

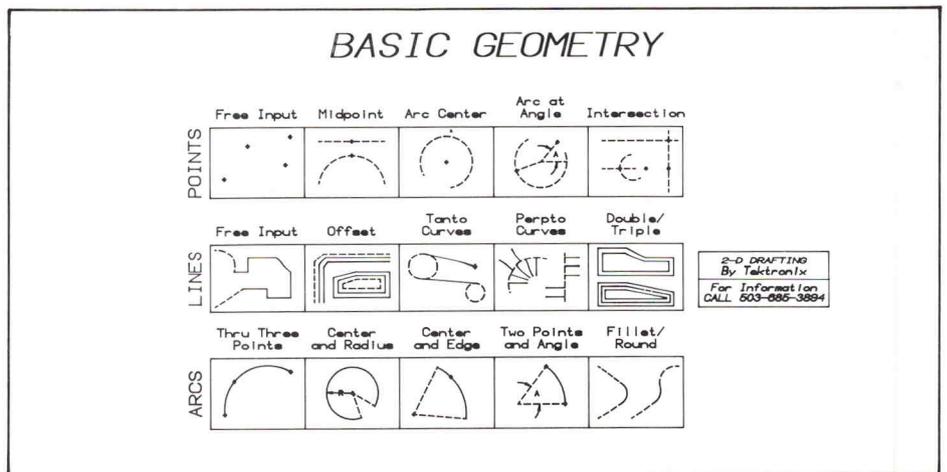


Figure 11. A variety of input methods help the drafter create all types of fundamental geometry with means appropriate to the situation.

But the Tektronix system is not limited to improving drafting productivity. The 4054 is a powerful standalone computer and terminal which provides access to an engineering/scientific portfolio of software. Productivity gains are possible in other areas also.

As with all Tektronix products, PLOT 50 2-D Drafting is backed by a commitment to excellence and a worldwide support team.

Get in touch with the local Tektronix sales engineer for more information on automating with the PLOT 50 2-D Drafting workstation. 

3.14159	DECIMALS
5/16	SIMPLE FRACTIONS
3-13/16	INTEGER WITH FRACTION
15' 3"	FEET AND INCHES
48° 15' 3"	DEGREES, MINUTES, SECONDS
C5	USER CONSTANT
C° C6' 14"	MULTIPLE USER CONSTANTS
15' - 3 3/8"	MIXED FEET AND INCHES
6.02 E+23	SCIENTIFIC NOTATION

Figure 13. Numeric values may be entered in a variety of formats and units.

* The PLOT 50 Graphic Model Exchange File standard was discussed in TEKniques Vol. 5, No. 2.

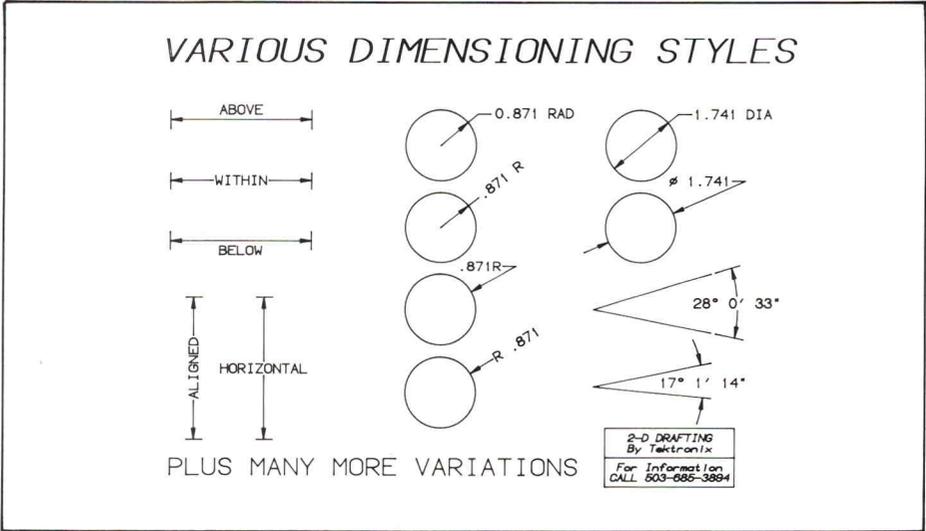


Figure 12. Flexible dimensioning styles allow company standards to be maintained.

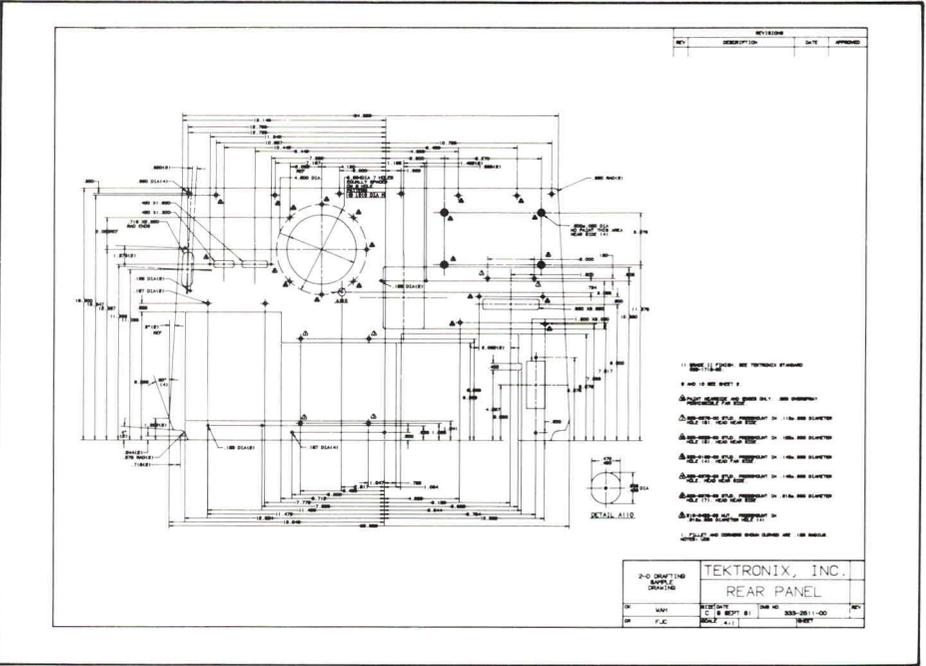
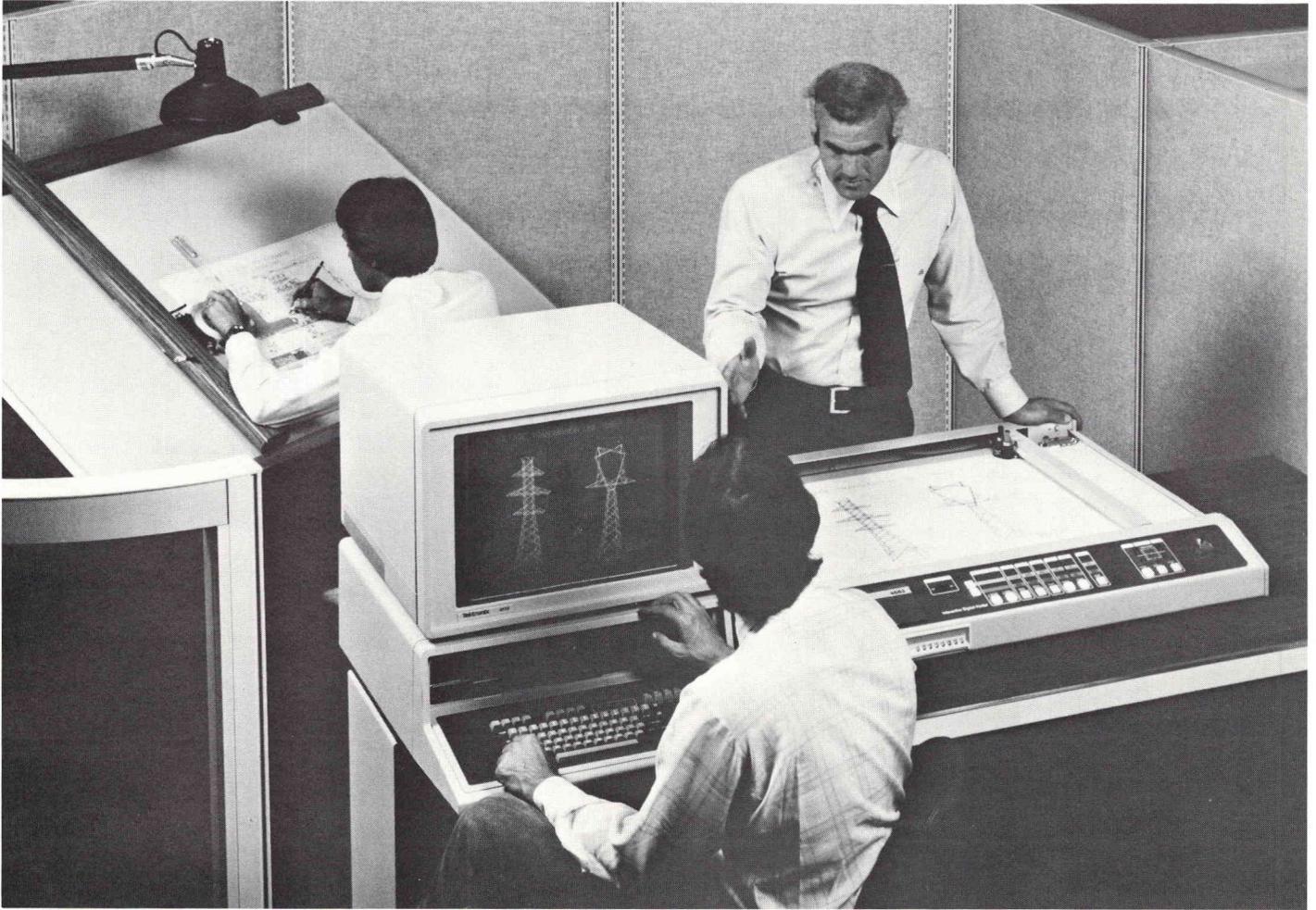


Figure 14. Regardless of the number of revisions, the drafter can produce original quality drawings with PLOT 50 2-D Drafting.

Solution Vendor Program Informs Customers of Third Party Software



The rapid growth of computer graphics for problem solving is bringing an intensified interest by graphics users in prepackaged applications software. Consequently, Tektronix has inaugurated a program to make customers aware of application-specific software designed for Tektronix 4050 Series Desktop Computers, 4010 Series Terminals and the new 4100 Series Terminals.

Promoting the expertise of outside resources, the Tektronix Solution Vendor Program will identify software offerings, verify a satisfied customer base, and publish the *Solution Vendor Register*. Through its extensive field sales force and marketing efforts, Tektronix can effectively channel third party software information to existing and prospective customers. The Solution Vendor Program will simplify the customer's task to find cost-effective solutions.

The program will focus on engineering and scientific applications. Computer-aided de-

sign, cartography, computer-aided manufacturing and technical data analysis packages are currently being considered.

Submittals accepted in the Solution Vendor Program will be announced in *TEKniques*. *TEKniques* will publish articles profiling characteristics and applications of selected software packages.

By informing customers of application programs developed by selected third party vendors, Tektronix hopes to help users choose timely solutions to their problems, solutions which directly affect productivity and competitiveness. 

BASIC Language Enhancements Highlight New Models

by Steve Duncan
and Mark Mehall
Tektronix, Inc.
Wilsonville, OR

As technology advances, so does the 4050 Series. Two new desktop computers introduced in June replace the existing 4052 and 4054. Known as the 4052A and 4054A, the new systems bring an improved BASIC language, a more accessible GPIB, and overall speed increases to the 4050 family of stand-alone computers, while retaining the same compact appearance and compatibility.

A Powerful BASIC

The 4052A and 4054A Desktop Computers offer an enhanced BASIC language. Introduced into the friendly extended BASIC of the 4050 Series are 46 new features for more structured programming and flexibility. Multicharacter variable names of up to 31 characters may be used for numeric or string variables. Variables may be redimensioned upward. Named "callable" subroutines with local variables and parameter passing strengthen program design and allow greater control over memory.

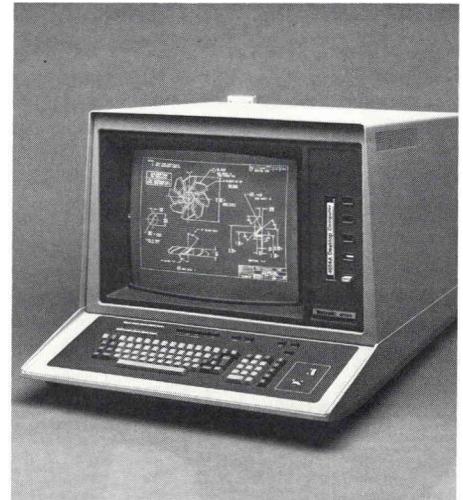
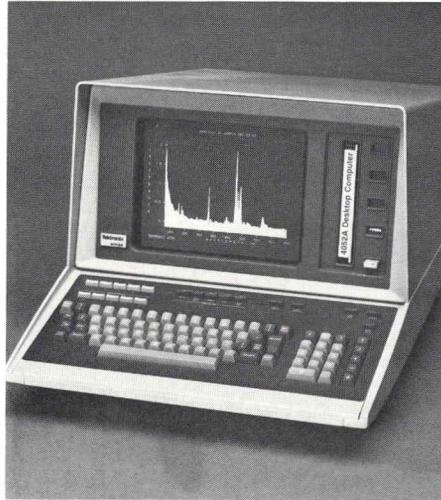
Powerful conditional branches are provided in expanded IF statements and loop processing.

Improved program editing is implemented through several commands. RENUMBER will operate on blocks of code, or the entire program. Comments may be added to the end of executable statements following the "!" character. REMARK and comment tails can be stripped from programs being loaded. A new status command allows program listings to be indented at each IF, FOR, DO and SUB command, with END SUB, EXIT IF END IF, NEXT and LOOP cancelling an indent. Several of these features are illustrated in the following code.

Flexibility in character string handling is increased. Multiple concatenations and string functions may now be performed on one line:

```
list 1000,2000
1000 A$="one string"&CHR(13)
      &"another string"
1010 PRINT A$

run 1000
one string
another string
```



A powerful BASIC language coupled with an extremely accessible GPIB and fast RS-232 endow the TEKTRONIX 4052A and 4054A Desktop Computers with superior capability. The new functions of the 4052A and 4054A can speed tasks in the design room, in the research lab or on the manufacturing floor.

```
list
100 DO
110   PRINT "NUMBER: ";
120   INPUT I
130   EXIT IF I<=0 ! Terminate program on illegal argument
140   CALL Fac(I,Fac_i)
150   PRINT "FACTORIAL ";I;"=";Fac_i
160 LOOP
170 PRINT "DONE"
180 END
190 SUB Fac(N,X) ! Recursive definition
200   IF N=0 THEN
210     X=1
220   ELSE
230     CALL Fac(N-1,X)
240     X=X*N
250   END IF
260 END SUB

run
NUMBER: 1
FACTORIAL 1=1
NUMBER: 10
FACTORIAL 10=3628800
NUMBER: 100
FACTORIAL 100=9.332621544E+157
NUMBER: 0
DONE
```

This small program — using the familiar example of factorial recursion — illustrates several features of the improved 4052A/4054A BASIC language.

Lines 100–160 contain a DO/EXIT IF/LOOP. The EXIT IF statement provides a clean way to get out of the DO/LOOP, allowing greater control over branching and memory.

Statement 140 calls a named subroutine (FAC) and passes a variable (I) to the subroutine. The result of the computations in the subroutine will be returned in a multicharacter variable (Fac_i).

An example of the expanded IF statement, the IF...THEN statement in line 200 transfers control to the next statement if the expression is true. If the expression is false, then the 4050 interpreter looks at succeeding lines for the matching ELSE or END IF statements.

Two parameter passing techniques are illustrated. A call by reference occurs in the initial call to the subroutine at line 140 since a variable is being passed. A call by value occurs at line 230 since an expression is being passed.

Lines 130 and 190 contain comment tails.

The indented listing clearly shows the program structure.

The CHR command allows arguments between 0 and 255, and the ASC command can return the value of any element in a character string, rather than just the first.

Several new commands expand string functions. SEARCH examines strings for specified matches. Producing a new string from a source string and a translate table is possible with TABLE command, for example, to convert ASCII to EBCDIC. TRIM removes leading and trailing blanks from strings. With the conditional character input command, CCINPUT, the type ahead buffer may be read one character at a time without screen echo. ALTER allows a string to be modified using the line editor keys.

A valuable feature for instrument control, data acquisition or automated test systems is bit manipulation in character strings. Strings may be treated as binary data and operated on by eight different commands.

An Efficient GPIB

Redesigned GPIB hardware implements faster data transfers and provides greater control of the GPIB lines. The 4052A and 4054A have increased data transfer rates of up to four times those of the 4052 and 4054. Improved line control permits easy access to the IFC and REN lines.

Several new commands expand GPIB programmability. An addressing list allows multiple devices to be specified with a single statement. CONFIGURE tells which devices are active on the GPIB, and OFF SRQ ignores unwanted service requests. Timeout capability prevents bus hang ups.

A Strong RS-232

The 4052A and 4054A Data Communications Interface (Option 1) has been re-engineered to provide expanded data transmission capabilities at rates up to 9600 baud. For faster and more effective data transfer to printers and other output devices which use the DTR line, hardware flagging has been included in the Option 1 of the new "A" models.

Soft flagging is also supported on both input and output transmissions using XON-XOFF protocol. This allows the 4052A and 4054A to receive and transmit data without buffer overflows on the host. Data transmission is much more reliable.

Faster Processing

Existing programs will run up to 20% faster on the 4052A and 4054A. By reprogramming to use the new features, programs will run even faster. Areas of improvement include string operations, array indexing, di-

mensioning, defined functions, renumber and append operations.

Family Compatibility

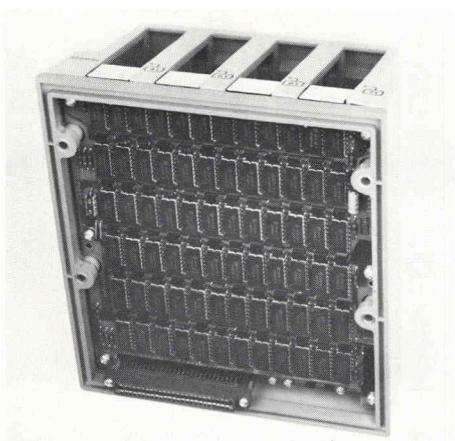
Maintaining the commitment to compatibility in the 4050 Series, the 4052A and 4054A can run programs developed on the 4051, 4052, or 4054. This means all PLOT 50 software, Applications Library software and other user-developed software can be transferred to the 4052A or 4054A without changes. However, frequently used or critical programs can be converted to take advantage of the new features while other programs could be maintained as is. The investment in previously developed software is secure.

The new features give the 4052A and 4054A one of the most powerful BASIC languages available. The increased speed of the GPIB and versatility of the RS-232 provide quality I/O interfacing. All applications areas will benefit from the additional strength and ease of use of these two new desktop computers.

Field kits are available to upgrade existing 4052 and 4054 systems with all the functionality of the "A" models. Check with your local Tektronix Sales Engineer for the details of the 4052A and 4054A Desktop Computers. 

Extended Memory File Manager Expands High Speed Memory in the 4052/54 Series

by Mark Mehall
Tektronix, Inc.
Wilsonville, OR



If you require fast access to data files or fast access to program and overlay storage, consider the new TEKTRONIX Extended Memory File Manager. Complementing 4050 main memory, the Extended Memory File Manager adds either 256K bytes (Option 27) or 512K bytes (Option 28) of random access memory.

Data can travel at speeds of up to 400K bytes per second between Extended Memory and the 4052A/54A,* and at speeds of up to 50K bytes per second to and from the 4052/54. Program loading and storing time is significantly decreased.

The Extended Memory File Manager is packaged in a special version of the Option 3 Data Communications Interface backpack so it doesn't require backpack slot space. Because special ROMs contain its operating system, no 4050 memory is taken up.

A built-in battery backup power supply means the Extended Memory File Manager retains information indefinitely when the 4050 is plugged in, or for a minimum of four hours after the unit is unplugged. Thus, frequently used programs or data are instantly accessible upon power up without searching and reloading from tape or disk.

The Extended Memory File Manager is compatible with the 4907 File Manager. In fact, its command set is a subset of the 4907 File Manager commands with some of the restrictions removed, plus diagnostic test commands (see sidebar).

File organization is uncomplicated and similar to the 4907 (a proper subset). You store a file by name (up to 10 characters), plus an optional extension (up to four characters); the file may be random or sequential. Multi-file access is convenient for data comparison or file sorting.

Utility functions include a directory that maintains user files and available space, and internal diagnostic routines.

Whether you are upgrading your present 4052/54 with the Extended Memory field kit, or ordering a new 4050 system with the Extended Memory option, you automatically receive the four slot backpack and the data communications interface. For more details, call your local Tektronix Sales Engineer. 

* An overview of the new 4052A/54A is contained in a separate article in this issue.

Built-In Diagnostics Enhance Extended Memory File Manager

by Richard Bailey
Tektronix, Inc.
Wilsonville, OR

Although the Extended Memory File Manager doesn't require routine maintenance or adjustment, and Tektronix makes every effort to prevent equipment failure, malfunctions can occur (Murphy's famous law). When that happens, it's nice to know where the breakdown occurred, get it fixed and get going again. To aid in troubleshooting the 4050 Extended Memory if the need arises, a power-up test, a unique diagnostic test and several helpful utilities have been included in the Extended Memory's operating system.

Power-Up Test

When the 4050 equipped with Extended Memory is powered up, the Extended Memory operating system determines whether the Extended Memory unit is working correctly. It checks the data register, the operation of its main memory, the parity generator and checker, and the volume label. If any of the first three fails, or it finds the volume label corrupted, an error message will be displayed on the 4050 screen.

Extended Memory Test

For a more thorough checkout, the Extended Memory firmware also contains a diagnostic routine called "TESTEM" (Test Extended Memory) which will run a variety of memory tests which can isolate circuit failure to the functional block level and a

RAM failure to the chip level. The test command is:

```
CALL "TESTEM", TYPE$, LEVEL$, STATUS$
```

where:

TYPE\$ is a string variable input parameter that defines the type of test to run

LEVEL\$ is a string variable input parameter that defines the severity of the test

STATUS\$ is a string variable output parameter that accepts the results of the test

TYPE\$ may have the following values:

"D" (Destructive test)

"N" or "" (Nondestructive test)

LEVEL\$ may have the following values:

"A" Parity Test on current files

"B" Level A plus Pattern Test

"C" Level B plus Address Test

"D" Level C with additional diagnostic information output to the status string and I/O display.

STATUS\$ will contain a message depending on the results of the tests:

By building in diagnostic test functions and access utilities, the Extended Memory File Manager's dependability is enhanced by its serviceability. That's an important plus when Murphy's law takes over. 

Command Summary

Command	Function
File Management Commands	
CLOSE	Closes one or more files
COPY...TO	Copies one or more files from one file to another
CREATE	Creates file space on extended memory
DIRECTORY	Lists information about files
FILE	Returns status of named file
KILL	Deletes existing file
OPEN	Identifies file, type of access and returns status
RENAME	Changes names of files and extensions
REWIND	Repositions pointer to beginning of sequential file
SPACE	Reduces or increases assigned file space
File I/O Commands	
APPEND	Adds program lines to program currently in 4050 memory
INPUT	Retrieves ASCII data from extended memory
OLD	Loads BASIC program from extended memory into 4050 memory
PRINT	Stores ASCII information on extended memory
READ	Retrieves binary data from extended memory
SAVE	Stores program presently in memory, on extended memory
SECRET	Prevents future listing or copying of a program
WRITE	Stores binary information on extended memory
System Commands	
DELALL	DELETE ALL closes all files and erases memory
INIT	Initializes File Manager
UNIT	Allows selection of extended memory device
Controller Device Commands	
CUSTAT	CONTROLLER UNIT STATUS generates and returns the status of extended memory and 4907 File Manager (if present)
DSTAT	DEVICE STATUS returns status of addressed unit
FFRMT	FAST FORMAT formats device (removes all files)
FMVALS	FILE MANAGER VALUES return to the program or user the number of the current device
FORMAT	Initializes extended memory for data storage (removes all files)
HERRS	HARD ERROR STATUS returns number of hard errors
MRKBBG	MARK BAD BLOCK GROUP allows marking and bypassing of defective 256 byte blocks
Diagnostic Test Commands	
TESTEM	Runs a variety of memory tests

Price/Performance Advantage in Tektronix 4663S Plotter



Circular interpolation, programmable macros and downloadable character sets are just a few of the features built into the Tektronix 4663S C-Size Plotter.

by **Craig Montgomery**
Tektronix, Inc.
Wilsonville, OR

Whether the task is computer aided design or information management, the newly con-

figured 4663S C-Size Plotter offers an increased selection of functions to ease plotting jobs.

Features normally obtainable only as options are built into the 4663S. Circular interpolation permits drawing of circles or

arcs with a single command, and choice of smoothness. Programmable macros can be used to construct special symbols to tailor data representation. Downloadable character sets and the ability to program and store four operating environments benefit any discipline. Two resident interfaces, the RS-232 and the GPIB, allow interfacing with most any computer or desktop computer.

The firmware commands, alphanumeric, digitizing capabilities and operating ease of the 4663S make it a natural fit for printed circuit board layout, numerical control, mapping, drafting, report generation or other areas needing a medium large format plotter.

For those plotting tasks not requiring the enhanced performance of the 4663S, the 4663 Interactive Plotter is still available. And, of course, the Paper Advance, with form length either host or front panel programmable, may be ordered with either plotter.

Your local Tektronix Sales Engineer will be happy to provide you with details of the 4663S Interactive Digital Plotter. 

Plotter Manuals Updated

Two new manuals for the 4662 Interactive Digital Plotter replace the previous single manual, and two for the 4663 Interactive Digital Plotter replace the two previous manuals. New formats in both sets provide quick access to items of interest. Programming and device information has been expanded.

The *Operator's Manuals* introduce the major features of the plotters and guide the operator through a series of operations required to prepare the plotters for plotting. Much of the information is tutorial.

Step-by-step instructions familiarize the user with the front panel controls (and rear switches for the 4662). The 4663 *Operator's Manual* includes additional technical information about the extensive functions of the front panel switches and indicators.

Sprinkled liberally with illustrations and photographs, a section in both manuals describes all aspects of the plotter's pens and media, including pen types, media types, instructions for loading media, pen installation, selection and maintenance. The manuals also review instrument care and cleaning.

Aimed at the programmer who is writing a package of computer subroutines that communicate with the 4663 or 4662 Plotter, the *Programmer's Reference Manuals* describe the plotter's commands and tell how to issue them by entering sequences of ASCII characters at a compatible terminal or controller.

Fundamental plotter graphics concepts and first-time operation of the plotters is covered. The manuals provide specific programming needs for Serial (RS-232) or GPIB in-

terfaces. Explicit instructions for using each plotter command followed with meaningful examples promote quick understanding and effective use of the plotters.

The four manuals and revised reference cards may be ordered through Tektronix Central Parts Ordering (see page 32 in this issue of *TEKniques*). Your local Tektronix Sales Office can answer any questions. 

4662 Interactive Digital Plotter

(all material includes Option 31 information)

Operator's Manual	070-4165-00	Nov 81
Programmer's Reference Manual	070-4164-00	Apr 82
Programmer's Reference Guide	070-2556-01	Feb 82

4663 Interactive Digital Plotter

Operator's Manual	070-3807-00	Jan 82
Programmer's Reference Manual	070-3806-00	Jan 82
Programmer's Reference Guide	070-2828-01	Feb 82

Programmable Spectrum Analyzers and 4050 Desktop Computers Automate Signal Monitoring

by Dave Barnard
Tektronix, Inc.
Beaverton, OR

Two spectrum analyzers produced by Tektronix are fully GPIB programmable and may be controlled by a 4050 Desktop Computer. The 492P is a microprocessor-controlled spectrum analyzer serving the higher frequency applications from 50 KHz to 220 GHz. The 496P mp-controlled spectrum analyzer is designed for use in the 1 KHz to 1.8 GHz range.

In both spectrum analyzers an internal microprocessor simplifies and enhances both manual and remotely controlled applications. Compact size and rugged construction make these spectrum analyzers transportable to the field or from one laboratory application to another.

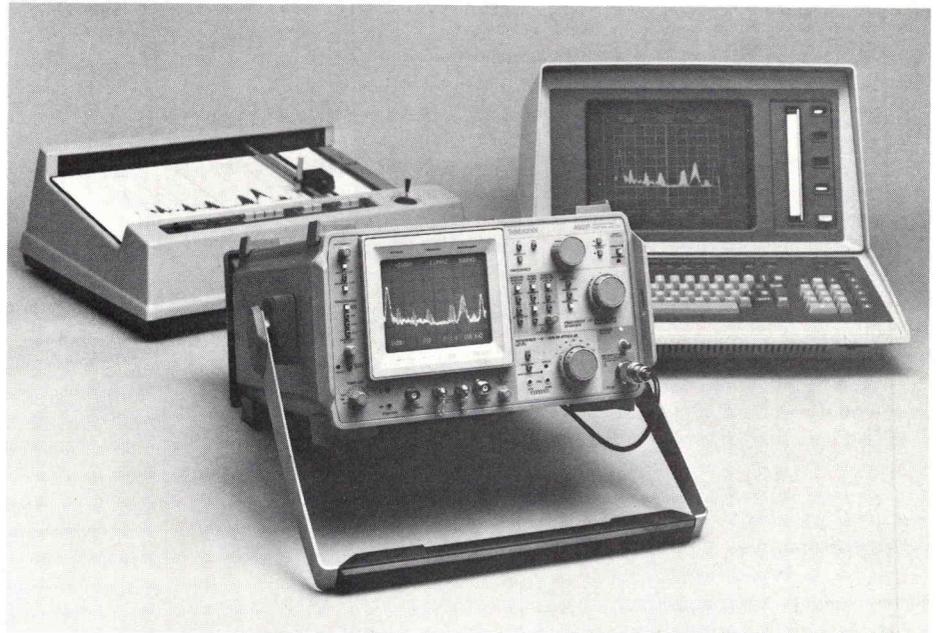
When connected by the IEEE Std-488 bus to a 4050 Series Desktop Computer, measurements can be fully automated. Automated measurements reduce operator fatigue and error, provide repeatable tests, and save time. Here again the built-in intelligence in the 492P and 496P gives an assist by:

- Adding finer control to some control settings than is possible from the front panel. For example, by front panel, manual control, the vertical scale is either 10 dB or 2 dB per division in logarithmic mode. Under program control the vertical display may be from 1 dB to 15 dB per division.
- Searching display data stored in memory for signals using commands such as FMAX,FMIN plus others.
- Providing English-like commands for control settings. Most are abbreviations of the front panel control names.

Simple, logical commands lead to reduced software development and maintenance costs. Programs are more readable both to the originator and to personnel who may have to maintain the software.

For example, to set the spectrum analyzer center frequency to the center of the FM broadcast band (98 ± 10 MHz), a 4050 program line would be:

```
PRINT@Z:"FRE 98MHZ;SPAN 2MHZ"
```



Portable and easy to use, the TEKTRONIX 492P Spectrum Analyzer is fully programmable from the 4050 Desktop Computer.

The spectrum analyzer responds by moving its center frequency to 98 MHz and setting the scanned frequency range to 2 MHz per division for a total width of ± 10 MHz over the full range of the CRT display.

Tektronix Codes and Formats Standard of communicating over the bus is fully implemented in both spectrum analyzers. To inquire about a measurement parameter, the command is followed by a "?".

All instrument settings can be read at once using the SET? command. As a result, a series of measurement steps can be memorized by the 4050 with the operator as the instructor.

An example of a learning program is in figure 1. Line 100 and 110 prompt the oper-

```
100 PRINT "ENTER BUS ADDRESS"  
110 INPUT Z  
120 DIM A$(400),B$(400)  
130 PRINT "MAKE FIRST SET-UP PRESS RETURN WHEN DONE"  
140 INPUT D  
150 PRINT @Z:"SET?"  
160 INPUT @Z:A$  
170 REM MAKE NEXT SET UP  
180 PRINT "NEXT? PRESS RETURN WHEN DONE"  
190 INPUT D  
200 PRINT @Z:"SET?"  
210 INPUT @Z:B$  
220 END
```

Figure 1. The "learn" mode of operation is illustrated in this program. The 4050 monitors and stores all the measurement parameters of each of the two steps in this simplified example.

Spectrum analyzers play an important role in the telecommunications and broadcast industries. Although not a household word, a spectrum analyzer probably played a role in evaluating performance of the television broadcast system you enjoy in your home. If you use a microwave oven, it too may have been examined at one time with a spectrum analyzer.

Whether used to measure quality of signals off the air, in cable TV systems, or at satellite Earth terminals, the method is the same. The spectrum analyzer displays the components of the signal in order of frequency on its cathode-ray tube (CRT). This amplitude versus frequency (frequency domain) technique is a powerful evaluation and troubleshooting aid also used to maintain radar systems such as those used for aircraft navigation.

ator to enter the bus address of the analyzer. Line 120 dimensions character strings to hold the ASCII encoded strings of information. Lines 130 to 160 prompt the operator, wait for completion (wait for "return"), and read all the settings of the analyzer into the string variable. Lines 170 to 210 repeat this for another measurement in the sequence. This scheme can be expanded to include more steps.

When the measurement must be echoed back to the analyzer, it can be done as part of a more elaborate program or simply placed under function key control as shown in figure 2. The first and second set-ups are recalled and sent to the analyzer by function keys 1 and 2 respectively.

```

1 GO TO 100
4 PRINT "REPEAT OF FIRST LEARNED SETTING"
5 PRINT @2:A$
6 END
8 PRINT "REPEAT OF SECOND LEARNED SETTING"
9 PRINT @2:B$
10 END

```

Figure 2. Restoring measurements to prerecorded conditions can be as simple as pressing a User-Definable Key.

If necessary, setting information can be processed for display formatting or storage on the 4050 magnetic tape. This setting information is also humanly readable because it is in printable ASCII. Settings come out of the analyzer in the same command format required to control the analyzer. No SET command is needed to restore control setting, just the command sequence. It's as simple as sending A\$ or B\$ (shown in figure 3).

The high level commands for the 492P and 496P are identical except for one extra command in the 496P (zero span). Programs developed for one spectrum analyzer will work on the other within the frequency range covered by the analyzer.

```

A$
FINE OFF;DELFR OFF;FRQRNG 1;EXMYR OFF;MINATT +0;RLMODE MDIST;REFLUL +30,
0;FINE OFF;URTDSP LOG:10;FREQ +2.39E+8;DELFR OFF;SPAN +5.0E+7;PHSLK OFF;
IDENT OFF;RESBW AUTO;PEAK KNOB;TIME AUTO;TRIG FRERUN;AUIEW ON;BUIEW ON;S
AUEA OFF;BMINA OFF;MXHLD OFF;CRSOR KNOB;PLSTR OFF;UIDFLT OFF;REDOOT ON;I
G RAT OFF;CLIP OFF;WFMPRE HFID:FULL,ENCDG:ASC;POINT 500,225;RQS ON;EOS OFF

B$
FINE OFF;DELFR OFF;FRQRNG 1;EXMYR OFF;MINATT +0;RLMODE MDIST;REFLUL +30,
0;FINE OFF;URTDSP LOG:10;FREQ +8.02E+8;DELFR OFF;SPAN +5.0E+7;PHSLK OFF;
IDENT OFF;RESBW AUTO;PEAK KNOB;TIME AUTO;TRIG FRERUN;AUIEW ON;BUIEW ON;S
AUEA OFF;BMINA OFF;MXHLD OFF;CRSOR KNOB;PLSTR OFF;UIDFLT OFF;REDOOT ON;I
G RAT OFF;CLIP OFF;WFMPRE HFID:FULL,ENCDG:ASC;POINT 500,225;RQS ON;EOS OFF

```

Figure 3. Response to the "SET?" query is lengthy — and detailed enough to restore all settings to a prerecorded condition.

Input/ Output

The following question comes from David Walcutt, propagation & frequency management specialist at Radio Free Europe/Radio Liberty in New York City.

When a series of matrices have to be put through the same operation, in my case a sorting operation, is there any straightforward way to do the repetitive operations in a GOSUB instead of having to repeat the steps of programming for each separate matrix? (See example 1.)

I had five matrices — C1, C2, C3, C4 and C5 — and wanted to do something such as sort on C(N) where N=1, N=2, N=3, N=4, N=5 and C(N) is dimensioned to (P,7).

I ended up copying each matrix onto the new matrix (Q) using five FOR...NEXT loops then doing a GOSUB to the sorting operation. This saved perhaps 30 steps of code but increased the processing time by

requiring the duplication of each matrix C1, C2...C5 to matrix Q. (See example 2.)

Is there a more efficient way of doing such recurring operations to different matrices without simply repeating the same routine over and over with different variables?

The digit following an uppercase letter in a symbol used to represent numeric constants, e.g., the 1 in C1, cannot be a variable.

```

1470 REM *** SORT CONTENTS OF 'C3'
1480 F=0
1490 FOR I=1 TO D3-1
1500 IF C3(I,5) < C3(I+1,5) THEN 1570
1510 FOR C=1 TO 7
1520 B1(C)=C3(I,C)
1530 F=1
1540 C3(I,C)=C3(I+1,C)
1550 C3(I+1,C)=B1(C)
1560 NEXT C
1570 IF C3(I,5) < C3(I+1,5) THEN 1650
1580 IF C3(I,3) < C3(I+1,3) THEN 1650
1590 FOR C=1 TO 7
1600 B1(C)=C3(I,C)
1610 F=1
1620 C3(I,C)=C3(I+1,C)
1630 C3(I+1,C)=B1(C)
1640 NEXT C
1650 NEXT I
1660 IF F=1 THEN 1480
1670 REM *** SORT CONTENTS OF 'C4'
1680 F=0
1690 FOR I=1 TO D4-1
1700 IF C4(I,5) < C4(I+1,5) THEN 1770
1710 FOR C=1 TO 7
1720 B1(C)=C4(I,C)
1730 F=1
1740 C4(I,C)=C4(I+1,C)
1750 C4(I+1,C)=B1(C)
1760 NEXT C
1770 IF C4(I,5) < C4(I+1,5) THEN 1850
1780 IF C4(I,3) < C4(I+1,3) THEN 1850
1790 FOR C=1 TO 7
1800 B1(C)=C4(I,C)
1810 F=1
1820 C4(I,C)=C4(I+1,C)
1830 C4(I+1,C)=B1(C)
1840 NEXT C
1850 NEXT I
1860 IF F=1 THEN 1680
1870
1880
1890
1900
1910
1920
1930
1940
1950
1960
1970
2150 FIND 7
2160 WRITE @33:C1
2170 FIND 9
2180 WRITE @33:C2
2190 FIND 11
2200 WRITE @33:C3
2210 FIND 13
2220 WRITE @33:C4
2230 FIND 15
2240 WRITE @33:C5

```

Example 1. Each matrix is sorted by separate lines of code.

Your local Tektronix Sales Engineers can provide you with more details about the capabilities of the 492P and 496P Programmable Spectrum Analyzers. 

Editor's Note: A program recently included in the 4050 Series Applications Library allows a 492P or 496P programmable spectrum analyzer user to functionally verify the IEEE-488 interface in that instrument. See Program 10 of TEKniques Vol. 6, No. 3 T1 in the New Abstracts section of this issue of TEKniques.

But, an element-by-element transfer of matrices can be quickly done by:

$$Q = C1$$

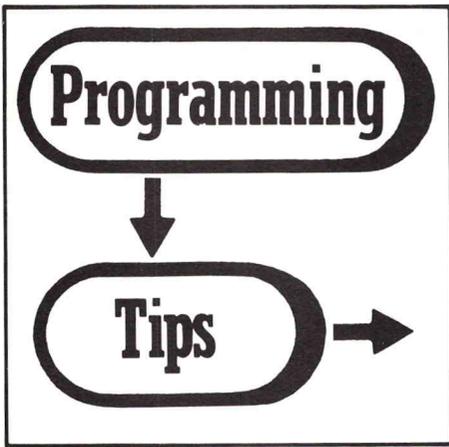
eliminating the FOR...NEXT loop processing time. The matrices (or arrays) must be of the same dimension, which you did in your second example through lines 1960 and the lines preceding each matrix transfer (1400 and 1510 in the second example). 

```

1380 REM *** SORT CONTENTS OF 'C3'
1390 P=0
1400 DIM B(P,7)
1410 FOR X=1 TO P
1420 FOR C=1 TO 7
1430 Q(X,C)=C3(X,C)
1440 NEXT C
1450 NEXT X
1460 GOSUB 1740
1470 FIND 11
1480 GOSUB 1940
1490 REM *** SORT CONTENTS OF 'C4'
1500 P=0
1510 DIM Q(P,7)
1520 FOR X=1 TO P
1530 FOR C=1 TO 7
1540 Q(X,C)=C4(X,C)
1550 NEXT C
1560 NEXT X
1570 GOSUB 1740
1580 FIND 13
1590 GOSUB 1940
1600
1610
1620
1630
1640
1650
1660
1670
1680
1690
1700
1710
1720
1730
1740 F=0
1750 FOR I=1 TO P-1
1760 IF Q(I,5) < Q(I+1,5) THEN 1830
1770 FOR C=1 TO 7
1780 B1(C)=Q(I,C)
1790 F=1
1800 Q(I,C)=Q(I+1,C)
1810 Q(I+1,C)=B1(C)
1820 NEXT C
1830 IF Q(I,5) < Q(I+1,5) THEN 1910
1840 IF Q(I,3) < Q(I+1,3) THEN 1910
1850 FOR C=1 TO 7
1860 B1(C)=Q(I,C)
1870 F=1
1880 Q(I,C)=Q(I+1,C)
1890 Q(I+1,C)=B1(C)
1900 NEXT C
1910 NEXT I
1920 IF F=1 THEN 1740
1930 RETURN
1940 WRITE @33:Q
1950 CLOSE
1960 DELETE Q
1970 RETURN

```

Example 2. FOR...NEXT loop transfers selected matrix to Q to permit GOSUB for sorting operation.



Translating 4050 Tape Files to 4909 Disk Files

by Tony Freixas
Tektronix, Inc.
Wilsonville, OR

To convert programs that run on a 4050 tape to run on the 4909 File Manager, the input/output commands must be changed. The following series of translations on tape input/output commands and their arguments will work most of the time. However, some judgment is needed to determine exactly what needs to be changed; the translations can't be done automatically. Let's look at what the translations are trying to do.

File Renaming

On the tape are files numbered 1, 2, 3, etc. Using the translations, the 4909 would have a series of files called T1, T2, T3, etc., where T1 would correspond to file 1 on the tape. So wherever anything was done to file 1 in the tape version of the program, the same operation would be performed on file T1 on the 4909 in the converted program.

If two tapes of programs are to be converted, however, a different method of renaming is required since two files can't both be file T1 or T2, etc. One solution is to group the files into libraries by changing the translations here, or group them into separate user directories. Call them all T1, T2, and so forth, but change user directories when moving from one set of tape programs to another.

Translation Terminology

In the translation table, when the tape version says FIND <n1>, <n1> is an expression meaning the first argument for that block of commands. It could be (depending on the command) a file number, device number, line number, etc. The second argument is defined as <n2>, and so on.

The brackets [] mean the argument is optional just as they do in all of Tek's documentation.

Translation Grouping

Operations that usually take place as a unit are grouped together in the translation series. This is exactly what makes translation under program control very hard; depending on the context you may not be looking at just one statement, but at several. For instance, normally when you want to APPEND some-

thing, you FIND the file then do the APPEND. But a program may have the FIND statement several statements removed from the APPEND. Or a program may FIND a file, then do an APPEND three times. This is where judgment becomes involved.

The commands are in alphabetical order according to the function being performed.

Translation Comments

In the first block (APPEND), <n1> represents the file number. The translation takes

APPEND

```
Tape: FIND <n1>
      APPEND [@<n2>:] <n3> [, <n4>]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      APPEND N$, "A"; <n3> [, <n4>]
```

BAPPEN

```
Tape: FIND <n1>
      CALL "BAPPEN", [<n2>:] <n3> [, <n4>]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      APPEND N$; <n3> [, <n4>]
```

BOLD

```
Tape: FIND <n1>
      CALL "BOLD" [, <n2>]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      OLD N$
```

BSAVE

```
Tape: FIND <n1>
      CALL "BSAVE" [, <n2>]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      SAVE N$
```

CLOSE

```
Tape: CLOSE [0]
4909: CLOSE [1]
```

FIND

```
Tape: Find <n1> (if to be followed by READ's,
                WRITE's, PRINT's or INPUT's)
4909: CLOSE 1
      CALL "OPEN", "T"; <n1>, "LFN:1"
```

INPUT

```
Tape: INPUT @<n1>: ...
4909: INPUT #1: ...
```

KILL

```
Tape: KILL [@<n1>:] <n2>
4909: CALL "DELETE", "T"; <n2>, "NINQUIRE"
```

LINK

```
Tape: CALL "LINK", [<n1>:] <n2>
4909: —
```

MARK

```
Tape: FIND <n1>
      MARK [@<n2>:] <n3>, <n4>
4909: FOR I = <n1> to <n1> + <n3>
      CALL "CREATE", "{B}", "D", "T"; I
      NEXT I
```

MTPACK

```
Tape: CALL "MTPACK"
4909: REM
```

OFF EOF

```
Tape: OFF EOF (0)
4909: OFF EOF (1)
```

ON EOF

```
Tape: ON EOF (0) THEN ...
4909: ON EOF (1) THEN ...
```

OLD

```
Tape: FIND <n1>
      OLD [@<n2>:]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      OLD N$, "A"
```

PRINT

```
Tape: PRINT @<n1>: ...
4909: PRINT #1: ...
```

READ

```
Tape: READ @<n1>: ...
4909: READ #1: ...
```

SAVE

```
Tape: FIND <n1>
      SAVE [@<n2>:] [<n3> [, <n4>]]
4909: N$ = STR (<n1>)
      N$ = REP ("T", 1, 1)
      SAVE N$, "A" [:<n3> [, <n4>]]
```

TLIST

```
Tape: TLIST
4909: DIR 0, "T*" ← not exact translation"
```

TYP

```
Tape: TYP (0)
4909: TYP (1)
```

WRITE

```
Tape: WRITE @<n1>: ...
4909: WRITE #1: ...
```

the number, converts it to a string, adds a T in front of it and then performs a 4909 APPEND operation. The second argument <n2> is unnecessary in the 4909 command and <n3> and <n4> don't need to be changed.

In the second block, the Binary APPEND is translated similarly, as are Binary OLD and Binary SAVE.

The CLOSE is the same except that logical file number 1 is used in place of the tape's default of 0. You can also leave it out in the translation. CLOSE without an argument means CLOSE all.

If you have just a FIND operation by itself, you can think of it as being equivalent to an OPEN. In some cases where you have a FIND followed by another tape command (such as MARK or OLD), this may not be the translation you want. Again, you have to know the context.

An INPUT statement is pretty much the same. Simply translate the @33, which is the tape device address, to #1.

KILLing a file on tape is DELETE on the disk.

There is no 4909 equivalent to LINK. Not too many programs use this command but it does allow you to old in another program and keep the variables around.

MARKing a tape is just the same as CREATing a file.

The MTPACK has no equivalent on disk since it rewinds the tape to its end. A possible translation might be just REM.

The OFF EOF and ON EOF arguments just get changed from 0 to 1.

An OLD operation is just an OLD to the specific file desired.

The closest thing to TLIST is DIRectory. This is normally done in immediate mode so the fact that the information returned is slightly different probably won't hurt. If a program does depend on a TLIST command and it knows it's so many columns, that could involve some changes.

For the most part, the translations are pretty much just straightforward substitutions with the exception of combinations such as FIND/APPEND, etc., where the context must be considered.

Uppercase/Lowercase Keywords in Listings

by Ron Boerger
Dept. of the Air Force
San Antonio, TX

Do you want listings in which keywords are printed both in uppercase, and lowercase (i.e., 'Print X' instead of 'PRINT X')? By using an undocumented feature on the 4052 or 4054, you may do just that.

On pages 2-9 of the '4050 Series Reference Manual', the several system FONTS are displayed. FONT 7 is described as 'reserved'

and 'same as font 0'. While all of the letters are the same, FONT 7 has one unusual feature; when LISTing a program, keywords are printed in uppercase and lowercase. Such listings are often easier to read than those with keywords totally in caps (see example).

```
list
130 REM* This program PRINTs a list of integers 1-100 and their squares.
140 FOR X=1 TO 100
150 PRINT X;X^2
160 NEXT X
170 END

font 7

list
130 Rem* This program PRINTs a list of integers 1-100 and their squares.
140 For X=1 To 100
150 Print X;X^2
160 Next X
170 End
```

Defusing Dangerous UDKs

by Duke Ganote
Institut fuer Klinische
Pharmakologie
Bobenheim-am-Berg, FR Germany

There is the possibility of destroying data or missetting parameters when an out-of-sequence UDK is pressed. Extending Jon C. Mutton's programming tip on program initialization (Vol. 4, No. 6, Sept. 15, 1980) allows us to ensure proper UDK sequencing.

In the following program the UDKs have a hierarchy. UDKs 1 or 2 may be pressed anytime. UDK 3 may be used only after UDKs 1 or 2. UDKs 4 or 5 may be used only after UDK 3 or each other. If a UDK is pressed out-of-sequence, lines 85 through 91 will "disarm" it.

This, of course, aids in making the program "fool-resistant." But remember: nothing is fool-proof, because fools are so ingenious!

```
4 REM ***** UDK 1 ***** INPUT DATA FROM SCREEN
5 GO TO 100
9 REM ***** UDK 2 ***** DATA FROM TAPE
9 GO TO 200
12 REM ***** UDK 3 ***** DATA ANALYSIS
13 GOSUB 86
14 GO TO 300
16 REM ***** UDK 4 ***** PLOT ANALYSIS
17 GOSUB 88
18 GO TO 400
20 REM ***** UDK 5 ***** SAVE DATA & ANALYSIS TO TAPE
21 GOSUB 88
22 GO TO 500
85 REM ***** DEFUSE UDKs ? *****
86 N=REP('',1,0)
87 GO TO (N+1)*41 OF 90,91
88 N=REP('',1,0)
89 GO TO (N+2)*41 OF 90,91
90 END
91 RETURN
100 REM ***** INPUT DATA *****
110 PRINT 'INPUT'
120 REM
130 N="1"
140 END
200 REM ***** DATA FROM TAPE *****
210 PRINT 'TAPE'
220 REM
230 N="1"
240 END
300 REM ***** ANALYSIS *****
310 PRINT 'ANALYSIS'
320 REM
330 N="2"
340 END
400 REM ***** PLOT *****
410 PRINT 'PLOT'
420 REM
430 END
500 REM ***** SAVE ON TAPE *****
510 PRINT 'SAVE'
520 REM
530 END
```

Increasing Throughput to the 4663 Plotter

by **Frits Handgraaf**
Tektronix, Inc.
Amstelveen, The Netherlands

When using the GPIB, if you combine the programmable macros function with the DAB command mode, you can reduce communications overhead and speed throughput to your 4663 Plotter.

DAB Operating Mode

Before discussing an application of the programmable macros function, let's first review the DAB operating mode of the 4663 (DAB = DA**TA** Byte). By driving the plotter in DAB mode, you can reduce the number of addressing sequences (the number of times the ATN line of the GPIB is asserted). For example, you can draw a box by executing one program line containing the statement:

```
PRINT @1,32:A$
```

where A\$ = "M0,0D10,0D10,5D0,5D0,0" instead of executing the following program:

```
MOVE @1:0,0
DRAW @1:10,0
DRAW @1:10,5
DRAW @1:0,5
DRAW @1:0,0
```

Using the DAB method, blocks of data and commands can be built up and stored in a string with a size equal to or smaller than the number of bytes free in the plotter input buffer. This string is then sent to the plotter. When the input buffer fills, the plotter will hold off new data and execute commands from its buffer until enough space in RAM becomes available to contain new data. During this time a new block of data and commands can be built up in the 4050. From the 4050's point of view the communication will end after the last string is sent. This increases throughput by reducing communication overhead.

The method has the following restrictions:

1. The command string size should not exceed the available size of the input buffer of the plotter (RAM).
2. Commands with more than one argument need a separator between the arguments (a blank or comma, i.e.: " " or ",").
3. A "print character" command has to be terminated once all characters to be printed are defined. Therefore, if a PRINT command and the print string are included in a command string, the PRINT command and string have to be terminated by an ETX (Control-C). If this ter-

minator is omitted and the "print character" command (and the print string) is followed by a graphic command, the graphic command will be printed as a literal rather than executed. For example:

```
PRINT @1,32:"P this is a 4663M0,0D130,0" ECRJ
```

will give the result that everything after the "print character" (command P) until the end of the command string (EOI or CR) will be printed as literals. To avoid this, insert an ETX:

```
PRINT @1,32:"P this is a 4663M0,0D130,0" ECRJ
```

The table illustrates when it is necessary for ETX to terminate the print string:

	Print string is last part of command string.	Print string is not last part of command string.
Controller gives EOI at end of command string.	No ETX necessary (terminated by EOI).	ETX
No EOI	ETX	ETX

A 4662 Plotter can be used in the same way just described. To operate the 4662 in DAB mode, switch B has to be set on the appropriate selection.

DAB and Programmable Macros

Besides allowing generation of either a circle or an arc using one command, the 4663S Plotter or 4663 Plotter with Option 31 allow macros (a block containing a sequence of commands and data arguments) to be defined and stored in RAM. The benefit is clear: a sequence of plotter commands and/or data can be executed/plotted from one command ("expand macro") at different program locations without having to send the entire sequence over the interface line each time.

Thus, in a drafting or design application, symbols or sequences of the same commands and data can be defined as macros. A command string may then combine move commands and expand-macro commands to get a plot.

In the following program, a drawing created with a 4050 is plotted on a 4663S or 4663 with Opt. 31. The drawing contains a number of symbols. It is assumed the symbols used have been previously defined as macros in the 4663. The contents of the drawing are stored in a so-called "display list," which in this case is an array O, dimensioned with N1 rows and four columns. N1 is the total number of symbols in the drawing. The columns in array O contain the following information:

```
2870 REM ACTUAL PLOTTING ROUTINE
2880 PRINT @02,32:"U0";
2890 INPUT @02,32:P,P,I,P
2900 DELETE T$
2910 DIM T$(P1+0*MEMORY)
2920 T$=""
2930 T$=T$$"A00 100 0 90"
2940 T$=T$$"A00 120 0 100"
2950 T$=T$$"BP2CBBP1"
2960 FOR I=1 TO N1
2970 S$=STR(-O(I,4))
2980 R$=REP(" ",1,1)
2990 P$="AR"&S$
3000 R$=STR(O(I,4))
3010 R$=REP(" ",1,1)
3020 R$="AR"&R$
3030 P$=STR(O(I,2))
3040 P$=REP(" ",1,1)
3050 Q$=STR(O(I,3))
3060 P$=P$&Q$
3070 P$="M"&P$
3080 P$=P$&R$
3090 R$=STR(O(I,1))
3100 R$=REP(" ",1,1)
3110 R$="AE"&R$
3120 P$=P$&R$
3130 P$=P$&S$
3140 T$=T$&P$
3150 IF LEN(T$)<P1-100 AND NOT(I=N1) THEN 3100
3160 PRINT @02,32:T$;
3170 T$=""
3180 NEXT I
3190 PRINT @02,32:"AI";
3200 REM RETURN TO MAIN MENU
3210 RETURN
```

- O(I,1) = F1 where F1 is the symbol number
- O(I,2) = X where X is the X-position of the symbol's origin in the drawing.
- O(I,3) = Y where Y is the Y-position of the symbol's origin in the drawing.
- O(I,4) = R where R is the rotation of the symbol with respect to the origin point.

I is the symbol counter, with 1 = <I <= N1,

In line 2880 and 2890, a "read status register zero" command is executed to check the size of RAM left (remember: defined macro's occupy memory of the input buffer). The command string T\$ is dimensioned accordingly in line 2910.

Lines 2930 and 2940 specify the plotter window and viewport to be the same as those of the 4050.

Line 2950 contains commands to outline the viewport using pen 2 and to draw the following graphic information (macro's) using pen 1.

In the FOR-NEXT loop, the display list is emptied to the plotter: S\$ contains the negative rotation of the symbol, R\$ contains the rotation angle and the symbol number respectively, P\$ contains the X coordinate, and Q\$ contains the Y coordinate of the origin of the symbol in the drawing. Note that in line 3050 the leading blank in Q\$ is not replaced: this blank is used as separator between the X and Y coordinates. In line 3070 the X, Y coordinates are preceded by a M(ove) command. In line 3110 the "expand macro" command is added to the com-

mand string. Notice that the macro number is the same as the symbol number.

For each symbol of the drawing, the command string T\$ will contain the following sequence:

“. . .Mx yARdegreesAESymbolnumberAR-degrees. . .”

or: Move to symbol location, set plotter rotation to R degrees, expand macro F1, set plotter rotation back to zero.*

This sequence is followed by a sequence for the next symbol (line 3180) appended to the

command string T\$. If the length of the command string is smaller than the available buffer size minus 100, a next sequence is appended to the command string. The second part of 3150 assures that the last part of the command string is sent to the plotter as well. In line 3190 the pen carriage is moved to the load point once the plot is finished.

* Note: Each additional “set plotter rotation” command increments the absolute angle to the sum of the previous defined angles.

Print with ETX/ACK

by **Boyd Reasor**
Campbell, CA

By selecting an Alternate Delimiter for the INPUT operation, handshaking is possible with most I/O devices capable of ETX/ACK protocol. The result is faster output. For example, data transfer from the Tektronix 4051 to the XEROX 1700 terminal is increased up to 33%. This increase in speed is accomplished by sending data to the 1700 faster than it can print the data. The 1700 buffer will never completely empty, insuring that the print mechanism will always run at top speed as it tries to empty the buffer.

Using conventional methods (without handshaking), transmitting data at 1200 baud (120 cps) into a 158 character buffer that empties at 45 cps will quickly overrun the buffer. This condition can be handled using two methods: First, lower the baud rate to 300 and let the 1700 WAIT for the 4051. This method forces the 1700 to be idle 33 percent of the time. Second, retain the 1200 baud rate but package the data into sizes limited only by the buffer. In this case, the buffer status must be monitored so that no data transmission occurs until space is available in the buffer for a complete package.

Monitoring buffer status is simply done by tagging an end-of-text (ETX) character to each data package. This character (control-C) is unique to the RS-232 handshaking. When ETX moves to position one in the buffer, the 1700 acknowledges by sending an ACK (control F) back to the 4051 signaling for another load. To allow the 4051 to recognize the single ACK character, the al-

ternate delimiter must initially be set. (See statement 150.) This method almost optimizes the data exchange. The following simple program illustrates the handshaking technique:

100 CALL "RATE",1200,2,2	Initialize interface.
110 CALL "CMSET"	
120 DIM A\$(158),B\$(2)	Initialize variables.
130 B\$=CHR(13)	Set CR string.
140 B\$=B\$&"C"	Add ETX to CR string.
150 PRINT @37,0:6,255,255	Set alternate delimiter to ACK.
160 FIND 3	Position tape for input.
170 ON EOF (0) THEN 240	Set EOF flag.
200 INPUT @33:A\$	Get one buffer load from tape.
210 PRINT @40:A\$;B\$;	Send data to 1700.
220 INPUT @40:A\$	Wait for ACK from 1700.
230 GO TO 200	Next buffer load.
240 END	

One data package in the 1700:

```
-----
a a a a a M C
-----
```

a a a a a = data record
M = carriage return (CR)
C = end-of-text (ETX)

Although the data is being sent at 1200 baud, the printing is not continuous. A short wait in printing occurs due to retrieving the next data record off tape and transmitting it to the 1700. By stacking two records (each with an ACK) in the 1700, the buffer will never completely empty, therefore optimizing printing. Adding two

statements will accomplish the desired optimization.

```
180 INPUT @33:A$      Get one buffer
                      load from tape.
190 PRINT @40:A$;B$; Send data to
                      1700.
```

Two data packages in the 1700 buffer:

```
-----
a a a a a M C b b b b b M C
-----
```

Notice that the ACK from the first ETX, not the second ETX, causes the next data record to be placed in the buffer.

Decoding System Error Messages in the 4052 and 4054

by Steve Duncan

Tektronix, Inc.
Wilsonville, OR

When the 4052 or 4054 System discovers an error condition for which it has no conventional error message, a SYSTEM ERROR is generated, followed by a power-up restart. This puts the machine into a known state and clears the malfunction.

Since memory is cleared, there are no clues left to which operation caused the failure. Two parts of the System Error message may be able to help. Let's take a look at an error message and decode its seventh and eighth elements.

SYSTEM ERROR - FFFF 49 D2 D4 DF42
018F A0 01A4 6A

A0 . . . Bank Switch Information
01A4 . . BASIC Program Line in Hex
Form

The Bank Switch digits tell which ROM pack slot was being accessed at the time of failure. The following cross reference matches the hex digits with the corresponding ROM slot. If no ROM pack is enabled, 00 indicates the system ROMs were in use.

ROM Slot Decoding Table (in hex) for the System Error Message

System ROM: 00
Slot 41: A0
Slot 51: A8
Slot 61: 80
Slot 71: 88

4050E01 in Slot #41: B0-B8
 in Slot #51: B8-BF
 in Slot #61: 90-97
 in Slot #71: 98-9F

The four hex digits identifying the BASIC line number must be converted to decimal. For convenience, the following BASIC program will do the job.

No return is needed after inputting each hex digit since the POINTER is used for input. The program accepts uppercase or lowercase A-F but doesn't check for invalid input. The line number is printed out at the current position of the pointer.

```
100 PRINT "LINPUT HEX DIGITS, ";
110 PRINT "FROM LEFT TO RIGHT"
120 N=0
130 FOR I=3 TO 0 STEP -1
140 POINTER X,Y,Z$
150 Z=ASC(Z$)
160 IF Z<58 THEN 190
170 Z=Z-55-(Z>96)*32
180 GO TO 200
190 Z=Z-48
200 N=N+16*I*Z
210 NEXT I
220 PRINT "BASIC LINE #";N
```

Looking at our example, we see the ROM bank enabled was hex A0, or slot 41. The BASIC program line was 01A4, or $(0*16^3) + (1*16^2) + (A*16^1) + (4*16^0)$, or 420 decimal. We now know the error occurred in line 420 and involved the ROM pack in slot 41.

Graphics Input on 4052A Without a Joystick

by Terry Hamm

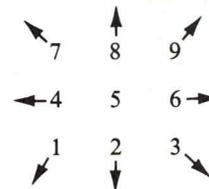
Tektronix, Inc.
Wilsonville, OR

The following program shows a method for graphics input on a 4052A by using the numeric pad to move the selection cursor around the screen instead of a joystick. The number pad is used in the standard way a key pad would be used to move the cursor.

The graphics input is modeled after POINTER and is written as a BASIC subprogram with three parameters, X, Y, and Z\$. All three return the values that POINTER would return, except that the numbers 1,2,3,4,6,7,8,9 do not terminate the gin, they move the cursor.

Notice also the use of new 4052A/54A BASIC features like DO...LOOP, multi-character identifiers, IF..THEN, UBOUND, CCINPUT and comment tails. CCINPUT is the statement that allows this capability in the 4052A. It takes one character at a time from the input buffer without printing it.

The cursor can move on the X axis, Y axis or on either 45 degree angle. The keys move the cursor in the diagrammed fashion.



A program uses the gin facility the same way it uses POINTER except with a call.

POINTER X,Y,Z\$ → CALL _POINTER(X,Y,Z\$)

The subprogram operates in GDU's and is not cognizant of the current window or viewport. To get more precision, change lines 360 and 370.

```
100 SUB _pointer(X,Y,Z$) !Simulate GIN for 4052A
110 LOCAL _cursor$,Show_cursor
120 _cursor$=CHR(127) !The cursor to display
130 Show_cursor=1 ! True
140 IF NOT(UBOUND(X,1)*UBOUND(Y,1)) THEN !one or both are undefined
150 X=65
160 Y=50
170 END IF
180 DO
190 DO
200 IF Show_cursor THEN ! Display it
210 PRINT @32,21:X,Y ! At this location
220 PRINT @32,24:_cursor$ ! In refresh
230 END IF
240 CCINPUT Z$ ! Simulates GIN without printing character
250 EXIT IF LEN(Z$) ! If there is a character
260 Show_cursor=1 !true
270 LOOP !try for a character
280 EXIT IF Z$<"1" OR Z$>"9" OR Z$="5" !Numeric keys move cursor
290 Show_cursor=0 !false
300 REM Following translates cursor location based on numeric keys
310 X=X+(ASC(Z$)-49 MOD 3)-1 ! The X coordinate
320 Y=Y+INT((ASC(Z$)-49)/3)-1 !The Y coordinate
330 LOOP
340 END SUB
```

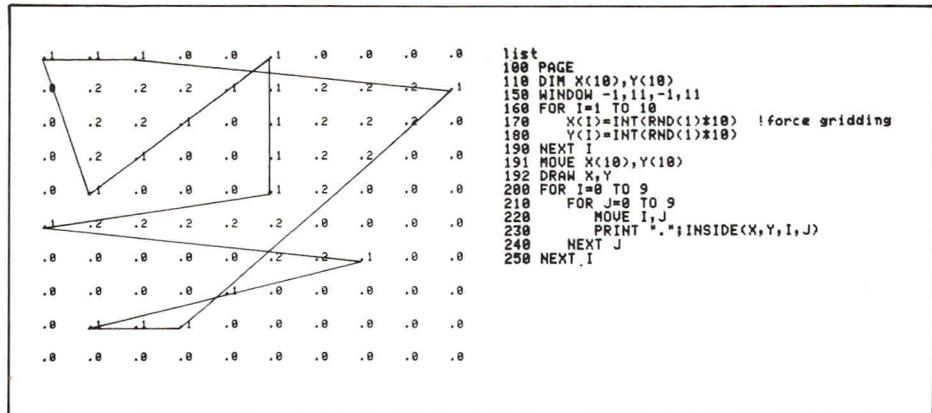
4052A/4054A INSIDE Function for Polygons

by Ed Post

Tektronix, Inc.
Wilsonville, OR

In regard to "Polygons — Do Points Lie Within" by Ir. A.C. Visser in the Spring, 1982 issue of *TEKniques*: the technique of slightly displacing the X' coordinate to handle the case when a point in the polygon has the same X value as the X coordinate of point P works only if the polygon is defined at a resolution less than that of the floating point numbers. A better technique is to check for valid intersections only lines whose lowest X coordinate is less than P.X, and whose highest X coordinate is greater than OR EQUAL TO P.X. This is part of the strategy used in the new INSIDE function in the 4052A/4054A.

An additional feature of the INSIDE function is that it gives a different value if the point is ON a line segment defining the polygon, to within FUZZ. The example program demonstrates this capability. The algorithm used in the INSIDE function is also included, to show the internal use of fuzzy comparisons.



Algorithm for INSIDE in the 4052A/4054A

1. INSIDE=0
2. Loop (I) over segments in the polygon.
 - a. If $X[I]=X[I+1]$ then $IntY=Py$
 - else $IntY=Intersect(Y[I],X[I],Y[I+1],X[I+1],Px)$
 - b. If $Y[I]=Y[I+1]$ then $IntX=Px$
 - else $IntX=Intersect(X[I],Y[I],X[I+1],Y[I+1],Py)$
 - c. If $Px EQ IntX$ or $Py EQ IntY$ then
 - If $Px LE (X[I] MAX X[I+1])$ and $Px GE (X[I] MIN X[I+1])$ and $Py LE (Y[I] MAX Y[I+1])$ and $Py GE (Y[I] MIN Y[I+1])$ then
 1. INSIDE=1
 11. goto step 3.
 - d. If $Py > (Y[I] MIN Y[I+1])$ and $Py <= (Y[I] MAX Y[I+1])$ then
 - If $IntX < Px$ then $INSIDE=2-INSIDE$.
 - e. Next I
3. Return $FLOAT(INSIDE)$.

notes:

- 1) EQ,LE,GE are the fuzzy compare operators. >, <, = are exact compares.
- 2) $X[I+1]$ and $Y[I+1]$ wrap around -- that is, if $I=N$ then $X[I+1]$ is $X[1]$.
- 3) $Intersect(X1,Y1,X2,Y2,Yint)=X1 + (YINT-Y1)*(X2-X1)/(Y2-Y1)$

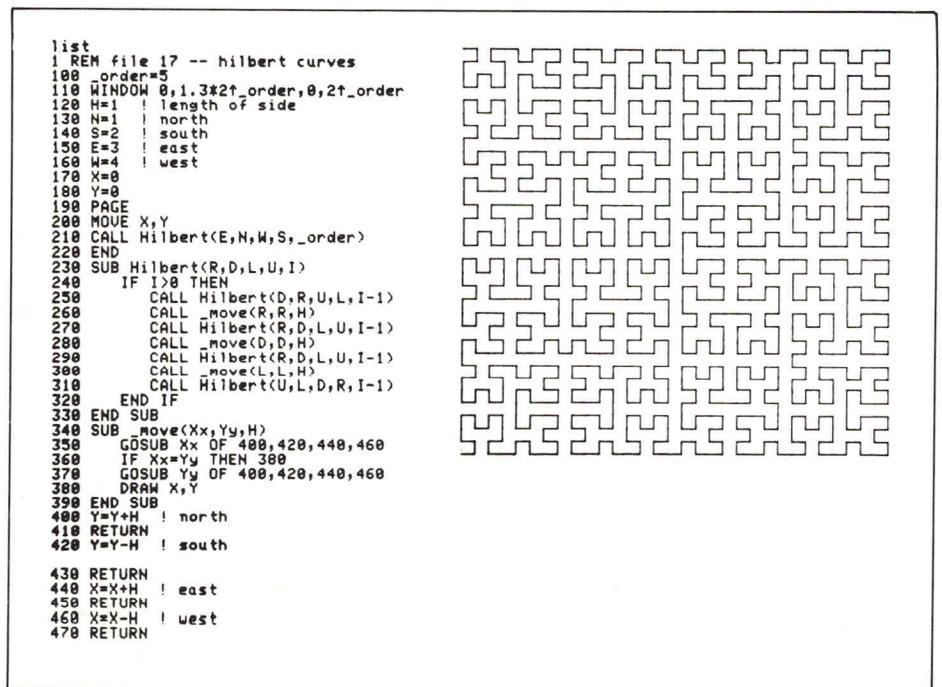
Graphics and Recursive Functions in 4052A/4054A

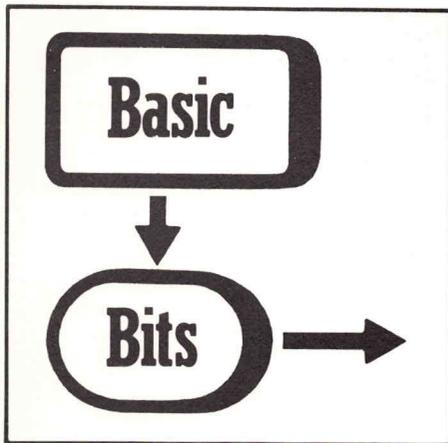
by Ed Post

Tektronix, Inc.
Wilsonville, OR

The new subroutine capabilities of the 4052A/4054A make possible some interesting interactions between graphics and recursive functions. In particular, the class of objects called "space filling curves" are typically defined in a recursive manner.

This program draws one type of space filling curve called a "Hilbert curve." The variable "__order" controls the depth of recursion. As __order increases, the curve gets more complex and "covers" more of the plane. 





UDK Versatility

by D.F. Capellaro
Imperial Cancer Research
Fund Laboratories
London, England

UDK Toggle

When writing a program which ended up having two distinct phases, a data collection phase and a data analysis phase, I rapidly ran out of UDKs. So I designated UDK 20 as a "toggle" to change the function of each of the other UDKs thus:

```
80 REM --TOGGLE FUNCTION--
81 IF X=1 THEN 84
82 X=1
83 END
84 X=2
85 END
```

Lines 83 and 85 could be GO TO instructions, if appropriate.

Now each UDK has two functions, for example:

```
4 REM --DATA COLLECTION(X=1); DATA ANALYSIS(X=2)
5 GO TO X OF 1000,2000
```

UDK 20 Advantage

Don't overlook the advantage that UDK 20 has over all the other UDKs. The series of program statements which it initiates can be as long as you like, although it is wise to keep the line numbers less than 99 to prevent problems with the default renumbering routines.

Subroutine Renumbering

Often I end up with many subroutines, each of which begins at n000 where n is from 1 on up. Sometimes when amending a program, I run out of space between existing lines and must renumber to get additional space. Consequently, one of my subroutines is a series of renumbering statements:

```
30000 RENUMBER 1000,10,1000
30010 RENUMBER 2000,10,2000
30020 RENUMBER 3000,10,3000
|
|
|
30250 RENUMBER 30000,10,30000
30260 END
```

If a UDK is dedicated to renumbering such as

```
72 REM -- RENUMBER --
73 GO TO 30000
```

then using this key will renumber all the subroutines without losing their basic location within the overall program. If a default RENumber command is called, then the last number in all the RENumber steps changes, as well as each line number, but line 73 also changes to allow a full renumbering sequence to be effected.

UDK Remarks

A final tip is quite trivial but I've found it very useful. Each UDK should start with a remark so that calling a LIST 1-99 gives a complete breakdown of the total program. It's much more convenient than a full list when the program has 6 or 700 lines! You'll notice this technique has been used in all the examples.

Update to Tape Index Tip

by Dave Quirt
Saskatchewan Research Council
Saskatoon, Sask., Canada

Three corrections need to be made to the "Tape Index" programming tip in *Basic Bits* published in *TEKniques Vol. 5 No. 4*.

Line 160 should read: 160 FOR I=1 TO 10
Line 210 should read: 210 FIND I+25*(I0-1)
Line 470 should read: 470 NEXT I0

Also, by adding the following lines of code, data file titles can be displayed provided the title is the first data item in the file, and binary program titles can be displayed as

long as the word 'BINARY' is the last word in the title line. A maximum of 250 files may be indexed, but this can be increased or decreased by changing line 160.

Editor's Note: The above corrected statements were inadvertently copied incorrectly when generating a legible hard copy. We apologize to the original author, Ron Robinder.

```
1 REM THIS IS A FILE OF BINARY
4 RUN 100
100 INIT
110 PRINT "INPUT NUMBER FOR OUTPUT DEVICE (32=SCREEN): ";
120 INPUT D
130 I$=" "
140 DIM A$(50)
150 FOR J=1 TO 1
160 FOR I0=1 TO 10
170 PAGE 0D:
180 PRINT 0D:"INDEX OF PROGRAMS ON TAPE,"
190 PRINT 0D:"FILE EXTENDED TITLE(LINE 1)"
200 FOR I=1 TO 25
205 PRINT 033,0;0,0,1
210 FIND I+25*(I0-1)
220 INPUT 033:I$
225 PRINT 033,0;0,0,0
227 FIND I+25*(I0-1)
230 H$=SEG(I$,9,1)
240 IF H$="A" THEN 280
250 IF H$="B" THEN 391
260 IF H$="L" THEN 480
270 IF H$="H" THEN 420
280 H$=SEG(I$,17,1)
290 IF H$="P" THEN 350
300 IF H$(">"T" AND H$(">"D" THEN 330
301 IF H$(">"D" THEN 310
302 INPUT 033:A$
303 A$=A$% " DATA"
304 GO TO 380
310 PRINT 0D:I+25*(I0-1),"TEXT FILE"
320 GO TO 450
330 PRINT 0D:"ERROR IN FILE ";I+25*(I0-1)
340 GO TO 450

350 INPUT 033:A$
360 A=POS(A$,"REM ",1)

370 A$=SEG(A$,A+4,LEN(A$))
380 PRINT 0D:I+25*(I0-1),A$
390 GO TO 450
391 H$=SEG(I$,17,1)
392 IF H$="P" THEN 398
393 IF H$(">"T" AND H$(">"D" THEN 330
394 IF H$(">"D" THEN 310
395 READ 033:A$
396 A$=A$% " BINARY DATA"
397 GO TO 380
398 READ 033:A$
399 A2=POS(A$,"BINARY",14)
400 A$=SEG(A$,13,A2-7)
405 GO TO 380
410 GO TO 450
420 H$=SEG(I$,35,6)
430 H=VAL(H$)
440 PRINT 0D:I+25*(I0-1),"NEW FILE; LENGTH = ";H;" BYTES"
450 NEXT I
460 INPUT I$
470 NEXT I0
475 GO TO 560
480 PRINT 0D:I+25*(I0-1),"LAST FILE"
490 PRINT "-----"
500 PRINT "INPUT FILE YOU WANT ";
510 INPUT F
520 IF F=0 THEN 550
530 FIND F
533 PRINT "IF FILE IS BINARY, INPUT 'B'G, ";
534 PRINT "OTHERWISE, PRESS [CR]";
535 INPUT B$
537 IF B$="B" THEN 550
540 OLD

550 CALL "BOLD"
560 PAGE 0D:
570 PRINT "MAXIMUM INDEX CAPACITY (250 FILES) EXCEEDED"
580 END
```

IDD Applications Library Programs

Ordering

4050 Series

The 4050 Series programs included in the Applications Library prior to September 1981 are packaged and nomenclated by function. Those programs accepted into the Library after September 1981 are packaged and nomenclated with the Volume and Number of the corresponding issue of *TEKniques* in which the package was announced.

Each package includes the source code on tape or disk (T=tape; D=disk) together with the supporting documentation; listings are not

included in the documentation. Documentation may be purchased separately.

The 4050 Series Applications Library Programs catalog contains the abstracts describing the programs in each package along with representative output in most cases. The catalog part number is 062-6343-00.

Package Title	Documentation Part #	Package Part #
Business Aids T1	062-5987-00	062-5987-01
Business Aids T2	062-5988-00	062-5988-01
CAD T1	062-5976-00	062-5976-01
CAD D1	062-5977-00	062-5977-01
Character Generator T1	062-5951-00	062-5951-01
Education/Research T1	062-5982-00	062-5982-01
Education/Research T2	062-5983-00	062-5983-01
Electrical Engineering T1	062-5978-00	062-5978-01
Graphing T1	062-5964-00	062-5964-01
Graphing T2	062-5965-00	062-5965-01
Graphing T3	062-5966-00	062-5966-01
Graphing D1	062-5967-00	062-5967-01
Graphing D2	062-5968-00	062-5968-01
Interfacing T1	062-5984-00	062-5984-01
Mapping T1	062-5980-00	062-5980-01
Mechanical Engineering T1	062-5979-00	062-5979-01

Package Title	Documentation Part #	Package Part #
Programming Aids T1	062-5971-00	062-5971-01
Programming Aids T2	062-5972-00	062-5972-01
Project Aids T1	062-5985-00	062-5985-01
Project Aids D1	062-5986-00	062-5986-01
Recreational Plots T1	062-5989-00	062-5989-01
Slidemaker T1	062-5962-00	062-5962-01
Slidemaker D1	062-5963-00	062-5963-01
Text Processing T1	062-5969-00	062-5969-01
Text Processing D1	062-5970-00	062-5970-01
Utilities T1	062-5974-00	062-5974-01
Utilities D1	062-5975-00	062-5975-01
Tekniques Vol. 5 No. 4 T1	062-5981-00	062-5981-01
Tekniques Vol. 6 No. 1 T1	062-6443-00	062-6443-01
Tekniques Vol. 6 No. 1 D1	062-6442-00	062-6442-01
Tekniques Vol. 6 No. 2 D1	062-6515-00	062-6515-01
Tekniques Vol. 6 No. 3 D1	062-6516-00	062-6516-01
Tekniques Vol. 6 No. 3 T1	062-6514-00	062-6514-01

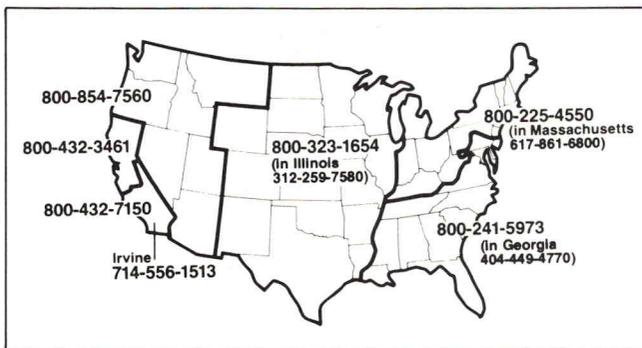
4110 Series

As 4110 Series program packages become available, they will be announced in *Tekniques*. A catalog will be published when a sufficient number of packages are collected.

U.S. Orders

The field office has the current prices.

Order IDD Applications Library programs through the toll-free number of Tektronix Central Parts Ordering. The following map delineates the geographical regions and the toll-free number serving each region.



Call the number serving your area and give the customer service representative the nine-digit part number and name of the Applications Library program you wish. If you have any questions, call your local Tektronix Field Office.

Orders Outside U.S.

To order a copy of the catalog, or to order a package, contact the local Tektronix sales office or one of the Libraries serving your area. See Library Addresses section of *TEKniques*.

Program Contributions

Contribute one program to the Applications Library and receive the package of your choice in exchange. Send in the membership card from your 4050 Series Graphic System Reference Manual to get the details. In the U.S., you may call us at (503) 685-3618. Outside the U.S., call your local Tektronix sales office or your Applications Library. See Library Addresses section of *TEKniques*.

Software Support Category C

The program material contained herein is supplied without warranty of any kind, and without any representation regarding quality, performance or suitability. *TEKTRONIX* specifically disclaims any implied warranties of merchantability or fitness for a particular purpose. Software support is *TEKTRONIX* Category C: Software is provided on an "as is" basis.

TEKniques Vol. 6 No. 3 T1

Part #062-6514-01

TEKniques Vol. 6, No. 3 T1 tape consists of 18 programs: five programming aids, one slidemaker, three CAD, one mapping, two recreational, two interfacing, and four educational/research.

Three of the programs must be transferred to their own dedicated tapes. Complete instructions for accomplishing the transfers are included in the documentation.

The individual abstracts describe the programs.

Program 1

Title: Simplot/Simpose

Author: G. Marksteiner
Intelab Applied Physics Laboratories, Inc.
Hingham, Massachusetts

Memory Requirement: 16K
Peripherals: Optional-4052R07, R08 Signal Processing ROMS
-4662/3 Plotter

Files: 1 ASCII Program
Statements: 516

SIMPLOT/SIMPOSE is a flexible graphics utility designed to quickly display the contents of one-dimensional arrays during program development and testing or during discrete data processing tasks. It may reside as a background subroutine which is accessible through the User-Definable Keys.

Plotting is performed within a limited field on either the 4052 display or the plotter. Adequate space on the left-hand side of the display is preserved for sequential data manipulation commands and other calculations.

This utility's versatile commands permit sectional plotting and superposition of arrays of varying size, and perform array integration, differentiation and Fourier transformation, without disturbing data in the input array. Input arrays may be labelled from A(A0)-Z(Z0). Autoscaled plots can be constructed using linear segments, vertical lines, bars or any keyboard symbol which is appended to the plot command.

An error-tolerant monitor oversees user syntax and spacing, and permits liberal use of abbreviation. Pressing UDK 3 displays an abbreviated listing of available forms supported by the utility.

Program 2

Title: ASCII Program Comparison

Author: Barbara Garbarino
Tektronix, Inc.
Wilsonville, OR

Memory Requirement: 16K
Peripherals: 4924 Tape Drive
Files: 1 ASCII Program
Statements: 157

This program will make a line-by-line comparison of two ASCII programs and display the changes necessary to make the two identical. The programs to be compared must reside on two different tapes, with the same file number. The program will prompt for the master program and the program to be compared to it and instruct which to place in the 4050 internal tape drive and which in the 4924 Tape Drive.

The changes may be displayed on the 4050 screen or sent to the printer.

Program 3

Title: Numerical Expression Evaluator

Author: Seymour Hatch
CR Industries
Elk Grove Village, IL

Memory Requirement: 8K
Files: 1 ASCII Program
Statements: 74

Frequently, when inputting parameters during the execution of a program, you wish to enter a simple numerical expression instead of pure

numbers. This subroutine will evaluate such an expression and return the pure number to the main program. A sample main program is included to demonstrate operation.

Numerical functions evaluated: COSine, SINE, TANgent, sum, difference, product and division, as well as a constant.

Program 4

Title: Syntax Diagrams

Author: Bennet Yee
Tektronix, Inc.
Beaverton, OR

Memory Requirement: 32K
Peripherals: Optional-4662/4663 Plotter
Files: 1 ASCII Program
1 ASCII Data (example)
Statements: 713

The program allows you to generate a syntax diagram, store it on tape and do limited editing (delete last entry) and graph the diagram on the screen or plotter.

Graphics operations are based on two points: the predefined point and the current point. For example, when constructing a line, you will move the cursor to one end of the line, define that point (UDK 1), then move the cursor to the other end and press the LINE key (UDK 5). A line is drawn between the points.

Predefined constructs such as arrow, bypass, rectangles, circle and parallelogram will automatically update the predefined point to convenient positions (i.e., CIRCLE will move predefined point to right-hand edge of the circle). Characters may be inserted in the graph.

Graph titles may be saved on tape. These do not show on the graph itself, but serve to identify the graph. The extended header is used to store this title.

Program 5

Title: Pretty Page Listings

Author: Thomas A. Price
Lorillard div. of Loews
Greensboro, NC

Memory Requirement: 32K
Peripherals: 4641 Printer
Files: 1 Program
Statements: 182

This program produces compiler-like listings for 4050 BASIC programs. The enhanced listing includes loop nesting and a complete variable cross-reference listing.

The listing appears on numbered, titled pages. If the system has a 4907 disk, the date and time also are printed on the page.

Leading zeros are added to the line numbers so that all statements begin in the same column.

FOR/NEXT loops are indented.

A single line is skipped before each section of code following a RETURN or END statement.

A complete list of the variables used in the program and the statements in which they were referenced is printed.

The source program must reside in an ASCII file on either the internal tape unit or a 4907 disk drive. Output is directed to the 4641 Printer.

Program 6

Title: Viewgraph

Author: Michael J. Nusca
U.S. Army
Aberdeen Proving Ground, MD
Memory Requirement: 32K
Peripherals: 4662/3 Plotter
Files: 1 ASCII Program
Statements: 1049

A viewgraph consisting of text, geometric figures and designated areas for photographs or plots is produced through this program. User-Definable Keys direct the flow of the program. Text may be positioned or centered. Titles are printed in boldface. Boxes, lines, diamonds and bullets may be drawn. The viewgraph may be previewed on the 4050 screen, then transmitted to the plotter.

No provision is made for storing data or selecting character size.

Program 7

Title: Plot Digitize

Author: Michael J. Nusca
U.S. Army
Aberdeen Proving Ground, MD
Memory Requirement: 32K
Peripherals: 4662 Plotter
Files: 1 ASCII Program
Statements: 362

The program will digitize and convert plotted points to user data units from a plot drawn on any set of cartesian axis — no semilog or logarithmic scales. Plots may be on paper or photographs, but they must fit within the 4662 Plotter's bed, i.e., 16" × 10½".

Once the plot is digitized, you may replot it, list the data, write the data to tape, or transfer the data to a Cyber 173 host computer.

Accuracy of the digitizing depends on the accuracy of the user in locating each data point. The speed depends on the time it takes the user to locate each data point and depress a switch on the plotter.

Program 8

Title: Arrowhead Macro

Author: Terry L. McCain
Purdue University
W. Lafayette, IN
Memory Requirement: 8K
Peripherals: 4663 Plotter w/Opt. 31
or 4663S
Files: 1 Program
1 Data (example)
Statements: 55

The Arrowhead Macro is a short program which loads a macro into the 4663 Plotter (the plotter must have Option 31 — Circular Arcs and Programmable Macros).

When expanded directly or from the user's program, the macro will draw an arrowhead at the current pen location. The orientation of the arrowhead will be the orientation of the last line drawn by the plotter. The program will prompt for the plotter address, macro number, and length and width of the arrowhead desired.

The window in use at the time the program is executed determines the actual physical size of the arrowhead — subsequent window commands will not alter the size of the arrowhead. The user may store several different sizes of arrowheads by changing the macro number each time the program is executed.

Program 9

Title: Drill Log - Error Checking

Author: Ron DeJarnette
Geological Survey of Alabama
University, AL
Memory Requirement: 8K
Files: 1 ASCII Program
Statements: 168

This program checks the math associated with manual drill log entry. You enter lithology intervals directly into the computer in feet and inches/100 or feet/100. The program increments each interval to calculate the depth and decrements each interval from the surface elevation to calculate the top of the lithology elevation.

The display lists the interval thickness, the top of the lithology elevation (tle) and the depth.

There is no provision to store data.

If using a 4054, you may choose character size 1 or 4. The screen page size will be calculated automatically.

Program 10

Title: 49XP IEEE-488 I/F Verification

Author: Dave Green
Tektronix, Inc.
Beaverton, OR
Memory Requirement: 32K
Peripherals: 492P or 496P Spectrum Analyzer
 GPIB Cable
Files: 1 ASCII Program
Statements: 355

Using a 4050 Desktop Computer, this program allows a 49XP program-mable spectrum analyzer user to functionally verify the IEEE-488 interface in that instrument.

All interface lines are checked for functionality, as well as all interface messages, with the exception of Parallel Poll Configure and Parallel Poll Unconfigure. The instrument's interface is checked for operation on other primary addresses, as well as in the talk only and listen only modes. Correct selection of Input/Output delimiter via a rear panel switch is also verified.

Faults in the interface are diagnosed only to the level that they affect normal communications with a 4050 controller. Detailed conformance to the IEEE standard is not checked.

Program 11

Title: Computer Poker

Author: Louis A. Matej
Tacoma, WA
Memory Requirement: 32K
Files: 1 ASCII Program
Statements: 903

From one to five players may play a regular-standard game of poker. Each player receives five cards, can bet up to \$50, can fold before or after any bet is made, and can discard one to five cards in exchange for new ones, or can retain all.

Each player is addressed by name. All hands are examined and recorded on the screen and the winner is determined.

The only difference from regular poker is that all cards may be seen by all players and only one bet is allowed. Ties are resolved or the pot is divided.

Program 12

Title: 4050 Series Clock

Author: Ron Boerger
SARPMA/ACD
San Antonio, TX

Memory Requirement: 8K
Peripherals: Optional-4907 File Manager
-4662 Plotter

Files: 1 ASCII Program
Statements: 105

This program draws either a 24- or 12-hour clock on the screen of a 4050 Series Desktop Computer. You may also choose to draw the clock on another device such as a plotter.

The time is normally calculated from the 4907 File Manager; however, a modification for those who do not have disk drives allows you to directly enter the time on the keyboard.

The program includes many REMarks to help you tailor the program for your own use.

Program 13

Title: Granulometry

Author: G.L. Kirkpatrick
Florida State University
Tallahassee, FL

Memory Requirement: 24K
Peripherals: Optional-4641 Printer
Files: 1 ASCII Program
Statements: 360

This program computes and prints statistical information valuable to geologists (clastic sedimentologists) when comparing size relationships among various sediment samples.

It uses raw weights in equally spaced size classes of sieve data and calculates the statistical parameters: Mean, Standard Deviation, Skewness, Kurtosis, weight percent and cumulative weight percent (by size class), total sample weight, and nine important percentile values for the sample.

Four separate pages of output are produced. The first contains moment measures and weight percent data. The second contains percentile information. The third is a frequency percent by size class histogram. The last is a probability plot of cumulative frequency percent data.

All input and output are designed to be in grams (for weight) and phi units (for size).

Program 14

Title: Graphical Linear/Nonlinear Regression Analysis

Authors: Khiem Ho
George Tzitzikalakis
Orthopedic Biomechanics Lab
Columbia University
New York, NY

Memory Requirement: 32K
Peripherals: 4662 Plotter
Files: 1 ASCII Program
1 ASCII Text
Statements: 900

This program will:

- A) *Plot two- or three-parametered functions* defined by the user. The user will also select the appropriate dimensions for the axes as well as the labels and titles.
- B) *Analyze data by nonlinear least squares fit to an arbitrary function.* The user will determine which type of function is appropriate and define the function. The program will find the values of coefficients which make that function best fitting to the data.
Analyze data by linear least squares fit to a polynomial. The user will select which order of polynomial is appropriate. The program will find the best coefficients for the polynomial.
Analyze data by linear regression with confidence bands. The user will decide which confidence bands. The program will find the best fitted curve and its appropriate confidence limits.
- C) *Plot scatter diagram as well as the best fitted curve.* The user will select the format, i.e., axes parameters, labels, connection of data points, symbols for data. The user can also select the necessary plotting: two- or three-parametered function, scatter diagram with/without the best fitting curve, linear regression with confidence bands, etc.

Data may be edited, listed, stored and loaded.

Program 15

Title: Two Phase Simplex Method

Author: Paul J. Ossenbruggen
University of New Hampshire
Durham, NH

Memory Requirement: 16K
Peripherals: Matrix ROM Pack
Optional-Line Printer
Files: 1 ASCII Program
Statements: 127

The simplex method is a technique for finding the optimum solution of a mathematical model consisting of a linear objective function and a set of linear constraint equations. By introducing slack, surplus, and artificial variables, the model is transformed into a model consisting of two linear objective functions and a set of linear constraint equations with strict equalities.

Students interact with this program to learn the two phase simplex method. The students must know how to transform the original model to the standard two phase simplex form. The parameters of the model are introduced and stored in the 4050 as constants, vectors and matrices. By eliminating tedious mathematical calculations, students are able to focus their attention on learning the rules of the method.

Program 16

Title: Exercise Test Evaluation

Author: Glenn Street
Cleveland Clinic Foundation
Cleveland, OH

Memory Requirement: 32K

Files: 1 ASCII Program

Requires dedicated tape

Statements: 804

This program plots post-exercise patient data. Heart rate, blood pressures (systolic and diastolic) and rate pressure product responses to an exercise or stress test are graphically displayed on the 4050 screen. Data input required:

Patient's name and clinic number

Date

Type of test (treadmill, bicycle, step)

Workload labels (speed and % grade of treadmill, or Kpm/min if using bicycle ergometer, or steps/min if step test)

Rest or recovery intervals (if any) and minute numbers

Heart rates and blood pressures for each interval of test (workloads and (if selected) rest and/or recovery intervals).

Workload labels, heart rates and blood pressures may be edited from the keyboard.

If a lab uses the same protocol(s) — treadmill, bicycle and step — regularly, you can store your protocols on tape for future recall.

Program 17

Title: RDrawing

Authors: Steve Duncan
John Carter
Tektronix, Inc.
Wilsonville, OR

Memory Requirement: 4054 w/Opt. 30/31
64K

Peripherals: Optional-4907 File Manager
-4662 Plotter

Files: 7 Binary Programs

1 Binary Data

Transfer to a dedicated tape or disk

Statements: 1600

The Refresh Drawing program enables you to create drawings on the 4054 (with Option 30/31) that can be output to a plotter in color and saved on tape or disk.

The program takes advantage of the refresh graphics capability of the 4054 allowing you to pick a graphic object (any symbol previously defined by you or a standard symbol provided by the program) from a menu, and scale or rotate the object, and place the object anywhere in the workspace of the 4054 screen.

A zoom feature allows finite control over detailed drawing in the workspace.

Vectors, labels, editing, status, trim, plotter output, reference grid, and circle/arc drawing are included.

Program 18

Title: 4050 Data Comm Utility

Author: Stephanie Ranney
Tektronix, Inc.
Woodbridge, NJ

Memory Requirement: 24K

Peripherals: Option 1/3 I/F

Optional-4662 Plotter

Files: 6 ASCII Program

Transfer to a dedicated tape

Statements: 1230

This program will help 4050 users interface to any host computer that supports an ASCII asynchronous terminal. It explains and allows you to input the parameters describing: host, downloading and uploading environments. Once the parameters are established, the program will transfer data to or from the host.

The program will save on tape the parameters for each of the three environments. A worksheet may be plotted on the 4662 Plotter.

TEKniques Vol. 6 No. 3 D1

TEKniques Vol. 6 No. 3 D1 disk consists of four disks, each containing from one to 12 symbol libraries for the PLOT 50 2-D Drafting package (4054D08). The disks may be used "as is" with PLOT 50 2-D Drafting since they all have space for the temporary files created by the drafting package. The individual abstracts describe the programs.

Program 1

Title: 2-D Drafting/Complex Fonts 1

Author: Jim Gish
Tektronix, Inc.
Portland, OR

Memory Requirement: 4054/64K
Peripherals: PLOT 50 2-D Drafting
software plus necessary
hardware

This is a symbol disk for PLOT 50 2-D Drafting (4054D08) that contains 11 symbol libraries. The libraries contain uppercase and lowercase alpha, numeric, arithmetic, and punctuation characters in the following fonts: Medium Script, Old English, German, Italian and Cyrillic. Characters from various fonts are included in a miscellaneous library.

The disk can be used "as is" since space has been reserved for the temporary files created by PLOT 50 2-D Drafting.

Underlays for each library are included showing the graphic content.

The symbols were transcribed from the Superfont program in the Applications Library (Character Generator T1 package). Some reorganization was done to realize as much consistency between fonts as possible.

Program 2

Title: 2-D Drafting/Complex Fonts 2

Author: Jim Gish
Tektronix, Inc.
Portland, OR

Memory Requirement: 4054/64K
Peripherals: PLOT 50 2-D Drafting
software plus necessary
hardware

This is a symbol disk for PLOT 50 2-D Drafting (4054D08) that contains 11 symbol libraries. The libraries contain uppercase and lowercase alpha, numeric, arithmetic, and punctuation characters in the following fonts: Medium Roman, Bold Roman, Medium Italic, Bold Italic, and Medium Gothic. Characters from various fonts are included in a miscellaneous library.

The disk can be used "as is" since space has been reserved for the temporary files created by PLOT 50 2-D Drafting.

Underlays for each library are included showing the graphic content.

The symbols were transcribed from the Superfont program in the Applications Library (Character Generator T1 package). Some reorganization was done to realize as much consistency between fonts as possible.

Program 3

Title: 2-D Drafting/Complex Fonts 3

Author: Jim Gish
Tektronix, Inc.
Portland, OR

Memory Requirement: 4054/64K
Peripherals: PLOT 50 2-D Drafting
software plus necessary
hardware

This is a symbol disk for PLOT 50 2-D Drafting (4054D08) that contains 12 symbol libraries. The libraries contain uppercase and lowercase alpha, numeric, arithmetic, and punctuation characters in the following fonts: Light Gothic, Light Script, Greek Gothic, and Greek Roman. Additional libraries have collections of math symbols, cartography symbols, and some general marker symbols. Miscellaneous characters from various fonts are included in another library.

The disk can be used "as is" since space has been reserved for the temporary files created by PLOT 50 2-D Drafting.

Underlays for each library are included showing the graphic content.

The symbols were transcribed from the Superfont program in the Applications Library (Character Generator T1 package). Some reorganization was done to realize as much consistency between fonts as possible.

Program 4

Title: 2-D Drafting/ELECONTROL

Author: Mark Hoffman
Tektronix, Inc.
Rochester, NY

Memory Requirement: 4054/64K
Peripherals: PLOT 50 2-D Drafting
software plus necessary
hardware

This is a symbol disk for PLOT 50 2-D Drafting (4054D08) that contains a library of 37 symbols. The library was taken directly from a template which conforms to the NEMA and JIC standards.

The disk can be used "as is" since space is available for the temporary files created by PLOT 50 2-D Drafting.

An underlay of the library is included showing the graphic content. 

4050 Series Program Updates

Package/Program/File

062-5964-01/GRAPHING T1
Program 1/Timeseries III
File 2

062-5967-01/GRAPHING D1
Program 1/Timeseries III
"@TIME3/MAIN"

Submitted by: Mallory Green
Dept. of HUD
Washington, D.C.

The main file in the tape and disk packages should have one statement changed:

```
OLD:  940 DIM A$(A0*1+1),A(A0),B1(A0),D$(1),D(D0,H0)
NEW:  940 DIM A$(A0*1+1),A(A0),B1(A0),D$(2),D(D0,H0)
```

If D\$ is not dimensioned to (2), it causes 4051 tape save problems.

062-5966-01/GRAPHING T3
Program 5/General Graphing
File 14*

Part of this program uses constants for the plotter address and part uses variables. Change the following statements to reflect the variable K5 instead of address 10 for the plotter address:

```
2460
2550-2570
2660
2700-2780
2930-2950
3030
3070-3140
3610-3630
3650-3680
3700-3720
```

Make a note in your documentation that if your plotter address is not 10, statements 640 and 1020 of this file will also have to be changed.

Future releases of this package contain the changes, plus K5 will be set to 1 in statements 640 and 1020.

* File 14 is the correct file on the GRAPHING T3 tape. However, if you have transferred this program to its own dedicated tape as instructed in the documentation, this file will be File 4 on the dedicated tape.

062-5971-01/PROGRAMMING AIDS T1
Program 12/FORTRAN to BASIC Converter
Files 16 and 17

Submitted by: Herman D'Hondt
Tektronix, Inc.
Sydney, Australia

The above program requires a 4924 which users may not possess. It's quite simple to make it run with a 4907 or even with just the internal tape.

The changes below enable the program to work on a 64K 4050 without any peripherals.

After the above files have been transferred to a dedicated tape, tape files 1 and 3 contain the programs, as before. File 4 must contain the FORTRAN source, and File 2 is used both as a scratch file and to store the resultant BASIC program. This file should be made to 20000 bytes.

Changes to File 1

Delete lines 1770, 1780, 2930.

Change and/or add the following lines:

```
1070 DIM T$(1500),M$(2000),Z$(1)
1075 M$=""
1077 Z$=CHR(13)
1800 FIND 4
1810 ON EOF (0) THEN 2920
1830 INPUT @33:A$
2860 M$=M$&C$
2865 M$=M$&Z$
4030 FIND 2
4032 PRINT @33:M$
4035 CLOSE
4037 FIND 3
```

Changes to File 3

Delete line 5040.

Change and/or add the following lines:

```
5030 M$=""
5550 M$=M$&C$
5555 M$=M$&Z$
5810 M$=M$&C$
5815 M$=M$&Z$
5840 REM SAVE THE BASIC PROGRAM
5850 FIND 2
5860 PRINT @33:M$
5865 CLOSE
10410 M$=M$&C$
10415 M$=M$&Z$
```

062-5985-01/PROJECT AIDS T1
Program 13/Project Schedule Chart
Files 31 and 32

Submitted by: Dan Butcher
Tektronix, Inc.
Beaverton, OR

A CLOSE statement needs to be added to each of the files listed above if you are using this program on a 4052 or 4054 (see the programming tip in *TEKniques* Vol. 5 No. 4, "Changing a 4052/54 File Header," for an explanation).

File 31: Add statement
1355 CLOSE

File 32: Add statement
315 CLOSE

If you have already transferred these programs from the *TEKniques* tape to a dedicated tape, they will be programs 2 and 4 on the dedicated tape.

062-5989-01/RECREATIONAL PLOTS T1
Program 5/Othello
File 7

Submitted by: Joel T. Hicks, P.E.
General Technology
Little Rock, AR

In some versions of Othello, statements 245 and 250 are in error. The program always branches to statement 281.

Several fixes are available; I prefer the POS statement as suggested in one of the programming tips:

```
OLD:  245 G$=SEG(G$,1,1)
NEW:  250 IF G$<>"YES" THEN 281
```

```
245 GO TO POS(" YN",G$,1) OF 250,260,280
250 GO TO 230
```

062-5962-01/SLIDEMAKER T1
Program 2/Signmaker IV
File 3

To access the correct fonts on this file, one statement in File 3 requires a change. Statement 750 should read:

```
750 V0=4
```

062-5970-01/TEXT PROCESSING D1
Program 2/Recordkeep II
"@DBMPROG/FILEINIT"
"@DBMPROG/RECORDKEEP/OV5"

Submitted by: Nathan Oxhandler
Tektronix, Inc.
Vancouver, BC

If you are using more than one disk, i.e., a data disk and a program disk, add the following statements to the above named files:

```
@DBMPROG/FILEINIT
1555 CALL "UNIT", A8
```

```
@DBMPROG/RECORDKEEP/OV5
4275 CALL "UNIT", A8
```

Without these, the system remains set to the data disk unit of A9 and you can't OLD in a program file.

062-5970-01/TEXT PROCESSING D1
Program 2/Recordkeep II
"@DBMPROG/RECORDKEEP/OV1"

Submitted by: Mary Fielding
DuPage County Development Dept.
Wheaton, IL

Line 5800 in the above named file asks: "JDo you wish to delete another item? (Y/N): ". If answered with a 'Y', the program stays in the loop. If answered with an 'N' or anything else except 'Y', it goes on to renumber the remaining items. More than once I answered with the item number instead of a 'Y' which caused the renumbering. Before I could finish deleting, I had to relist everything to find the new item numbers. By adding the following statement, you won't get caught:

```
5815 IF Y$ <> "Y" AND Y$ <> "N" THEN 5800
```

062-6443-01/TEKniques Vol. 6 No. 1 T1
Program 3/4054 Dynamic Graphics Flowchart Symbols
Abstract

Submitted by: Craig Bulmer
Tektronix, Inc.
Chicago, IL

The abstract for the above program omitted some features which are very important.

- 1) Objects can be scaled both horizontally and vertically, and rotated.
- 2) The four character sizes of the 4054 are supported and proper scaling of characters is output to the plotter.
- 3) Reference dots (not snap-to-grids) are optional and their scale can be changed.
- 4) Reference dots and menu are not output to the plotter.
- 5) The 4054 screen can be hard copied with or without reference grid and menu.
- 6) Objects, text, and vectors are brought up in refresh for positioning.

062-5981-01/TEKniques Vol. 5 No. 4 T1
Program 8/CROSSREF
Files 9 and 20

Submitted by: J.A. Loughran
General Electric
Schenectady, NY

Bob Manthey has made a very commendable contribution to the package. His programs are well organized and admirably documented, making them easily understood. However, some of his programs contain a practice which can seriously limit their performance, namely branching out of incompleting FOR-NEXT loops.

Perhaps the worst case is found in CROSSREF, Program number 8, Files 9 and 20. Two very active loops, lines 1320-1360 and lines 1810-1840, are

regularly broken resulting in a large loss of memory. A minor broken loop is also found at lines 1810-1840. Correcting each of these loops with a properly branched dummy loop for memory recovery nearly doubles the cross-referencing capacity of the program. The following illustrates the correction code I used. There must be other methods too.

With these modifications, redimensioning R1 (line 220) and S1 (line 240) from 400 to 1000, a 4054 with 64K of memory easily cross-referenced a 50K program with 407 source and 700 reference lines. 

Original Version of File 9:

```
1300 REM ----- Find Reserved Word -----
1310 RESTORE 1100
1320 FOR I=1 TO T
1330 READ W$
1340 LET C=POS(L$,W$.1)
1350 IF C THEN 1400
1360 NEXT I
1370 GO TO 1200

1400 REM ----- Parse Reserved-Word -----
1410 LET C=C+LEN(W$)+1
1420 LET C9=LEN(L$)
1430 IF C>C9 THEN 1200
1440 LET C1=C9
1450 LET C2=C9-C+1
1460 GO TO 1 OF 1800,2000,2000,2000,2000,2000,1800,1800
1470 GO TO 1-B OF 1800,1500,1500,1500,1500,1600,1700,2000

1500 REM ----- 'LIST', 'SAVE' & 'APPEND' (I = 10 to 12) -----
1510 LET C1=POS(L$,"",C)
1520 IF NOT(C1) THEN 1800
1530 IF C1=C9 THEN 1200
1540 LET C=C1+1
1550 GO TO 1800

1600 REM ----- 'RENUMBER' (I = 13) -----
1610 FOR J=1 TO 2
1620 LET C1=POS(L$,"",C)
1630 IF NOT(C1) THEN 1200
1640 LET C=C1+1
1650 NEXT J
1660 GO TO 2000

1700 REM ----- 'CALL' (I = 14) -----
1710 LET C1=POS(L$,"",C)
1720 IF C1=0 OR C1=C9 THEN 1200
1730 LET C=C1+1

1800 REM ----- Validate Numerals -----
1810 FOR C1=C TO C9
1820 LET Z$=SEG(L$,C1,1)
1830 IF NOT(POS("0123456789".Z$.1)) THEN 1900
1840 NEXT C1

1900 REM ----- Count Numerals -----
```

Revised Version of File 9:

```
1300 REM ----- Find 'Reserved Word' -----
1310 RESTORE 1100
1315 FOR A0=1 TO 1
1320 FOR I=1 TO T
1330 READ W$
1340 LET C=POS(L$,W$.1)
1350 IF C THEN 1370
1360 NEXT I
1370 NEXT A0
1380 IF C THEN 1400
1390 GO TO 1200

1400 REM ----- Parse 'Reserved-Word' -----
1410 LET C=C+LEN(W$)+1
1420 LET C9=LEN(L$)
1430 IF C>C9 THEN 1200
1440 LET C1=C9
1450 LET C2=C9-C+1
1460 GO TO 1 OF 1800,2000,2000,2000,2000,2000,1800,1800
1470 GO TO 1-B OF 1800,1500,1500,1500,1500,1600,1700,2000

1500 REM ----- 'LIST', 'SAVE' & 'APPEND' (I = 10 to 12) -----
1510 LET C1=POS(L$,"",C)
1520 IF NOT(C1) THEN 1800
1530 IF C1=C9 THEN 1200
1540 LET C=C1+1
1550 GO TO 1800

1600 REM ----- 'RENUMBER' (I = 13) -----
1610 FOR A0=1 TO 1
1620 FOR J=1 TO 2
1630 LET C1=POS(L$,"",C)
1640 IF NOT(C1) THEN 1670
1650 LET C=C1+1
1660 NEXT J
1670 NEXT A0
1680 IF NOT(C1) THEN 1200
1690 GO TO 2000

1700 REM ----- 'CALL' (I = 14) -----
1710 LET C1=POS(L$,"",C)
1720 IF C1=0 OR C1=C9 THEN 1200
1730 LET C=C1+1

1800 REM ----- Validate Numerals -----
1810 FOR A0=1 TO 1
1820 FOR C1=C TO C9
1830 LET Z$=SEG(L$,C1,1)
1840 IF NOT(POS("0123456789".Z$.1)) THEN 1860
1850 NEXT C1
1860 NEXT A0

1900 REM ----- Count Numerals -----
```

IDD Applications Libraries

Africa, Europe, Middle East

Contact local sales office

Australia

IDD Applications Library
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

IDD Applications Library
Tektronix Canada Ltd.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

Caribbean, Latin America and Far East (excl. Japan)

IDD Group
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
U.S.A.

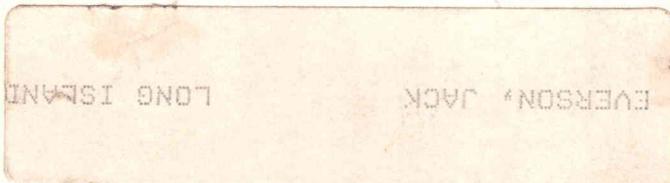
Japan

IDD Applications Library
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

United States

IDD Applications Library
Tektronix, Inc.
Group 451
P.O. Box 500
Beaverton, Oregon 97077

Address Correction Requested — Forwarding and Return Postage Guaranteed.



TEKTRONIX, INC.
Information Display Division
IDD Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97077

COMMITTED TO EXCELLENCE
Tektronix

TEKTRONIX, INC.

BULK RATE
U.S. POSTAGE
PAID