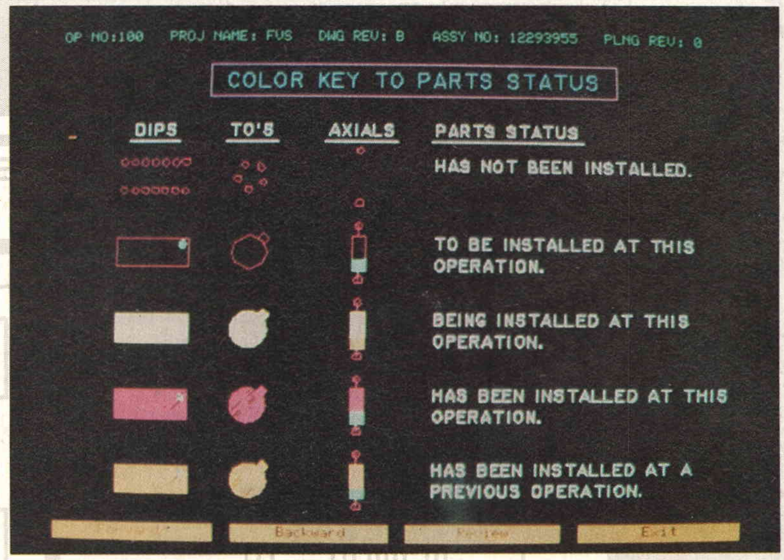
THE IDG
APPLICATIONS NEWSLETTER

VOL. 9 NO.3
SPRING, 1986

Tekniques



Computer Graphics Aids PCB Assembly at Hughes Aircraft Co.

Tektronix®
COMMITTED TO EXCELLENCE

Computer Graphics, Artificial Intelligence Speed Circuit Board Assembly

Assembly instructions generated by an expert system and displayed on Tek's 4105 terminals are replacing manufacturing workbooks at Hughes. Foot pedals and a bar code reader attached to the terminal keep the assembler's hands free for work.

Mounted on an articulated arm, with a foot pedal and bar code reader replacing its keyboard, Tek's 4105 Computer Display Terminal is part of an innovative manufacturing planning and control system in which assembly procedures are derived from a knowledge-based software program.

The system is currently used in the assembly of printed circuit boards (PCBs) at Hughes Aircraft Company's Electro-Optical and Data Systems Group (EDSG) in El Segundo, California. The boards are designed for the U.S. Army's Bradley Fighting Vehicle, M1 tank, A6 aircraft, etc.

Graphic Instructions

Instead of a bulky instruction workbook, PCB assemblers refer to instructional frames displayed on the 4105 (see Figure 1). The frames consist of both text and graphics; for example, they illustrate the circuit board to be assembled and the orientation of parts, and list the steps to be taken. The instructions indicate everything from tool choice to where to place the board for transfer to the next assembly station. Color coding conveys warnings and indicates which parts are to be installed in the current operation (Figure 2).

A custom-designed interface box plugs into the 4105's keyboard port and connects a bar code reader and two foot pedals to the terminal (see sidebar, "Bar Code Reader Available for 4100 Series"). To begin work, an assembler uses the bar code wand to scan a personal identification card and route sheets that contain information on the part to be assembled (Figure 3). The foot pedals are used to step through the menu and instruction frames, leaving the hands free for work. The left pedal selects menu options; the right pedal is used when

On the cover: Hughes Aircraft has developed a sophisticated PCB assembly system that makes use of Tek's 4105 terminal. Story on this page.

In this Issue

Tekniques, the IDG Applcations Newsletter, is published by the Information Display Group of Tektronix, Inc., Mail Stop 63-635, P.O. Box 1000, Wilsonville, OR 97070. It is distributed to users of TEKTRONIX computers and graphics products.

the assembler has completed an assembly step and is ready to advance to the next instruction frame (Figure 4). Information on the worker's progress in assembling the circuit board is recorded by software and used by production supervisors in planning and monitoring schedules and tracking the location of circuit boards.

Expert System Generates Instructions

The assembly workstations are part of an intelligent manufacturing planning system called the Hughes Integrated Classification System (HICLASS™ Software System). Manufacturing planning involves deciding the general sequence of operations required to manufacture an item, determining the step-by-step instructions of each operation, and then scheduling those operations. A number of companies have implemented computer-aided process planning (CAPP) systems that partially automate this process, but the Hughes system goes further. Using information from the engineering database, it generates detailed manufacturing plans automatically.

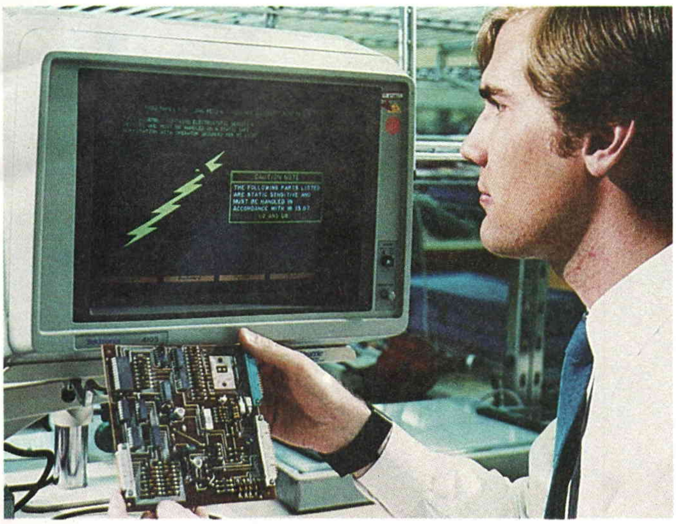


Figure 1. For circuit board assembly, instructions generated by an expert system and displayed on a 4105 terminal are replacing bulky instruction workbooks at Hughes Aircraft's Electro-Optical and Data Systems Group.

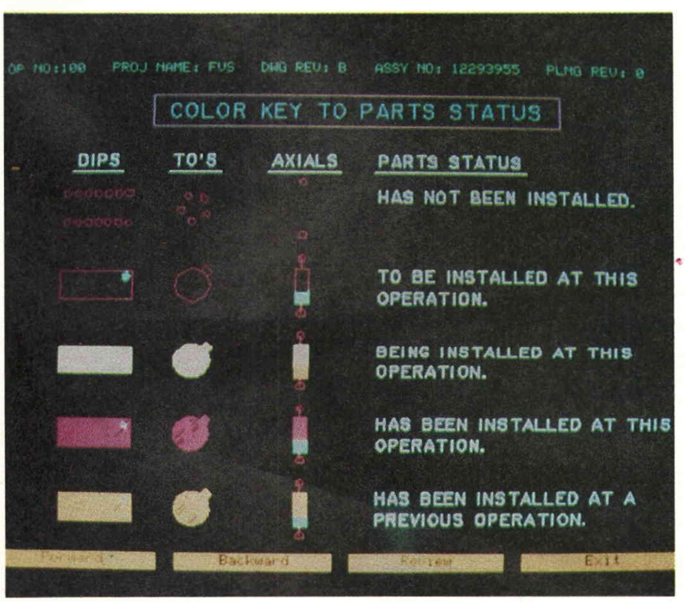


Figure 2. Color is used to indicate parts status.

The core of the HICLASS Software System is a Hughes-developed software package which was written in C and Pascal, and makes extensive use of artificial intelligence principles. The HICLASS Software System includes an expert system that mimics the reasoning process of manufacturing engineers. The system employs heuristics—rules of thumb determined after extensive interviews with manufacturing planners engineers—to deduce the required manufacturing operations, instructions and equipment from the information in the engineering database. For example, the HICLASS Software System includes over 1,000 rules that apply to PCB assembly.

The HICLASS Software System also utilizes natural language processing, another facet of artificial intelligence, to help interpret and understand notes made in the database by design engineers. The notes cover part specifications, special instructions and general design comments. The HICLASS Software System is also able to automatically extract part features from the CAD database, freeing planners from the need to describe the part.

Bar Code Reader Available for 4100 Series

by Tom Ireland
Terminals Division Marketing
Tektronix, Inc.
Wilsonville, OR

In its HICLASS™ Software System application, Hughes uses the PERCON 380A Bar Code Reader, made by Peripheral Connections of Eugene, Oregon, specifically for the 4100 Series terminals. The 380A scans standard Code 39 bar codes, allowing quick, accurate entry of data to the terminal.

The bar code reader is plug-compatible with the 4100 Series keyboard. It connects in-line between the keyboard and the terminal, or, for stand-alone operation, can connect directly to the terminal alone. In the latter case, the keyboard is optional.

To allow the use of macros during stand-alone operation, the 380A offers an optional, numeric entry pad with two special keys that emulate function keys on the 4100 Series keyboard.

Along with the port for connection to the terminal or keyboard, the 380A also has two optional input ports. One provides for attachment of up to two foot-activated switches that emulate two other terminal function keys. The other port accepts RS-232-C signals and converts them for input to the terminal. This enables an RS-232-compatible device, such as a voice input machine, to be attached to the bar code reader for input to the terminal.

For further information on using the PERCON 380A with your 4100 Series terminal, talk to your Tektronix field office or contact Peripheral Connections directly at 503/344-1189.

The HICLASS Software System operates in three phases, according to David Liu, Technical Head of the Software Engineering and Integration Section for EDSG and manager of the project that designed the HICLASS Software System. During the interpretation phase, the planning problem is defined by translating engineering databases into tokens that can be processed as symbols rather than numbers. In the reasoning phase, constraints and goals are evaluated, and symbolic processing continues until a conclusion or conclusions can be drawn. Finally, in the presentation phase, computer graphics are created and sent to the 4105s on the shop floor. Graphics in the instruction frames are in IGES format. Liu's section has written translation software in C that calls Interactive Graphics Library (IGL) routines to convert the graphics to PLOT 10® format. "We configured IGL for the 4105, then trimmed it to get the speeds we needed," Liu says.

4105 Fills the Bill

Hughes had several reasons for choosing Tek's 4105 to display the instruction frames. "The Tek 4105 was just what we wanted in terms of having sufficient resolution, a broad color palette to choose from, and the ability to display eight colors," says Liu.

"The speed of data communication is also very important," Liu notes. "High throughput is critical. You don't want the assembler to be waiting around while the terminal displays; you don't want the drawing speed of the terminal to be a bottleneck. So the 4105's 38.4K baud data transmission rate was very significant."

Liu gave high marks to the Tek support staff who helped in setting up the system and customizing the terminal. "We basically had to make the bar code reader and foot pedals mimic the keyboard, so we had to work closely with Tek to accomplish that," he says. "Tek's systems people were very helpful and very professional. They gave us the protocols between the keyboard and the terminal to enable us to do it."

Plans to Expand

Hughes will continue to apply artificial intelligence concepts to manufacturing, says Liu. As a Hughes document on the HICLASS Software System notes, "It takes a lot of effort to teach an expert system the basics. On the other hand, HICLASS integrates knowledge from many experts and disciplines, and it gives consistently accurate performance. And it won't retire or quit after two years."

Photographs courtesy of Hughes Aircraft Company.

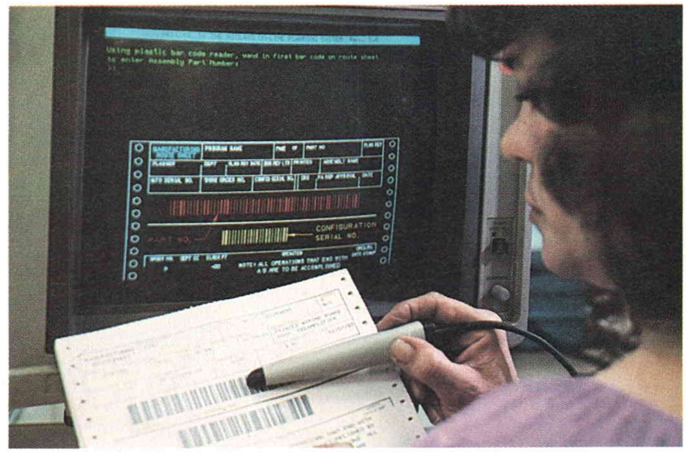


Figure 3. Assemblers use a bar code wand, connected via a custom interface box to the 4105's keyboard port, to scan their identification card and route sheets.

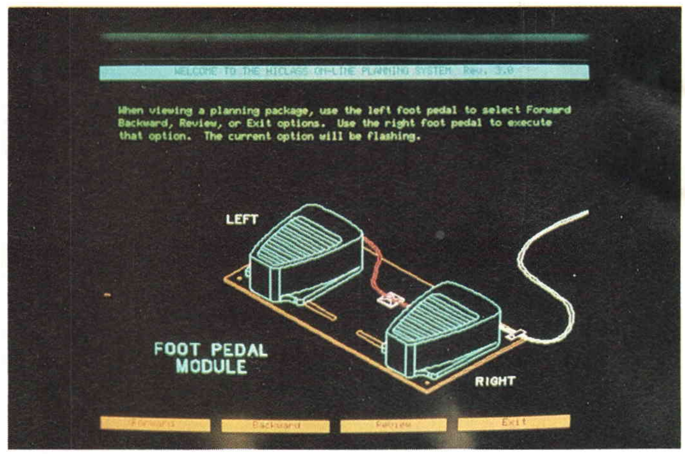


Figure 4. Instructional frames contain both text and graphics.

The new CX4692S multi-user output system connects directly to the IBM 3274 Cluster Controller and produces high-resolution paper and transparency output in brilliant colors.

Color Output System Provides Superior Graphics Hardcopy for IBM® 3270 Environments

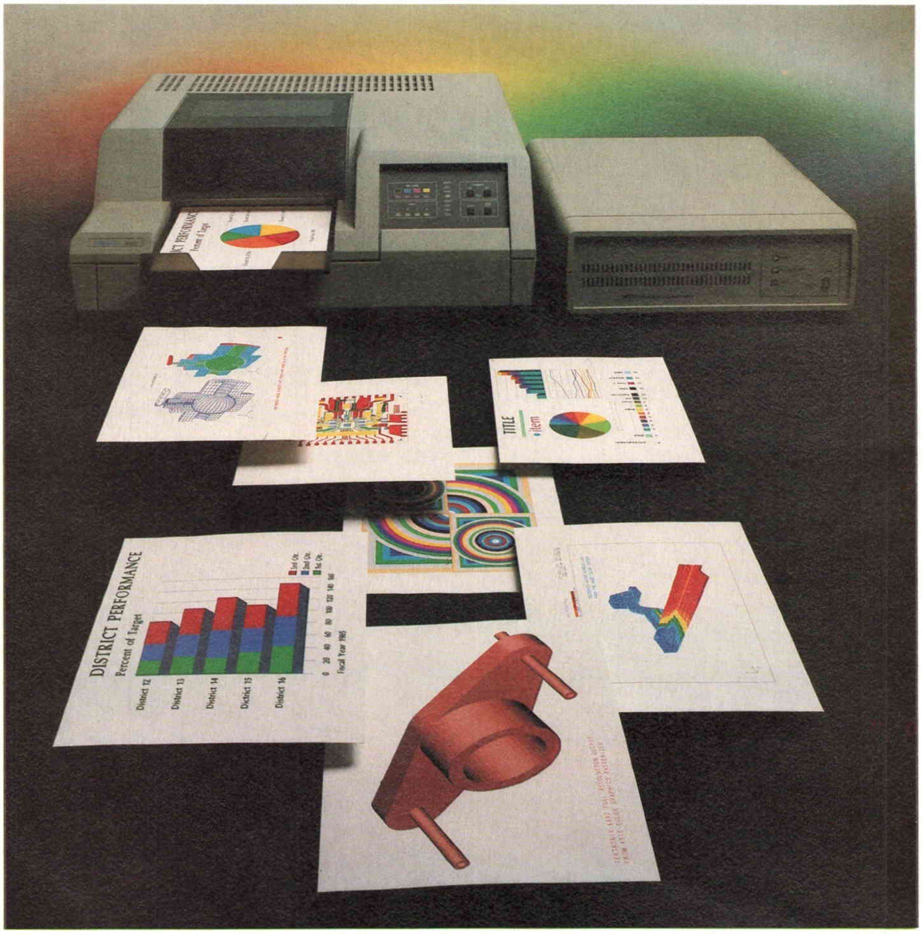


Figure 1. Output is produced at the copier's full resolution of 154 dots per inch, using 256 colors selected from a palette of over 130,000.

Whether a computer graphics application involves analyzing technical data, charting a sales trend, or designing a product, the ability to move the graphics image out of the computer and onto paper or transparency hardcopy is a necessity. Ideally, the process of obtaining that output should be fast, simple and cost-effective, and the results should be beautiful to behold.

With Tek's new CX4692S IBM-Compatible Color Graphics Output System, the ideal becomes a reality for users in IBM host environments. The system generates vivid paper copies (see Figure 1) in two minutes for about 15 cents each, and transparencies in three minutes for under a dollar—regardless of image complexity.

Joining Forces

The CX4692S system brings together two high-performance Tek graphics products: the 4692 Color Graphics copier, which uses proprietary ink-jet/airflow technology, and the CX4510A, an IBM-compatible version of the 4510A Graphics Rasterizer (introduced in the Winter, 1986, issue of *Tekniques*). The rasterizer enhances the graphics image to optimize output resolution, color and size. Output is produced at the copier's full resolution of 154 dots per inch, horizontally and vertically. Two hundred fifty six colors can be printed, chosen from a palette of over 130,000 colors. The image is also sized to optimally fill the page in either portrait or landscape format on A-size (8 1/2×11 inches) or A4-size (297×210 mm) media.

The CX4692S system connects directly to the IBM 3274 Cluster Controller via a coax interface using the standard IBM 3287 printer port configuration. This connection makes the output system a shared resource, available to all users on the IBM host system. Both the MVS/TSO and VM/CMS operating systems are supported. (A 4692S Color Graphics Output System is available for users of ASCII hosts, such as those manufactured by DEC, Prime, Harris, and so forth.)

Use of the color output system is independent of the terminal employed to create the original image. The image might be created using an IBM terminal, Tek's CX4100 Series terminals or 4120 Series workstations with Option 5, or other IBM 3270-compatible terminals—or even a non-graphics terminal. Figure 2 shows the color output system as a shared resource in a typical mid-sized IBM site.

In addition to the coax interface, two RS-232-C input ports provide extra flexibility by enabling the color output system to connect to ASCII hosts, terminals, or workstations.

Applications Software Support

A key question for potential users of the color output system is whether it will work with their applications software packages. Currently, software support is available for several of the most widely used graphics software packages: SAS/GRAPH® from SAS Institute Inc., and TELL-A-GRAF® and DISSPLA® from Integrated Software Systems Corporation (ISSCO®).

In addition, a new PLOT 10® software package, the Graphic spooler Interface (GSI), allows GDDM® -based application programs that produce ADMGDF files to send output to the color output system (see Figure 3). (For example, the Interactive Chart Utility (ICU) can be used with the Graphic spooler Interface.) PLOT 10 GSI also supports Tek's PLOT 10 Graphics Kernel System (GKS) and PLOT 10 Interactive Graphics Library (IGL). PLOT 10 GSI is not required for use of the ISSCO and SAS Institute packages.

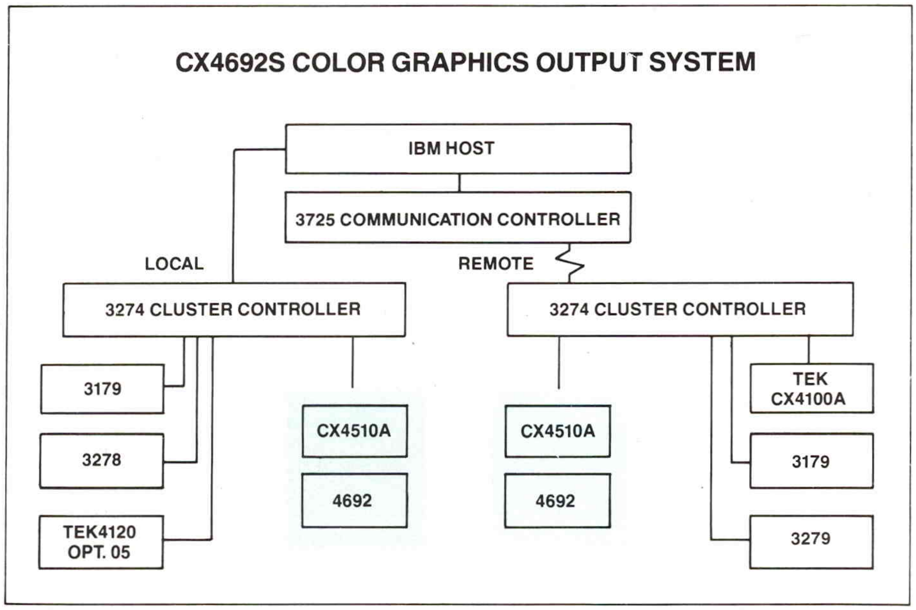


Figure 2. The CX4692S Color Graphics Output System is coax-attached to a local or remote IBM 3274. As a shared resource the CX4692S is supported on MVS/TSO and VM/CMS operating systems.

Performance and Productivity

The color output system promotes both human and system efficiency. For example, the system improves overall host system performance by offloading the CPU- and time-intensive tasks of rasterization and image enhancement necessary to achieve presentation-quality color hardcopies. The host spooler routes files to the output system, and handles queueing and priority management.

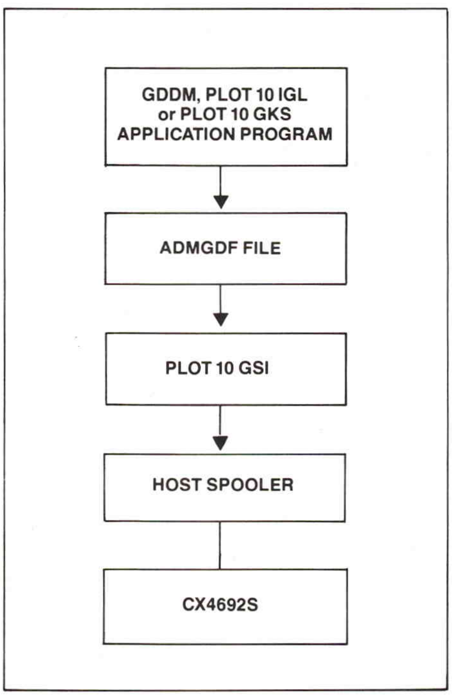


Figure 3. PLOT 10 GSI translates the ADMGDF files produced by IBM GDDM or PLOT 10 IGL or PLOT 10 GKS application programs into Tektronix CX4510A graphic format for output to the CX4692S Color Graphics Output System.

The color output system enhances user productivity by minimizing the time terminals are dedicated to transmitting data to get a hardcopy. When connected to the host, files can be copied for printing with virtually no terminal tie-up, thus increasing productivity over a terminal-dedicated printer or plotter.

The system also provides an easy-to-use interface through GSI. GDDM-based applications programs (or any application that is capable of producing an ADMGDF file) can specify options such as the number of copies, background colors and line widths. Convenient default values can be specified when the GSI software is installed.

Another dimension of productivity to consider in evaluating output systems is maintenance costs. The color output system includes features such as automatic self-maintenance routines and automatic sheet loading and stacking—up to 100 sheets of paper or up to 50 sheets of transparency material can be processed automatically, and changing between the two types of media is quick. The copier even takes care of the interleaved paper between transparency media. Ink quantity is electronically monitored, and a warning light for each color indicates when the ink is low. About 4000 copies can be made by a set of four self- sealing ink cartridges, and when it's time to replace them, the cartridges can be snapped into place in seconds from the front of the copier.

A Complete Color Graphics Output Solution

High-quality copies, produced quickly, easily and cost-effectively—it all adds up to a complete color graphics output solution for IBM 3274 environments. For further information on the CX4692S Color Output System, talk to your local Tektronix field office.

Fabric patterns are designed online and viewed on Tek's 4125 and 4115 workstations, making both pattern design and textile production more efficient.

At German Weaving Mill, Computer Graphics Aid in Design of Fabric Patterns

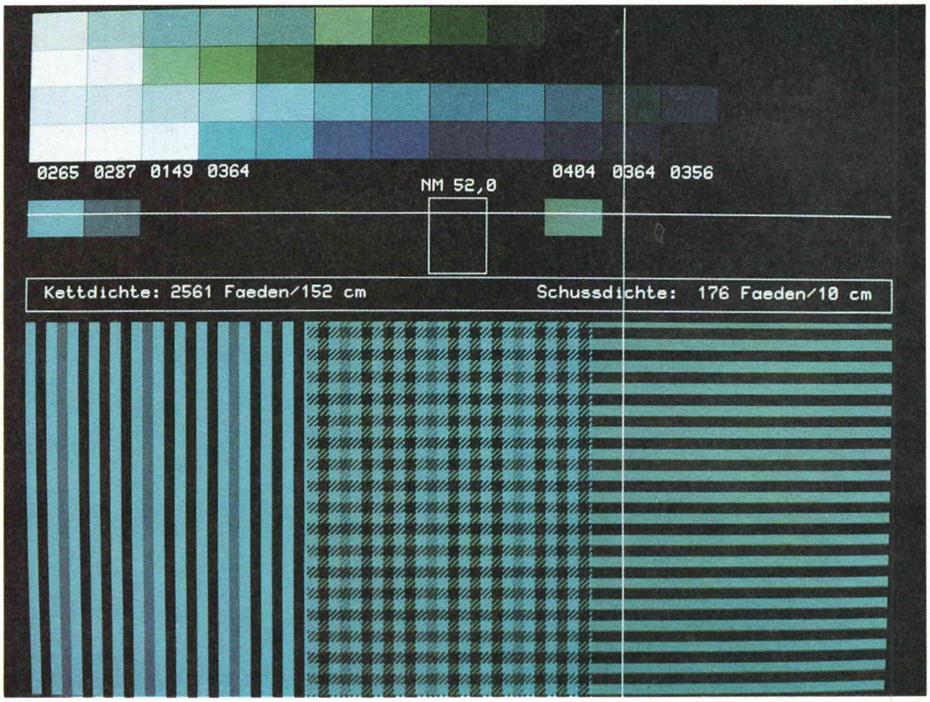


Figure 1. Textile designers at Gaenslen und Voelter in West Germany use the large color palette of the 4125 and 4115B Color Graphics Workstations to develop weaving patterns.

Computers and textile design have been linked ever since 1801 when the French weaver Joseph Jacquard invented an automatic loom that was controlled by punched cards similar to modern keypunch cards. In the modern era, textile and clothing designers are making innovative use of computer color graphics for tasks such as textile and garment design, and pattern layout.

In Metzingen, West Germany, designers at the weaving mill Gaenslen und Voelter have found that computer-aided textile design increases productivity while letting them explore new patterns more easily. Gaenslen und Voelter employs about 350 people and produces between 10,000 and 12,000 meters of fabric per day, primarily for men's and women's wear.

Follows Normal Design Sequence

Designers at Gaenslen und Voelter use Tek's 4125 and 4115B Color Graphics Workstations to view their computer-generated textile designs (Figure 1 shows a typical screen display). An IBM PC/AT™ serves as host. Software for the system was developed by Alexander Herrmann, one of the company's owners, in cooperation with computer scientists at the Polytech of Reutlingen. The user interface takes advantage of workstation features to make the system very easy to learn. With common key functions, menu control and extensive dialog interactivity, even designers without previous computer experience can master the system within a day or two.

Gaenslen und Voelter's CAD/CAM system closely follows the sequence of steps ordinarily used for manual textile design. The designer enters his idea into the computer, specifying parameters such as yarn and thread variations and weave, warp and weft sequences. He immediately receives a display that he can then modify if he chooses. Once the designer is satisfied with the pattern, the system automatically delivers an optimized pattern ready for production. The system also generates the production orders, a clerical task that the designer previously had to perform.

Color Features Play Key Role

Naturally, for weaving designers, the color graphics capabilities of the Tek workstations play a key role. Of particular importance, according to Herrmann, is the ability to display 256 colors from a palette of 16 mllion shades. Designers can create a new palette of 256 yarns as fashion trends change with the seasons. Also important for Gaenslen und Voelter are the workstations' high resolution and non-flickering image display, processing speeds, and autoconvergence feature, which, at the press of a button, ensures color fidelity.

Gaenslen und Voelter's designers have found that the workstations' Hue-Lightness-Saturation (HLS) color system allows colors to be specified more precisely than with yarn definition programs, in which the designer would define and enter yarns visually, and thus, unfortunately, subjectively.

From Idea to Reality in Nothing Flat

Before Gaenslen und Voelter put its CAD system into place, new fabric patterns had to be produced manually or with the help of machines before they could be shown to customers or salesmen, or brought to market. Thus, with a range of fashion-oriented products, the company faced considerable expense for pattern generation, and designers were often limited in their freedom to explore new designs. In addition, designers faced a problem common to artists in any discipline: how do you communicate a design when it exists only in your head? Gaenslen und Voelter says its CAD/CAM system has reduced the conflict between pattern production costs and the free design of patterns, and also allows new designs to be visualized and shared with others in a matter of minutes.

The ease with which new designs can be implemented is especially helpful when custom orders need to be generated. Working with a designer, a customer can watch the procedure and help create 'his' special pattern. In some cases, customers have created individual patterns directly on the color screen and ordered production coupons on the spot, completely bypassing the need for a sample. The CAD system is also useful at strategic planning meetings between designers and sales personnel. During these meetings, designers can display their ideas realistically, and special requests from salesmen for new designs can be realized in as little as 10 or 15 minutes.

In addition to aiding in the development process, the CAD system is also used to archive designs so that designers can refer to them later.

Both Productivity and Creativity Enhanced

As a result of the CAD system, Gaenslen und Voelter has realized a number of benefits, both for management and designers. The production costs of patterns and samples has been reduced by one third. Designers find that the system relieves them of a number of clerical chores, and that they can generate designs faster, more easily, and with less expense. As a result, designers have more time for creative work. And they are freer to experiment with vague or risky ideas that, under a manual system, might have been dropped.

According to Herrmann, the next logical step for Gaenslen und Voelter will be a fully integrated system with direct linkage between design and production, including such steps as automatic card punching and controlled warp and weft sequence.

Jacquard would no doubt be pleased.

Photograph courtesy of Gaenslen und Voelter.

IDG Leasing Programs: Flexibility in Equipment, Financing

Your department needs new terminals and copiers, but there's no money in the budget for new equipment purchases.

You need to purchase graphics equipment but are concerned that further technological advances render it obsolete.

You want to explore a new technology—perhaps artificial intelligence or automatic drawing scanning systems—but you aren't sure how the new technology fits with your existing systems and products.

In all three of these cases, IDG's leasing programs offer a solution:

Budget constraints are eased. Leasing increases your company's purchasing power because lease payments are generally made from operating or expense funds rather than capital funds. Capital and credit lines are still available for use. Leasing often provides tax and depreciation savings as well.

Concerns over obsolescence are minimized because many IDG lease programs include excellent upgrade features. Often, you can return a currently leased product and lease newer equipment at a discount.

Testing the waters of a new technology is much less risky when you can explore it via leased equipment.

Three Standard Programs

Tek's Information Display Group offers a variety of lease programs designed to make it easy and economical to acquire Tek graphics equipment. Three standard lease programs are offered:

The *Comprehensive Lease Program* offers maximum flexibility. Half of the lease payments apply toward the purchase of the equipment (up to 75% of the original price), and you can trade up to new products at a discount. Maintenance is provided by Tek at no extra charge. And you can cancel the lease early by paying 25% of the remaining payments.

The *Ownership Lease* provides a way to purchase Tek equipment via monthly payment plan. There's no down payment, no balloon payment at the end, and the title is automatically transferred to the customer at the end of the lease term.

The *Basic Lease* is, as the name suggests, our basic, "no frills" lease. It lets you access Tek graphics equipment at the lowest monthly rates.

Each of the three leases offers terms of one to four years. And with all three leases, standard quantity discounts apply.

Specialized Programs

In addition to the three standard programs, IDG also offers a number of specialized lease plans. For example, the *New Product Evaluation Lease* (also known as the "try and buy" program), and *90-Day Lease* are ideal for evaluating a new technology or a new product. Depending on the specific program, either 100% or 90% of the lease payments can be applied toward the purchase of the product. Other special leases are designed toward educational institutions, government agencies, and companies who have not yet established their credit history. There are even *Custom Leases* that can be custom-tailored to individual requirements.

TLC Program Trades Old Graphics Equipment for New Lease Credit

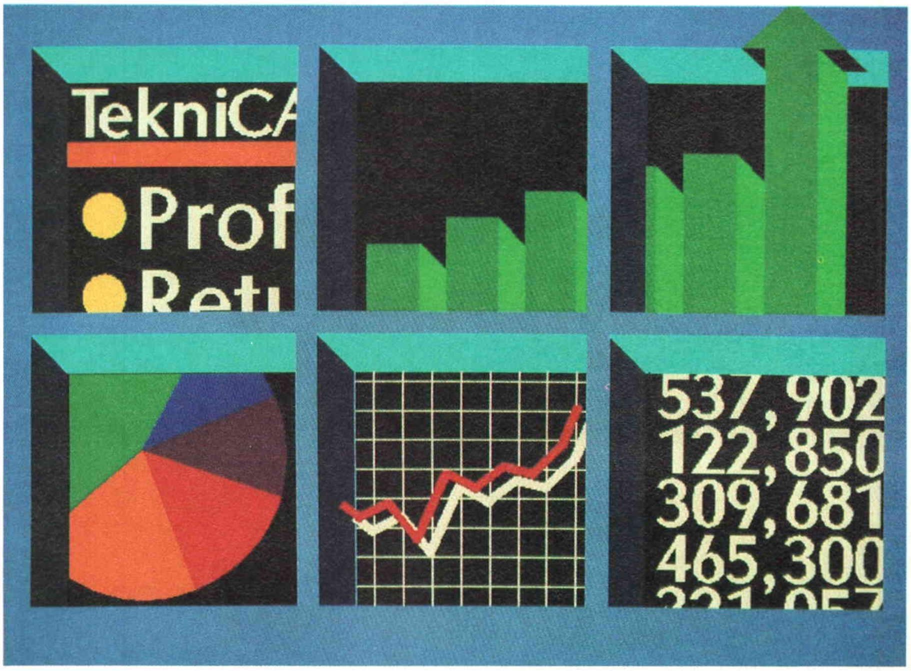
If you'd like to upgrade your DVST terminal by moving to color, now is the time to do so. For a limited time, Tek's Information Display Group (IDG) will accept for trade-in *any* previously purchased IDG end-user equipment. In return, you can lease new IDG graphics equipment at sharply reduced lease rates. The new program, called *Trade for Lease Credit* (or TLC), also accepts selected graphics equipment from other vendors. All equipment must be in good operating condition.

For every product traded in, you can lease several products at a discount. This means you can economically increase the number of graphics stations in your work areas.

The TLC program is a temporary special program subject to cancellation. If you'd like further information about graphics equipment trade-ins, contact your local Tektronix field office.

Tektronix escape sequence files can now be incorporated into TekniCAD files. Version 2 also supports overhead transparencies, 35mm slides and other hardcopy output, including the new interface to Matrix QCR and PCR film recorders.

PLOT 10® TekniCAP Adds New Input, Output Features



by Eleanor Mathews
Program Manager,
Presentation Graphics Software
Terminals Division
Tektronix, Inc.
Seattle, WA

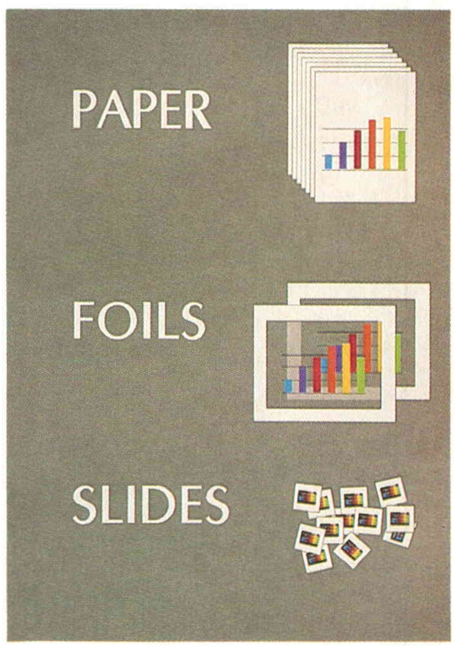
Compatibility is a key word in computer graphics. Users want to mix and match hardware and software to create systems that will work efficiently and satisfy unique requirements. People want to create individual combinations of input and output devices, terminals, and graphics programs. This depends on having effective links between each component and on having a foundation of compatibility among the parts. TekniCAP Version 2.0 introduces some important new links that offer increased compatibility for configuring and using a presentation graphics system.

PLOT 10 TekniCAP is a software product for making presentation graphics. Freehand drawing, graphing, data management, type fonts, color control and output features are all part of TekniCAP. Users select from a series of menus to make and store images. These selections and images may rely exclusively on the TekniCAP default settings for location, size, color and type font, or they may result from user-specified alternates. This combination of built-in guidelines and user options has contributed to TekniCAP's acceptance as an all-purpose presentation graphics package.

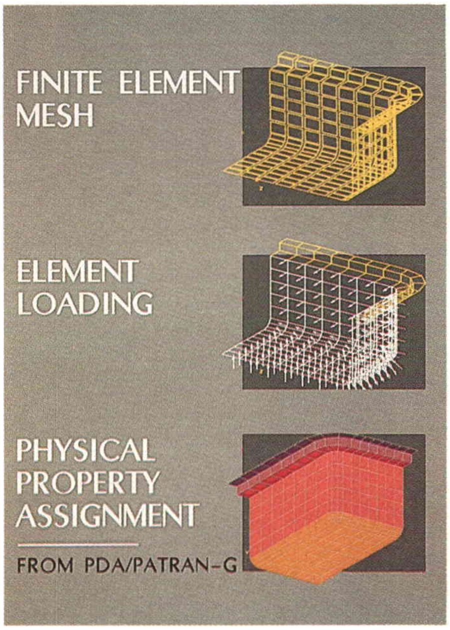
Outside Files

Many TekniCAP users apply their computer systems to multiple tasks. These people use TekniCAD for 2-D drafting, SAS®, ISSCO® or PDA software for sophisticated data analysis, or they may have written their own graphics software for some other purpose. Whatever kinds of pictures are being generated, it is not unusual to want to use them to present findings, proposals, designs or other aspects of the work. These presentations are made to clients, management, professional groups and project teams. However, not all graphics software is suited to making presentation materials.

That's where TekniCAP comes in. It is specifically designed to make overhead transparencies, slides and other hardcopies that are attractive, clear and highly readable. By bringing images created on the outside into TekniCAP, users can make presentation materials with a minimum of fuss. This input enhancement offers the capability of combining non-TekniCAP graphics files within TekniCAP. Outside image files in Tektronix escape sequence format are called from the INPUT/OUTPUT module as TekniCAP objects. Outside graphics can be inserted into TekniCAP display files at any point, even when a view is in effect. With the outside file feature, TekniCAP can be used to enhance a technical



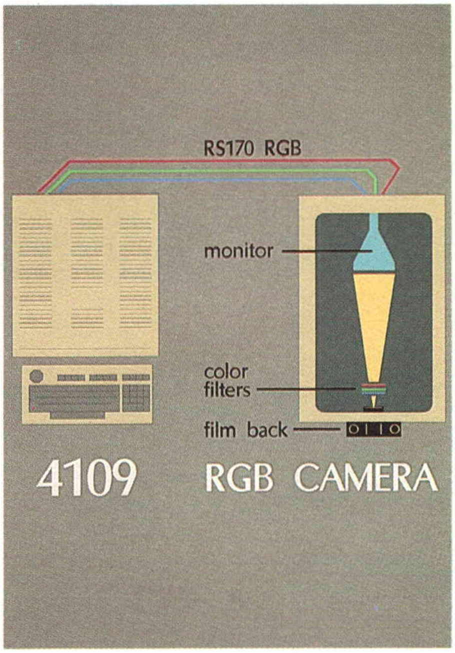
drawing—to make it more readable with improved type, to alter colors, to add elements for clarification or to combine the drawing with other graphics. With technical graphics, the inclusion of a logotype, annotations, a simple graph or small diagram will often make a difference in reaching an audience, and now those inclusions are possible. The outside file capability makes TekniCAP an ideal add-on to an existing system.



Since TekniCAP now links a range of software products together and in turn links those to its hardware family, users have increased flexibility in configuring systems and greater ease in using them. The TekniCAP hosts are Tektronix 4170 Local Graphics Processing Units, 4115 and 4120 Series Graphics Workstations running in stand-alone mode, 6130 Intelligent Braphics Workstations, and DEC® VAX™ and MicroVAX™ computers. Terminals supported by any of those hosts include the 4107, 4107A, 4109, 4109A, 4115, 4125, 4128 and 4129. Output devices linked to TekniCAP are Tektronix 4691, 4692, and 4695 color copiers, with or without a 4510 or 4510A graphics rasterizer. The color copiers put TekniCAP images on paper or overhead transparency media.

35mm Slides

In addition to the color copiers, TekniCAP now addresses photographic devices for making 35mm slides. This is accomplished either through the RS-170 video signal or an RS-232 connection, depending on resolution requirements. RS-170 serves imaging at screen resolution. The 4109A terminals provide RS-170 out, and the 4115B/4120 Series Workstations can be configured to provide RS-170 with the addition of Option 7, the Camera Interface. Since the RS-170 signal carries the red, green and blue components of an image, it is also referred to as the RGB signal. This signal can send an image to a remote monitor, such as that found in an RGB frame camera. The camera system picks up that signal on an internal, flat, monochrome monitor. The glass screen is fully enclosed to prevent light reflections from occurring. A 35mm camera is mounted in front of the monochrome monitor and exposes film to images as they are displayed. Since the images are displayed in monochrome, color filters are used to put color back into the pictures. Red, green and blue components of the RS-170 image are displayed sequentially with corresponding red, green and blue filters in front of the camera.



By enclosing the photographic process inside a light-free case, RGB frame cameras eliminate the glare problem often encountered when taking pictures of terminal screens directly. Since human vision compensates for glare subconsciously, frequently glare isn't seen while making pictures but appears all too clearly after slides have been processed and mounted.

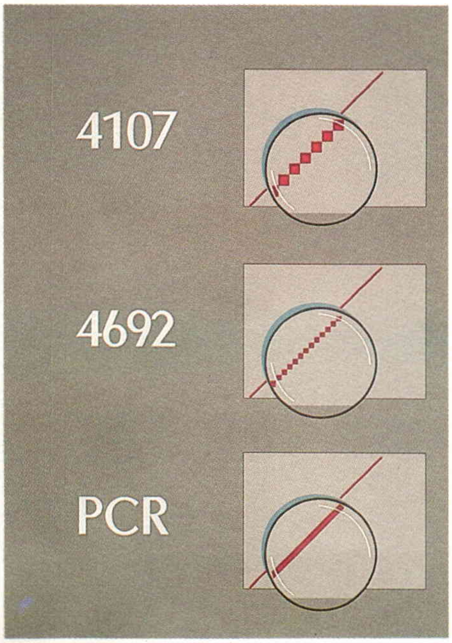
Other problems are avoided by using RGB cameras. The specially-made flat monitor eliminates focus difficulties related to the curvature of most terminal screens, so straight lines around the edge of a picture remain straight and evenly focused, rather than curved and blurry near the corners. RGB cameras cannot easily compensate for the difference in aspect ratio between terminal and slide, and this is a distinct drawback for people who want to mix TekniCAP slides with photography. Thirty-five-millimeter slides are 3:2, horizontal to vertical measures, but this is not the same shape as terminal screens. Consequently, slides made with RGB cameras do not "match" other slides. On the plus side, exposure is controlled electronically on the front panel of most RGB cameras and is calibrated to a variety of film types. Since the video signal is RGB and color film dye is also RGB, no color translation is necessary and very little color shift is noticed from screen to film. The color range of film is such that virtually all terminal colors can be duplicated on film in clear, even tone without dots, dithers or patterns.

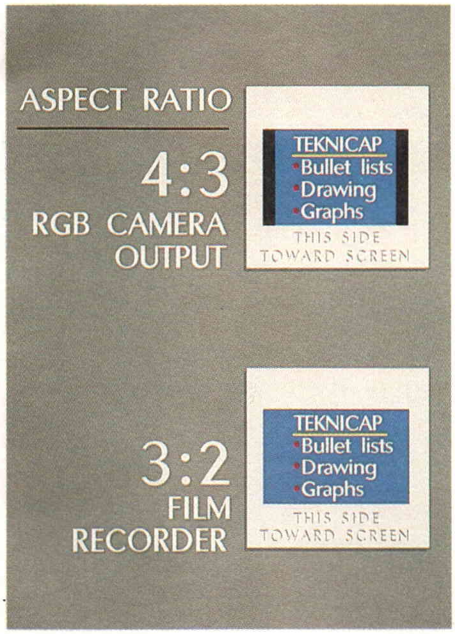
Many suppliers make and sell RGB cameras, and features differ from one model to another. Matrix Instruments and Dunn Instruments both make cameras that have been successfully tested with Tektronix equipment. When considering a specific model, it is important to match resolution and camera video rates to the resolution and video rate of the terminal used to send pictures.

Digital Film Recorder Output

Although RGB cameras are preferred to direct photography of a screen, they cannot increase resolution above that of a terminal. When greater resolution is called for, TekniCAP can send images out through an RS-232 port on the host to a Matrix Instruments digital film recorder. Film recording devices can expose slides at a resolution of 4096 points, horizontally. This ultracrisp definition approaches the resolution of the film itself, and images are free of the dreaded jaggies and other hallmarks of their digital origins. Matrix manufactures both the QCR and PCR film recorders. Either is compatible with TekniCAP.

The PCR and QCR bring all the benefits of RGB cameras to TekniCAP users—and then some. Nothing matches the brilliance and clarity of slides made on a film recorder, and unlike the RGB camera output, pictures are correctly shaped in the 3:2 aspect ratio of 35mm photography. The TekniCAP pictures accompanying this article were all made with a Matrix PCR.





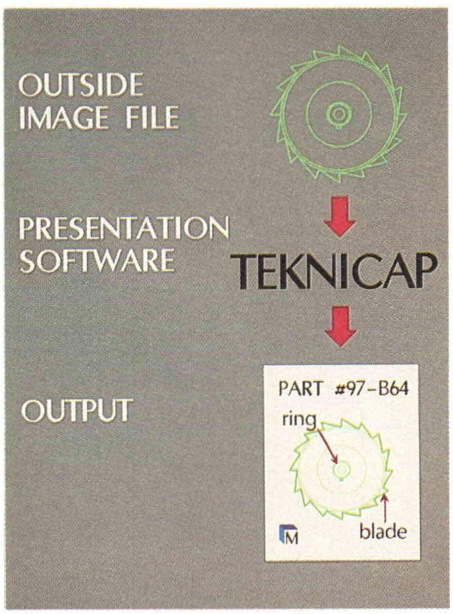
How It Works

Regardless of the terminal used, all TekniCAP display files are stored in 4096×3072 coordinate space so they are compatible with film recorder output, which is also 4096. The image is traced very rapidly on the film recorder monitor with a single beam. It is drawn raster by raster and color by color, having been converted from geometric graphic elements in the TekniCAP display file to high resolution rasters. The same color filtering system is used as with RGB frame cameras.

Since it is practical to expose 24 or 36 frames in a single session, TekniCAP allows an entire schedule of images to be put in order and sent together. To avoid tying up the terminal while this process takes place, TekniCAP spools the files to the host. This spooling allows continued use of the terminal, so more TekniCAP pictures can be made without waiting. Additionally, files can also be spooled to the 4510 and 4510A rasterizer.

Not all software includes such output and media flexibility as TekniCAP. That's the reason many people want to combine TekniCAP capabilities with their other graphics programs. The ability to bring outside image files into TekniCAP, then route those files to any or all of TekniCAP's output devices provides a powerful link. The fact that the output activity occurs in the background through spooling software is simply icing on the cake.

Think of the TekniCAD user presenting a proposal to management. He or she calls a drafted image into TekniCAP, frames the picture or otherwise adds graphic enhancement with the DRAW features, adds titling and annotation with TekniCAP's TYPE module, makes any necessary color alteration in COLOR, sets VIEWS as necessasry, then saves the file in INPUT/OUTPUT when it is complete. Once the graphic is saved, it can be directed to any of the TekniCAP output devices. Instead of a hodgepodge of different pictures, all of the visuals in a presentation can be coordinated and well-planned.



A presentor can project overheads to management and slides to the client. Either way, the presentor can address the audience with appropriate media and still provide color hardcopy handouts for future reference. Since TekniCAP includes a setting for multiple hardcopies, as many as 100 handouts can be made with a single command.

If the outside files do not need any alteration or enhancement, they may be saved and sent directly to output without change. Some people are using TekniCAP solely as an output driver and sending their images straight through to film. Software which does not support film or other hardcopy output can be used in harmony with TekniCAP.

Compatibility is the strength of TekniCAP version 2.0. By combining a powerful new input feature with an equally powerful set of output features, people can use their computer graphics systems to greater advantage. Tailored systems that meet specific requirements are economical and efficient. Systems can be expanded as needs grow; shared resources can be maximized. For the person making presentation graphics with TekniCAP, this means that clear, crisp images can be routed to output devices that will most successfully meet budget requirements and address audiences with effective materials.

Illustrations for this article were prepared using PLOT 10 TekniCAP.

At CAD Associates, Tek's 4991 Autovectorizer, 6130 workstation, 4125 terminals and PLOT 10 TekniCAD help with an around-the-clock drafting workload.

Tek Products Boost Productivity for Texas Drafting Firm

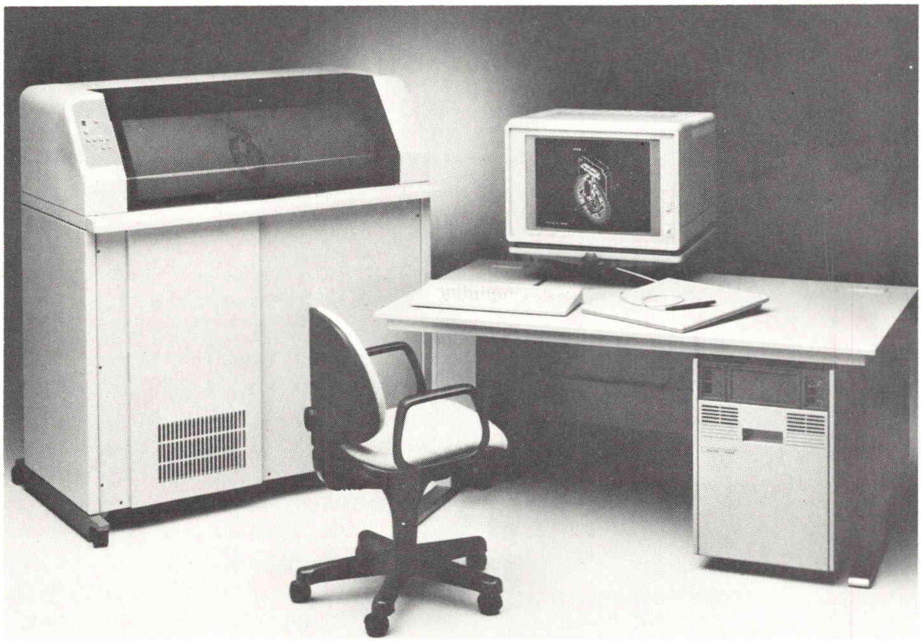


Figure 1. Tektronix 4991S1 System.

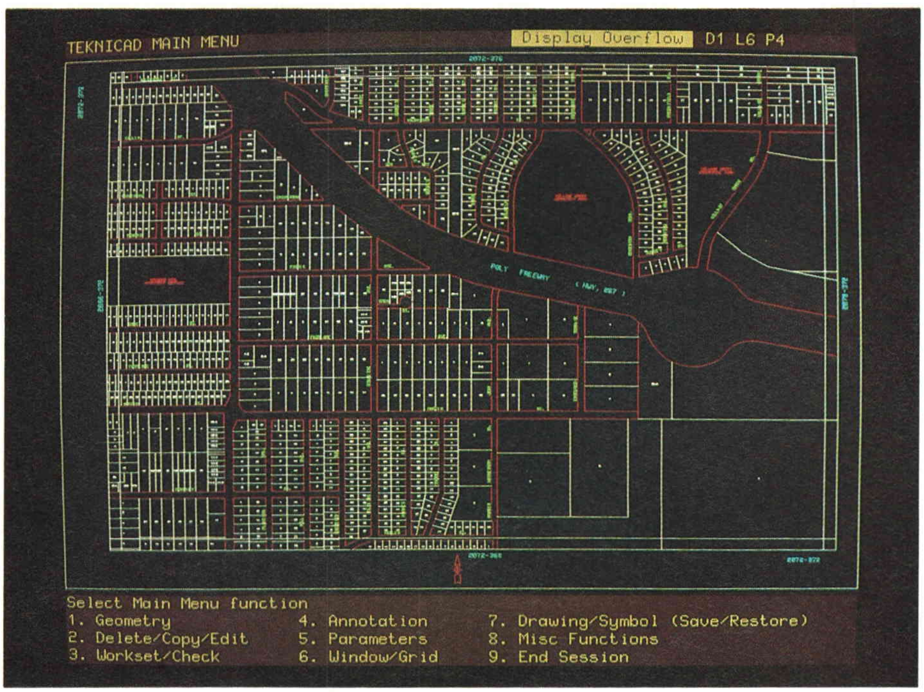


Figure 2. TekniCAD's functions simplify the creation of cadastral maps. One function divides a rectangular area into a number of equal-sized blocks. A macro performs sequential numbering; the drafter simply moves the screen cursor from one lot to another and presses a button, and each lot is numbered correctly.

When the City of Fort Worth needed to have water and sewer maps recreated, it turned to CAD Associates, a drafting service bureau that operates three shifts a day, six days a week, and handles everything from architectural and mechanical drafting to mapping of entire counties. And CAD Associates turned to Tek, using Tektronix equipment and software to do original drafting and to digitize existing paper drawings for entry into CAD databases.

Started with 4050 Series

"We've been with Tek since the days of the 4050 Series," says Tex Norwood, executive vice president of the Burleson, Texas, company. "We had the first version of TekniCAD on the 4116, and we've just kept up with Tek. We ran it using Local Programmability on the 4115 terminal. We tried the 4125, and the speed was so much better we just had to go to it. Then last year we moved to the 6130 workstation with an 80-megabyte hard disk."

Norwood has been happy with the combination of a 6130 and 4125 terminals. "We tried the 6130 and found it could do a lot more than we thought it would," he says. "The performance is very acceptable, even with four 4125s attached. The time savings compared to LoPro is tremendous—it's many times faster to bring up a drawing or to save a drawing. And the dependability has been extremely good. It's just a very cost-effective solution."

Productivity increased significantly when the drafting task moved onto the 6130, according to Norwood. "One of our drafters looked at the before-and-after figures and found that her output had doubled in her first two weeks on the 6130," he says. "That's particularly noteworthy when you consider the normal adjustments you have to make to any new system." He gives credit for the productivity improvement to both the 6130 and TekniCAD: "We got the first 6130 the same time as Version 8 of TekniCAD, which added a number of enhancements to TekniCAD and also boosted our throughput."

Time Savers

Another big time-saver for CAD Associates are TekniCAD's keyboard macros—user-defined miniprograms that can greatly reduce the keystrokes required to perform a task (see Figure 2). When Norwood has a contract for several maps, he sets up a macro to create the customer's title block. The macro runs a mini-program that prompts the operator to enter the sheet number, date, North arrow location and rotation, and school district. Another macro sets the level, pen number, and line type. A single macro saves them 111 keystrokes each time it is used. In addition, for large projects that may involve several drafters, keyboard macros help ensure consistency.

The system's ability to handle large drawings is also a plus for CAD Associates. "Many of our drawings take up to 500 or 600K bytes. We don't have to break them down at 400-500K like we did before," Norwood comments.

Scanning Existing Drawings

CAD Associates further expanded its capabilities last year by adding Tek's 4991S1 Graphic Input Workstation, which is built around a 4991 Autovectorizer and a 4125 Color Graphics Workstation. The autovectorizer scans and vectorizes original documents, and special Graphic Structuring Software on the 4125 is used to edit the scanned information and create intelligent CAD data structures. These structures can then be transferred to a number of CAD host systems. (See the accompanying article, "Moving Old Drawings into New CAD Systems.") Norwood estimates that the firm uses the autovectorizer for 30 percent of its contracts, particularly for mapping and mechanical engineering.

CAD Associates uses the 4991 system and PLOT 10 TekniCAD to produce maps that include a wide variety of information: property lines, lot numbers, ownership information, streets, sewer lines, utility lines, school districts, topography, and much more. Putting the information into a computer database makes the information easier to access. Different types of information can be assigned to different layers, and the layers can be plotted individually or in combination to produce map separates—for example, sewer lines on one layer, roadways on another. When one type of information changes, only the affected layer has to be revised.

For a recent project, CAD Associates scanned four subdivision maps to create one tax-ownership (or "cadastral") map. One of the "original" maps was a photocopy. The other three originals were blueprints: one was torn and barely useable, one had a variety of pencil and ink markings, and one was in fairly good condition.

Figure 3. CAD Associates Vice President Tex Norwood and CAD operator David Allen examine a map to be scanned.

Figure 4. Drafters Noris Reed and Sandy Bell compare an existing map with the structured version produced with the 4991S1 system. Both operators commented on the systems's accuracy and ease of use.

Three of the original maps were drawn at different scales, and the scale of the fourth map was undefined.

After using the 4991 Autovectorizer to scan and vectorize each original map, CAD Associates drafter David Allen (Figure 3) used the Graphic Structuring Software (GSS) on the 4125 to edit the scanned data. Allen used a GSS realignment function to make each subdivision the same scale, using registration points and a "best fit" software option to correct for media distortions. He also deleted extraneous markings, including some text.

Next, Allen used GSS to send the map as structured CAD entities to the 6130 workstation. There, using TekniCAD, he moved the four pieces of the map around the screen until they were properly aligned. He put lot lines, text and roads on separate levels, so that each level could be plotted separately. Finally, he added text, set the rotation relative to true North, and made final adjustments to boundary lines.

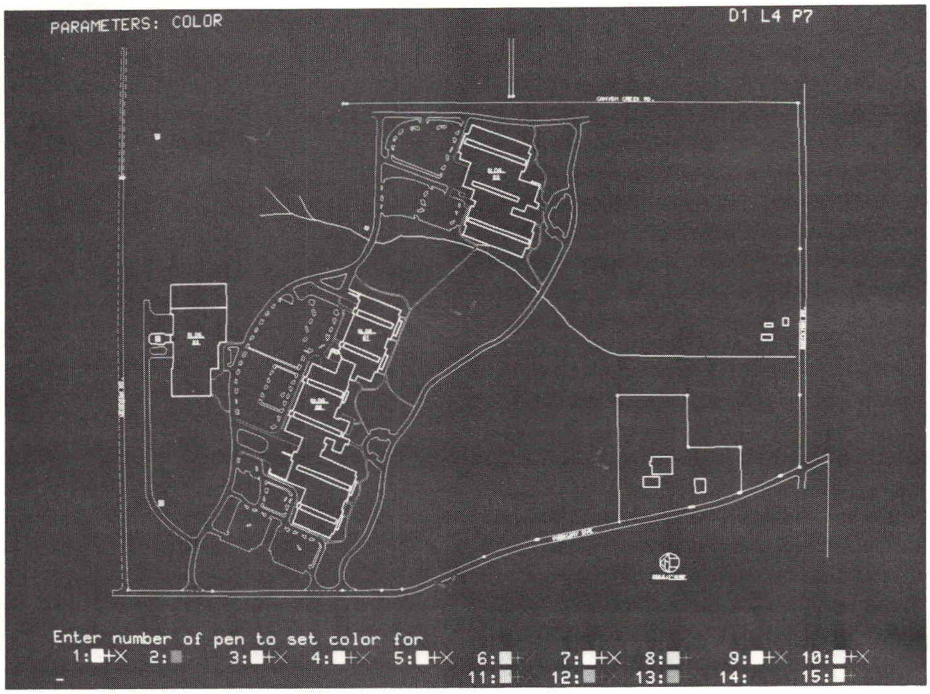


Figure 5. Irregular lines such as creeks and other waterways can be quickly scanned.

Enthusiastic

Norwood and the CAD Associates drafters are enthusiastic about the 4991 system. "Time and absolute accuracy are the most important factors in drafting," Norwood says. "Now (with the 4991S1 system), we have both."

The improved productivity has paid measurable results. Since installing the 4991S1 system, CAD Associates has re-quoted several jobs at 25-30% below the original bid and captured business that would have gone to a competitor. Turn-around time has also been reduced: one customer who contracted to receive three maps per month is now receiving eight. Seeing the results, the customer is sending additional business to CAD Associates.

Drafter Sandy Bell (Figure 4) notes that "you can do more with the scanner than with manual digitizing. The scanner is a lot quicker and more accurate—and your arm doesn't get tired." Bell cites the example of creeks, which ordinarily require the entry of a large number of data points—a tedious and time-consuming process when the map is manually digitized. When a map containing a creek is scanned, however, all the data for the creek is located accurately on the display and can be added to a data file in seconds (Figure 5). The drafter can use one of GSS's line-following cursors, which locks onto a line and allows the drafter to trace over an item quickly.

Noris Reed, a newer member of CAD Associates, appreciates the menu-driven interface on both GSS and TekniCAD. "It's so easy—the prompts are all on the screen," she says. "It all comes together."

Norwood particularly praises TekniCAD and UTek for offering both sophisticated features and ease of use. "You're not going to master every feature and get bored with them in the first month," he says. "There's always something more to explore, and always more features being added to TekniCAD. But at the same time, they're genuinely user-friendly. The Casual User Interface on the 6130 is ideal for people that don't want to learn the ins and outs of Unix. The terminology and organization of TekniCAD, and the way the screen is set up, all make sense to a draftsman. And you move around in the program fast—you don't waste time sitting around."

As for the 4991, Norwood sums up his evaluation in a sentence: "This machine is the most fantastic innovation in drafting since the computer."

Moving Old Drawings into New CAD Systems

by Ken Hunkins
Graphics Printing
and Imaging Division
Tektronix, Inc.
Wilsonville, OR

In heavy industry, where products generally have a long lifetime, and in fields such as cartography and architecture, a "new" project is often based on existing designs and drawings. Therefore, before professionals in these fields can reap the benefits of computer-aided design, they must first get their older drawings into the new CAD database.

This has generally been accomplished via one of two methods. In some cases, the data is tediously redrawn from scratch, using the new CAD system. This not only requires a highly skilled operator, it also ties up the resources of the CAD system—often for months at a time—with non-design work. In the second method, the old drawing is hand-traced by an operator using a large digitizing tablet. This method does not require as highly skilled operators, but it is still time-consuming and somewhat error prone.

Automatic scanning devices such as Tek's 4991S1 Graphic Input Workstation offers a third alternative. Figure 1 summarizes the 4991S1 process. The 4991 Autovectorizer scans drawings and produces a vector file, which is transferred to a 4125 terminal. There, a drafter uses special Graphics Structuring Software (GSS) to interactively structure the scanned data. Finally, Host Interfacing Software allows the output from the structuring process to be sent to a variety of host CAD systems, looking just as if it had been created using the CAD system directly. Three CAD packages are currently supported: CADAM™ Versions 19 and 20, Computervision's CADDS 4X, and Tek's own PLOT 10® TekniCAD. Documents to be scanned can be A through E size, in ink, pencil, transparent film, paper, blueline, or photographic line art.

Skill requirements for the 4991 system are modest. The autovectorizer operator only has to load the drawing and identify quality, size, input media and output format. The GSS structuring can be done by a person with basic drafting skills.

The Scanning Process

The autovectorizer hardware consists of a drum, an optical scanner, a carriage that moves the optical scanner, and an electronics module. The drum holds the document as it rotates during the scan. The optical scanner includes a lamp and lens unit and a 128-element Charge-Coupled Device (CCD) array. As the optical scanner moves across the document, the lamp-and-lens unit focuses light on the document while the CCD array "reads" the intensity of the light reflected from the document.

During scanning, the CCD array "looks at" scan lines—rows of discrete points parallel to the vertical edge of the document on the drum. As the drum rotates, the scanner examines 128 scan lines—a swath—approximately a centimeter wide. After each swath is read and processed, the optical scanner steps to the next position along the drum and scans the next swath.

As the document passes the optical scanner, the scanner sends a series of analog electrical pulses to the electronics module. Each pulse represents a point, or pixel, on the scan line. The electronics module converts the pulses to pixel data, consisting of binary numbers representing the written or unwritten state of each pixel. The electronics module then converts the pixel data to vector data and transmits the vector data to the 4125. Each swath is processed as soon as it is scanned, so by the time the scan is completed, all or nearly all of the processed data has been sent to the terminal.

Structuring the Data

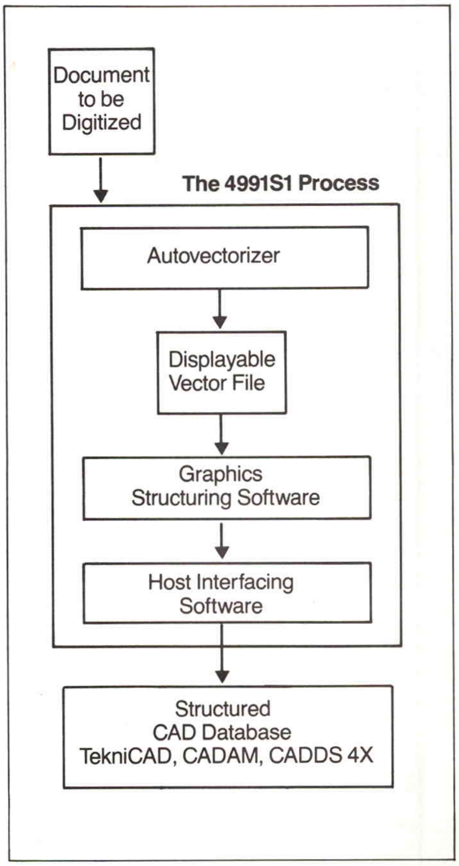
In structuring the data, the drafter selects functions from a menu displayed on the 4125 screen, using the graphics tablet and stylus. Function names are taken from drafting terminology, making the structuring an easy task for drafters without expertise in computer-aided design.

The drafter can replace vectors with graphics primitives, including lines, arcs, circles, text, symbols, ellipses, arrows and dimensions. She or he can also mask or delete portions of the drawing. Either standard or user-defined attributes can be assigned to features on the drawing. Attributes include layer number, pen number, color, line width, line style, text font, and character slant.

A number of other GSS features make the structuring process easy and efficient. Viewing capabilities such as zoom, pan, original view and previous view allow quick and accurate work on any area of the drawing. Transformation functions can align the drawing to one or more specified points or align all nodes on the drawing to a user-defined grid.

Time of Transition

The next decade will be a CAD transition period, with engineering and scientific industries converting from manual design techniques to computer-aided design systems. By efficiently and cost-effectively moving vast numbers of hand-drawn documents into the CAD database, automatic input systems such as the 4991S1 will open the benefits of CAD to new groups of users.



Programming Tips

Converting to 4100/4120 Series Numbers

by Steve Olin
Technical Support
Tektronix, Inc.
Newport News, VA

Here's a basic program for converting character to integer, integer to character, decoding of character reports, etc. The program is relatively simple to run. When it is executed, just enter the appropriate menu selection number. When entering X,Y coordinates, you can separate them by either a space or a comma.

When writing the program, I found no easy way to display a space (ADE 32), so when an answer contains a space, the program prints at the bottom of the screen that the answer contains a space and what location of the string the space is (for example, "There is a space in location 2").

```
      character tmpstr*6,lf*1,cr*1,sign*1,off*4
      integer Kcmdbf(72),Kntcmd,jchars(3),hiyloy(5)
      common /sticmd/ Kcmdbf,Kntcmd
      off=char(27)//'[0m'
      cr=char(13)
      lf=char(10)
      call system('stty cbreak')
      call llinit(jterm)
    1 call lldump
      print *,'^[[%!1^[[2J^[[1;1H'
      print *,lf,lf
      print *,'  1. Encode Integer parameter'
      print *,'  2. Decode Integer parameter'
      print *,'  3. Decode Integer Report'
      print *,'  4. Encode 32 bit Integer parameter'
      print *,'  5. Encode Real parameter'
      print *,'  6. Encode XY coordinate'
      print *,'  7.       <<<Stop>>>'
      print *,lf
      print 2
    2 format('           Which one? ',$)
      i=getc(j)
      go to (10,20,20,40,50,60,70),j-48
    3 format('  Position ^[1m',i1,'^[[0m contains a ^[[1mspace^7^[[0m')
      go to 1
*****************************************************
*          1. Encode Integer parameter              *
*****************************************************
   10 print *,cr,lf
      print 11
   11 format('  Enter Integer value: ^[[1m',$)
      read *,iarg
      print *,off
      call intlx(iarg)
      print *,lf
      write(6,12)'  The encoded value for ^[[1m',iarg,'^[[0m is: ^[[1m'
   12 format(a,i5,a,$)
      write(6,'(3(a1),$)')(char(Kcmdbf(K)), K=1,Kntcmd-1)
      print *,cr,lf,off
      do 13 n=1,Kntcmd-1
        if(Kcmdbf(n).eq.32)write(6,3)n
   13 continue
      call pause
      go to 1
*****************************************************
*          2. Decode Character parameter (ioff=64)  *
*          3. Decode Character report (ioff=32)     *
*****************************************************
   20 sign='+'
      print *,cr,lf
      tmpstr='@@@'
      print 21
   21 format('  Enter Character sequence: ^[[1m',$)
   22 i=getc(ichr)
      if(ichr.ne.10)then
        tmpstr(1:1)=tmpstr(2:2)
        tmpstr(2:2)=tmpstr(3:3)
        tmpstr(3:3)=char(ichr)
        go to 22
      else
        print *,off
        if(j.eq.50)ioff=64
        if(j.eq.51)ioff=32
        call Kam2as(3,tmpstr,jchars)
        hii1 = jchars(1) - ioff
        hii2 = jchars(2) - ioff
        loi = jchars(3) - 32
        ival=hii1*1024 + hii2*16 + mod(loi,16)
        print *,lf
        l=ichar(tmpstr(3:3))
        if((l.ge.32).and.(l.le.47))sign='-'
        if(tmpstr(1:1).eq.'@')tmpstr(1:1)=char(32)
        if(tmpstr(2:2).eq.'@')tmpstr(2:2)=char(32)
        write(6,'(4(a),i5)')'  The Integer value for ^[[1m',tmpstr,
     &  '^[[0m is: ^[[1m',sign,ival
      endif
      print *,off
      call pause
      go to 1
*****************************************************
*          4. Encode 32 bit Integer parameter       *
*****************************************************

   40 print *,cr,lf
      print 41
```

```
   41 format('  Enter 32 bit Integer value: ^[[1m',$)
      read *,iarg4
      print *,off
      call intclx(iarg4)
      print *,lf
      write(6,42)'  The encoded value for ^[[1m',iarg4,'^[[0m is: ^[[1m'
   42 format(a,i10,a,$)
      write(6,'(6(a1),$)')(char(Kcmdbf(K)), K=1,Kntcmd-1)
      print *,cr,lf,off
      do 43 n=1,kntcmd-1
        if(Kcmdbf(n).eq.32)write(6,3)n
   43 continue
      call pause
      go to 1
*******************************************************
*          5. Encode Real parameter                   *
*******************************************************
   50 print *,cr,lf
      print 51
   51 format('  Enter Real value: ^[[1m',$)
      read *,rval
      print *,off
      if(abs(rval) .lt. 0.00001)then
        jexp=0
        jmant=0
      else
        jexp = ifix(alog(abs(rval))/alog(2.0))-14
        jmant = ifix(rval*float(2**(-jexp)))
      endif
      call intlx(jmant)
      call intlx(jexp)
      print *,lf
      write(6,52)'  The encoded value for ^[[1m',rval,'^[[0m is: ^[[1m'
   52 format(a,f10.2,a,$)
      write(6,'(4(a1),$)')(char(Kcmdbf(K)), K=1,Kntcmd-1)
      print *,cr,lf,off
      do 53 n=1,Kntcmd-1
        if(Kcmdbf(n).eq.32)write(6,3)n
   53 continue
      call pause
      go to 1
*******************************************************
*          6. Encode XY coordinate                    *
*******************************************************
   60 print *,cr,lf
      print 61
   61 format('  Enter X,Y coordinates: ^[[1m',$)
      read *,ix,iy
      print *,off
      hiyloy(1)=iy/128+32
      hiyloy(2)=mod(iy,4)*4+mod(ix,4)+96
      hiyloy(3)=mod(iy/4,32)+96
      hiyloy(4)=ix/128+32
      hiyloy(5)=mod(ix/4,32)+64
      print *,lf
      write(6,62)'  The encoded value for ^[[1m',ix,',',iy,
     &'^[[0m is: ^[[1m'
   62 format(a,i4,a1,i4,a,$)
      write(6,'(5(a1),$)')(hiyloy(K), K=1,5)
      print *,cr,lf,off
      do 63 n=1,5
        if(hiyloy(n).eq.32)write(6,3)n
   63 continue
      call pause
      go to 1
*******************************************************
*          7.      <<<Stop>>>                         *
*******************************************************
   70 call system('stty -cbreak')
      print *,cr,lf
      end
*******************************************************
*         Subroutine to pause between selections.     *
*******************************************************
      subroutine pause
      character *1,cr,lf
      cr=char(13)
      lf=char(10)
      print *,cr,lf,lf
      print 10
   10 format('        Hit any key to continue...',$)
      i=getc(j)
      return
      end
```

Tip for New 4100 Series Users: VT-100™ Applications Disable Active Key Definitions

by Ginny Beyer
Technical Support
Tektronix, Inc.
Philadelphia, PA

If you're running a VT-100 application and have programmed any keys on your terminal, you may have noticed that your key definitions seem to "disappear" when you go into your application. This is because a VT-100 application usually sends a CODE EDIT command to the terminal; the CODE EDIT command causes a Tektronix 4100 terminal to emulate a VT-100 terminal.

VT-100 emulation disables any previously defined key definitions, because a VT-100 (or a 4100 in VT-100 emulation) uses certain keys for cursor positioning functions. Your key definitions are still in the terminal; they are just temporarily disabled and inactive. (Those of you who are skeptics can define a key, type in CODE EDIT in Setup mode, and then enter MACROSTATUS ALL in Setup mode to view the disabled key definition. You'll notice that the key is still defined but that if you press it, it will not execute.)

All of this can be confusing to new 4100 users. To add to the confusion, when you exit CODE EDIT, your key definitions are not automatically restored. Fortunately, it's easy to re-enable your key definitions. Press the Setup key and type in the command KEYEXPAND YES. (An application can send the equivalent escape sequence command from the host. Refer to the description of the "Enable-Key-Expansion" command in the Programmer's Manual for your terminal.)

You may issue the "Enable-Key-Expansion" command while the terminal is in CODE EDIT so that you can use programmed keys, but keep in mind that you'll lose the cursor positioning capabilities of the VT-100 emulation. A reminder to applications programmers who want to implement programmed keys while the terminal is in CODE EDIT: The Enable-Key-Expansion command is a 4100-style command; therefore, it is recommended that you precede the "Enable-Key-Expansion" command by a "Select-Code-Tek" command, and then issue a "Select-Code Ansi" command so the terminal will understand subsequent ANSI commands.

Here's a summary of what changes when the terminal enters CODE EDIT:

- The terminal interprets subsequent commands as ANSI X3.64 commands.
- As we've discussed, the terminal's ability to expand programmed keys is disabled until the terminal sees an "Enable-Key-Expansion" command.
- The dialog buffer size is set to 24 lines, and the number of visible dialog area lines is set to match the buffer size. Finally, the dialog area is enabled and made visible.
- Origin mode is set to Absolute and Insert/Replace mode is set to Replace.

All of these changes take place automatically when the application issues the CODE EDIT command, or when you issue it in Setup mode. Keep in mind that these settings are not restored to the original settings when you switch from CODE EDIT to some other code; the only thing that changes when you go from CODE EDIT to CODE TEK, for example, is the command parser that the terminal uses to interpret commands.

For more information, refer to the Operator's Manual or Programmer's Manual for your terminal.

For preview copies or everyday output, the 4631, 4632 and 4634 copiers produce high-resolution, monochrome output, with copy times between 9 and 26 seconds.

For Fast, Working Hardcopy, Try the 4630 Series

Color is an integral part of most graphics applications today. Still, users often need to generate quick, working copies of the screen display. Tek's 4630 Series of gray-scale graphics copiers meets this need, providing excellent monochrome output with a minimum terminal tie-up.

The 4630 Series consists of three products. The 4631 Hard Copy Unit supports the 4050 Series Desktop Computers and DVST terminals such as the 4014 and 4114. The 4632 Video Hard Copy Unit and 4634 Imaging Hard Copy Unit can be used with Tek's 4109, 4109A, and 4111 Computer Display Terminals and the 4115B/4120 Series Color Graphics Workstations, as well as a variety of terminals from other vendors. A video interface unit, available from PDS Video Technology, Inc., of San Diego, California, allows the 4104A, 4105A, 4106A and 4107A Computer Display Terminals (as well as the 4105, 06, and 07) to drive the 4632 or 4634. In addition, recently developed interface options allow the 4634 to be used with both 50-Hz and 60-Hz version of the IBM® 5080.

Both the 4632 and 4634 offer at least six distinguishable gray shades, with an option of up to eight shades minimum on the 4632. Optional high-performance paper on the 4634 gives up to 12 minimum gray shades on the 4634. The larger number of gray shades on the 4634 allows a smoother transition from one shade to another and is especially advantageous for applications such as mapping and solids modeling.

Copies are produced quickly. The 4631 and 4632 generate the first copy of a display in 18 seconds and subsequent copies of the same image in 9 seconds. For the 4634, the figures are 26 and 12 seconds, respectively.

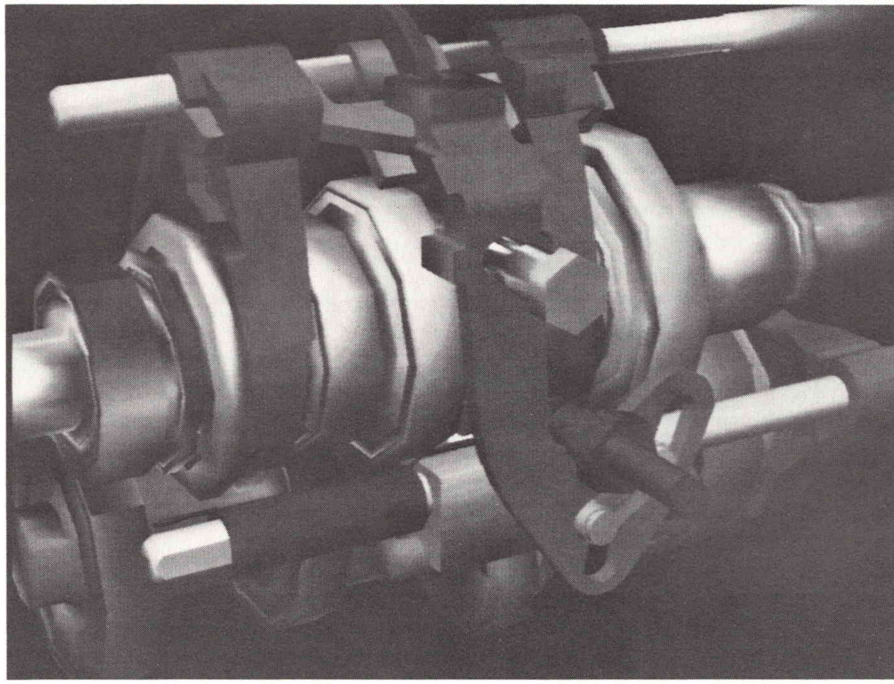


Figure 1. Photographic processes and fiber-optic technology on the 4634 Imaging Hard Copy Unit produce the fine detail and photographic quality of this transmission. Data courtesy of IBM®.

Monochrome Output from Color Displays

The 4632 and 4632 use photographic processes and fiber-optic technology to produce high-resolution, photographic-quality copies. Inside the copiers is a small CRT whose faceplate contains tiny fiber optic tubes. The CRT receives and displays the graphics and alphanumeric information that appears on the raster scan display. Simultaneously, special photosensitive, dry-silver paper passes by the CRT and is exposed to the light from the CRT image. The fiber optic tubes on the face of the CRT keep the light focussed, for extremely accurate photographic reproduction. The exposed image is then developed by heat for a crisp, clear image on the paper—made completely without toners or additives.

The copiers can accept a workstation's composite "video out" signal, or can use an RGB mixer that combines the display's red, green and blue color signals into a single composite video output containing the appropriate shades of gray. Because the copiers use analog interfaces, first-copy times are the same, regardless of the image complexity. This is in contrast to digitally interfaced copiers such as laser printers, electrostatic printers and pen plotters, where the more complex an image is, the longer it takes to output the hardcopy.

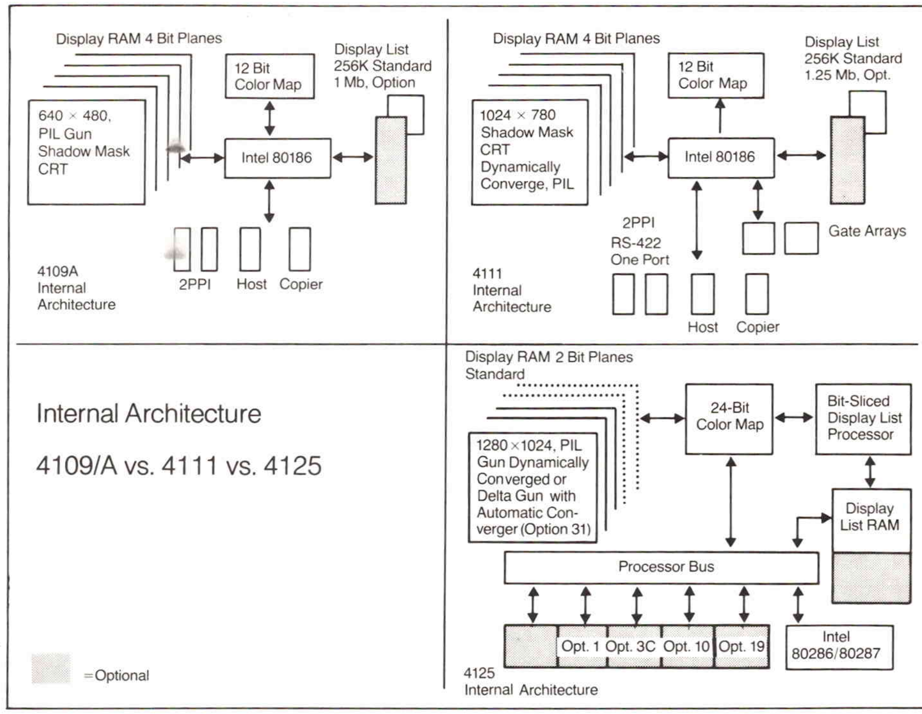
Working Copy

With the 4630 Series monochrome copiers, users can easily obtain fast, high-quality working output—whether preview copies before producing final output, copies of intermediate steps during interactive work sessions, or final output copies for reports and file records. And since the copier connects to the video out port, the peripheral port is still available for connecting a 4690 Series Color Graphics Copier, allowing users to make either quick, gray-scale copies or color copies on demand.

For further information on the 4630 Series copiers, contact your local Tek sales engineer. For information on the PDS monochrome video interface, contact PDS Video Technology, Inc. at 1152 Santa Barbara Street, San Diego, California 92107; (619) 222-7900.

Sizing Up the 4111 Terminal: How Does It Compare?

by Tom Ireland
Terminals Division Marketing
Tektronix, Inc.
Wilsonville, OR



Internal Architecture
4109/A vs. 4111 vs. 4125

The Winter issue of *Tekniques* introduced Tek's newest color graphics terminal, the 4111—a mid-range product whose speed and resolution are made-to-order for low-end CAD applications such as IC design and printed circuit board design. In this article, we'll look at the 4111 more closely, and examine the differences and similarities between the 4111 and its two closest neighbors in the spectrum of Tek terminals, the 4109A and the 4125.

Top of the Line

Tek's Information Display Group currently manufactures two families of terminals designed for engineers and scientists: the low-cost 4100 Series Computer Graphics Terminals and the high-performance 4120 Series Computer Graphics Workstations. The 4111 checks in as the top-of-the-line product in the 4100 Series. It offers a 19" color raster display, along with 4109A-style VT100 compatibility. At the same time, the 4111 borrows some of the firmware microcode of the 4125, and thus the feature sets of the 4111 and the 4125 are fairly similar. Included in the firmware are the new 4125 routines that handle segment editing, segment subroutines, multiple scrolling dialog areas, and pop-up menus.

To put many of the 4125's features into a 4100-Series-priced terminal, the 4111 design team adapted the architecture of the 4109A, building a single-board system based on Intel's 80186 processor but adding custom gate arrays and other custom hardware to boost performance. As a 4100 Series terminal, however, the 4111 does not have the 4125's options of adding 3-D wireframe and shaded surface capabilities to its graphic features. Figure 1 compares the architecture of the three terminals.

Graphics

The graphics area of the 4111 closely resembles that of the 4125. Both systems have either a 12-bit or a 32-bit virtual graphics space, providing up to four billion by four billion addressable points for graphics precision. Local zoom and pan allow users to make full use of this large address space to isolate and display fine details of an image at maximum resolution. Up to 64 views can be created to display selected graphics windows independently, each in its own viewport on the screen.

The 4111's display resolution is midway between that of the 4109A and the 4125, with a screen addressability of 1024×768, compared to the 4109A's 640×480 and the 4125's 1280×1024. This screen addressability, combined with the 4111's 19-inch screen and the ability to display up to 16 colors for graphics, means images are clear and crisp.

Four bit planes are standard on both the 4111 and the 4109A. Each bit plane can be treated as a separate display surface, permitting the user to manipulate layers of data—a feature that is particularly useful with multilayer drawings such as IC design and printed circuit board design.

Other features of the 4111 include the Draw-Rectangle command of the 4125, the ability to change the GIN cursor speed (using the Set-GIN-Rate command for the 4111's thumbwheels and mouse rather than the 4109A's Set-GIN-Cursor-Speed for its joydisk), and routines to handle pop-up menus, saving the graphics under the menu for restoring later. The 4111 also has pixel commands to allow the user to address any pixel on the terminal screen. Since the 4111 screen size differs from both the 4125 and the 4109A, pixel commands written for either of these terminals would need to be recomputed for use on the 4111.

The aspect ratio of the 4111 and the 4109A screen is 4×3, compared to the 4125's 5×4. The 4111 remains compatible with the 4125 in that its default window settings fit the larger graphics window onto its screen. Users who are not running 4125 software may want to change the size of the graphics window to 4095×3071 rather than the larger 4095×3276 of the 4125.

String precision graph text is not scalable on the 4111, as it is on the 4109A.

The 4111 has the most memory of any 4100 Series terminal, with 256 KB of RAM standard to store graphics locally for later use; up to 1 MB can optionally be added for a total of 1.2 MB.

Table 1 compares the graphics capabilities of the three terminals.

Dialog Area

As with the graphics area, the dialog area of the 4111 more closely resembles the 4125 than the 4109A. In both the 4111 and the 4125, the dialog areas (up to 64 areas) can be located and moved anywhere on the screen, and are definable by width as well as by the height definition of the 4109A. The maximum size of the visible dialog area (in characters by lines) is 80×32 for the 4109A, 132×48 for the 4111, and either 80×34 or 160×64 for the 4125.

In the 4111 and the 4125, the same 16-index color map is used for both dialog colors and graphics colors. This contrasts with the 4109A, which maintains a separate 8-index dialog area color map. The user can assign the dialog areas to a specific surface of the graphics bit planes, depending on how the bit planes are partitioned.

All three terminals operate differently in their use of colors in a dialog area. The 4111 uses color pairs (a character color and a background color make up a color pair) to define what the character cells of the dialog area look like; the background is transparent except under characters. In the 4109A, each color is specified separately, and a dialog area background color can be chosen. The 4109A Set-Dialog-Area-Index command defines the colors as in the following example:

```
  ┌Character color
DAI 1 2 3←Background of dialog area
      └Background of character cell
       color
```

The dialog area beyond the cursor changes to reflect the new color combination as soon as the command is entered.

The 4125 Set-Dialog-Area-Index works differently, as shown in the next example:

```
  ┌Character color
DAI 1 2 3←Ignored
      └Background of character cell
       color
```

Parameter 3 of the DAI command is unused because the dialog area background is transparent except under character cells. Dialog area visibility must be toggled for any dialog command to take place; the change in colors doesn't occur automatically, as in the 4109A. When the command does take place, however, it affects the entire dialog area rather than changing only the area past the cursor. The 4125 can have only one pair of colors in the dialog area at a time.

The 4111, on the other hand, can have up to eight color pairs in a dialog area at one time. These color pairs, numbered 1 through 8, are set up by the user with the DAI command.

```
  ┌Character color
DAI 1 2 3←Color pair
      └Background of character cell
       color
```

Parameter 3 assigns the specified number to the color pair that is set up in parameters 1 and 2. Users can define as many of the eight color pairs as desired (toggling the dialog area visibility between each DAI command), then can change to a new color pair with the ANSI Select-Graphics-Rendition (SGR) command. As in the 4109A, the new colors take effect only after the cursor, so that multiple colors can be used in one dialog area. Different color pairs may be defined for each dialog area. The 4111 includes an additional 4125-style command, Set-Alternate-Dialog-Area-Index, which defaults to color pair 2.

New CX4111 Tailored to IBM 3270 Environments

CAD users in IBM 3270 environments who need the high resolution and advanced feature set of the 4111 can now take advantage of the recently announced CX4111, the latest in a series of products developed by Tek for IBM host environments. (For information on another new product for use with IBM hosts, see "Color Output System Provides Superior Graphics Hardcopy for IBM® 3270 Environments" in this issue.)

Like other CX4100 Series terminals and 4120 Series Option 5 graphics workstations, the CX4111 provides easy, direct, coaxial connection to IBM 3274 Cluster Controllers. To provide users with an easy transition to the new terminals, the CX4111 includes the familiar 3270-style keyboard as well as full-screen, color, 3270 alphanumerics.

The CX4111 supports a wide variety of peripherals. The 4690 Series ink-jet copiers provide superior color hardcopy, and can be used with the CX4510A rasterizer for even higher resolution output. The 4634 monochrome copier is available for quick grey-scale output. For graphics input, the 4957 and 4958 tablets are supported, and a three-button mouse is optional.

Table 1. 4109A/4111/4125 Graphics Features

Feature	4109A	4111	4125
Addressable resolution	4096×4096	4096×4096 or 4 billion × 4 billion	4096×4096 or 4 billion × 4 billion
Display addressability	640×480	1024×768	1280×1024
Refresh rate	60 Hz non-interlaced	60 Hz non-interlaced	60 Hz non-interlaced
Displayable graphic colors	16	16	up to 256
Color palette	4096	4096	16 million
Number of graphics views	64	64	64
Alpha text sizes	2	1	2

Table 2. 4109A/4111/4125 Dialog Area Features

Feature	4109A	4111	4125
Maximum number of areas	1	64	64
Maximum size (characters×lines)	80×32	132×48	80×34 or 160×64
Displayable colors	8	8 color pairs	16
Set alpha cursor index	Yes	No	No
Alpha cursor type	Block or underline	Underline	Underline

The thumbwheel can be used to scroll the dialog area on the 4111, but does not move the alpha cursor, as the 4109A joydisk can.

Table 2 summarizes dialog area features.

Interfaces: Ports and Keyboards

Like the 4109A, the 4111 comes standard with two peripheral ports. Port 1 operates as an RS-232 port. Port 0 can be configured as either an RS-232 port or a high-speed RS-422 port, allowing DMA data transfers with a 6130 Intelligent Graphics Workstation. The 4111 has built-in "pseudo-devices" for the purposes of the DMA interface. Four memory areas are reserved for these pseudo-devices, as shown in Table 3. (An article on DMA elsewhere in this issue discusses pseudo-devices.)

Table 3. 4111 Pseudo-devices

Memory Area	Data
PX:	Pixel data
DS:	Unretained segments[1]
SG:	Segments[1]
CM:	Color map data

[1]In display list format

Additional baud rates are available for the 4111 (up to 737K baud) when Port 0 operates as an RS-422 port.

CAD programs typically require a high degree of interactivity. To meet this need, the 4111 uses the keyboard designed for the high-performance 4120 Series graphics workstations. The keyboard includes two thumbwheels (instead of the joydisk of the 4109A), a numeric keypad, and ports for a joystick or an optional mouse. Additional input devices include the Tektronix 4957 and 4958 Graphics Tablets. Multiple graphics input devices can be simultaneously activated for highly interactive system use. In addition, nearly all keyboard keys are programmable, making it exceptionally easy to adapt the keyboard to a particular application or to a user's preferred style of interaction.

The 4111 and the 4125 use slightly different hardcopy commands than the 4109A, and the 4111 does not support a monochrome copier through the hardcopy port. It does, however, support the Tek 4634 Monochrome Graphics Copier through the RGB video-out connectors. (For specific information, contact your local Tektronix field office.)

Table 4 summarizes the interface differences of the three terminals.

VT100 Support

As with other 4100 Series terminals, the 4111 offers a high degree of VT100 compatibility, allowing the user to run a variety of general-purpose alphanumeric programs—spreadsheets, text editors, and so forth. Only three VT100 capabilities available on the 4109A are not implemented in the 4111:

- Full-screen reverse video
- Double-high, double-wide characters
- Media copy

Like the 4109A, the 4111 has code edit and VT52 compatibility, as well as application keypad and cursor keys.

General Terminal Features

Table 5 summarizes some general features of the 4109A, the 4111, and the 4125.

In Summary

The 4111 is a personalized, high-resolution display terminal that offers superior price/performance characteristics for CAD applications such as electronics engineering and mechanical drafting. With the high degree of VT100 compatibility of the 4109A and many of the 4125's graphics features, it offers a cost-effective answer to computer-aided design.

For further information about the graphics capabilities of the Tektronix 4100 Series Graphics Terminals and the 4120 Series Graphics Workstations, refer to the following manuals: *4106/4107/4109/CX Programmers Manual* (070-4893-01), *4110/4120 Series Command Reference with 3D* (070-5141-01), and *4110/4120 Series Host Programmers with 3D* (070-4664-03).

Table 4. 4109A/4111/4125 Interfaces

Feature	4109A	4111	4125
Number of ports	2 RS-232	2 RS-232 or 1 RS-232, 1 RS-422	3 RS-232 optional RS-422 optional
Graphic input devices	Joydisk Graphics tablets	Thumbwheels Mouse Joystick Graphics tablets	Thumbwheels Mouse Joystick Graphics tablets
Hardcopy devices supported			
Color	469X	469X	469X
Monochrome	4632, 4634, 4644	4634	4632, 4634
Video out	Yes	Yes[1]	Yes (optional)

[1]60-Hz, non-interlaced, 48-kHz horizontal frequency. See your Tektronix field office for exact compatibility information.

Table 5. 4109A/4111/4125 General Terminal Features

Feature	4109A	4111	4125
Pseudo-devices	No	Yes	Yes
Nonvolatile parameters	Yes	Yes[1]	No
Block mode communication	No	Yes	Yes (optional)
Page full action	No	Yes	Yes
Mass media (flexible disks, etc.)	No	No	Yes (optional)
User-definable fill patterns[2]	No	Yes	Yes
Dim enable	Yes	No	No

[1]Cannot save keyboard macros.
[2]All have standard fill patterns. The 4111 and 4125 do not have 4109A pre-defined dithered fill patterns.

4128/4129 Offer Flexibility, Interactivity in Viewing Displayed Objects

Local intelligence in Tek's Color Graphics Workstations means users can interactively display different views of an object, without additional host computing or data transmission.

Designed for displaying the graphic output of tasks such as structural analysis, finite element modeling, and thermal and vibrational analysis, the 4128 and 4129 Color Graphics Workstations include a local viewing system—built-in intelligence that gives designers flexibility and responsiveness as they control how an on-screen object is displayed.

The 4128/4129 local viewing system speeds up the display of graphic objects, whether the object is displayed in two dimensions, three-dimensional wireframe or three-dimensional shaded surfaces. Data describing the object is sent to the workstation just once. The local viewing system then allows the designer to interactively perform a variety of 3D viewing transformations, to switch between display forms, or to switch from parallel to perspective projection. The work of redisplaying the object is performed locally—and quickly. There is no delay while the host computes and transmits data each time a new view is shown.

The local viewing system supports a number of 3D viewing transformations. The designer can rotate an object around the horizontal, vertical, or depth axes, can "go around" or "over" an object, viewing it from close in or farther back. The designer can also move inside the object, dropping out layers while viewing the interior, and can easily "reset" to the original view.

The Winter issue of *Tekniques* described how the 4129 stores information about an object. In this article, we'll survey key features of the 4128/4129 local viewing system and show how a designer might use it to view a displayed object as in real life—in parallel or perspective projection, from an overview to the fine details, or from all angles.

How It Works: The Viewing Volume and Simple Shapes

To understand the local viewing system, it's important to keep in mind one crucial idea: the system works as if the designer moves around for different views of an object, while the object itself remains stationary. This idea sometimes appears to contradict what is seen on the screen—the object seems to move, not the designer!—but is central to understanding how viewing transformations are calculated.

Establishing a new view of an object requires axes, points, and distances. The local viewing system uses a "frame of reference" called the *viewing volume* to do this. It is helpful to visualize the viewing volume as a box around the object (see Figure 1) even though the box does not appear on the display screen.

The viewing volume is built around the object as received from the host computer. Any object within the viewing volume can be seen; objects outside the viewing volume cannot. The local viewing system computes 3D viewing transformations on the viewing volume only, not on the object itself.

For example, Figure 1 shows a viewing volume around an object, including a "line of sight." Notice how a simple viewing transformation (PAN along the horizontal axis) moves the viewing volume without changing the object.

Calculating viewing transformations of the viewing volume only, along with local intelligence, allows great speed in computing and drawing new views. It also makes it easy to "reset" or draw the original view.

Although Figure 1 shows the viewing volume as a "shoebox" with a line of sight, the viewing volume also contains vertical, horizontal, and depth axes; reference points; and distances. We'll discuss these other components as we survey the local viewing system functions.

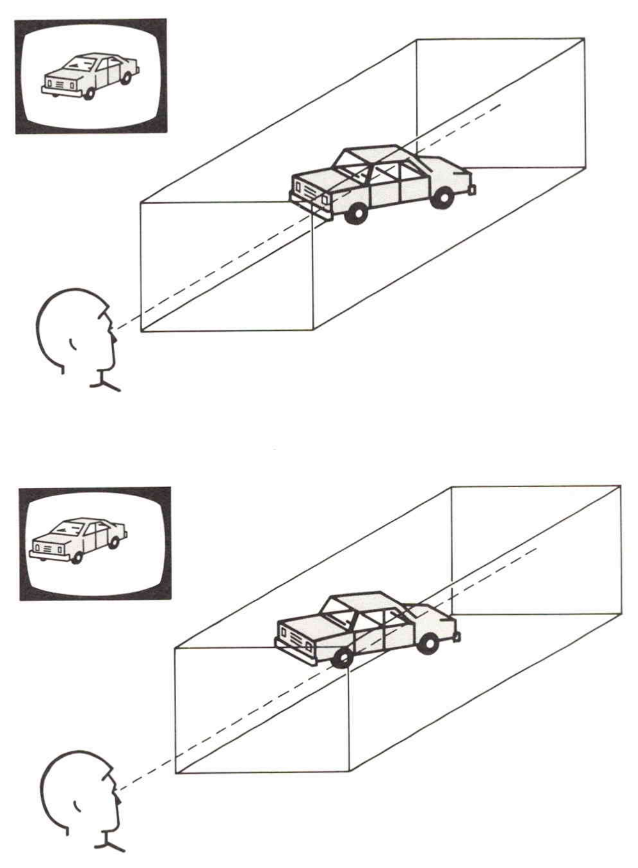


Figure 1.

Parallel and Perspective Projections

The 4128 and 4129 offer both parallel and perspective projections of an object (Figure 2). In a parallel projection, the shape of the viewing volume is a shoebox. A parallel projection shows actual distances and parallel lines, but is not as realistic as perspective projection. Parallel projection can be ambiguous about which lines are in front (although this can be cleared up with shading or hidden-line removal).

In a perspective projection, the shape of the viewing volume is a sideways truncated pyramid, with the small end at the eye position. A perspective projection looks realistic and shows what parts are in front of others, but distorts actual distances and parallel lines. With this projection, the parts of the object closer to the eye look larger, just as in real life, because the near parts of the object occupy a greater proportion of the viewing volume.

In both perspective projection and parallel projection, the object is the same; only the viewing volume changes.

Different Views: The Local Viewing System Functions

The local viewing system becomes active when the user presses the ZOOM or PAN key. Once the viewing system is active, the designer controls it with a combination of the Control key and the numeric keypad (or the special local viewing system function keys) and the two thumbwheels. A dialog area shows the current display form (wireframe, etc.), the projection type, and the coordinates of the components of the viewing volume (Figure 3). Once the designer selects the appropriate viewing transformation and specifies how much movement or rotation is desired, the VIEW key draws the new view. The designer can toggle between projection types by pressing Ctrl-keypad-Minus, then the VIEW key.

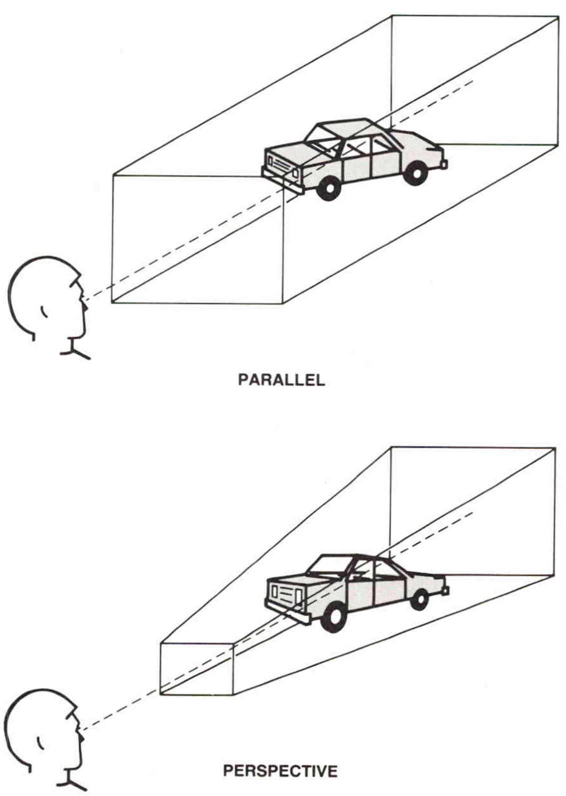


Figure 2.

The functions of the local viewing system allow the designer to view the displayed object as a whole, in fine detail, or from different angles. We'll discuss the viewing transformations through two groups of local viewing system functions. The first group of six functions are easy to use and can accomplish the majority of the viewing transformations required by most designers. These six functions are: ZOOM, PAN, ROTATE, ROLL, RADIUS, and CLIP.

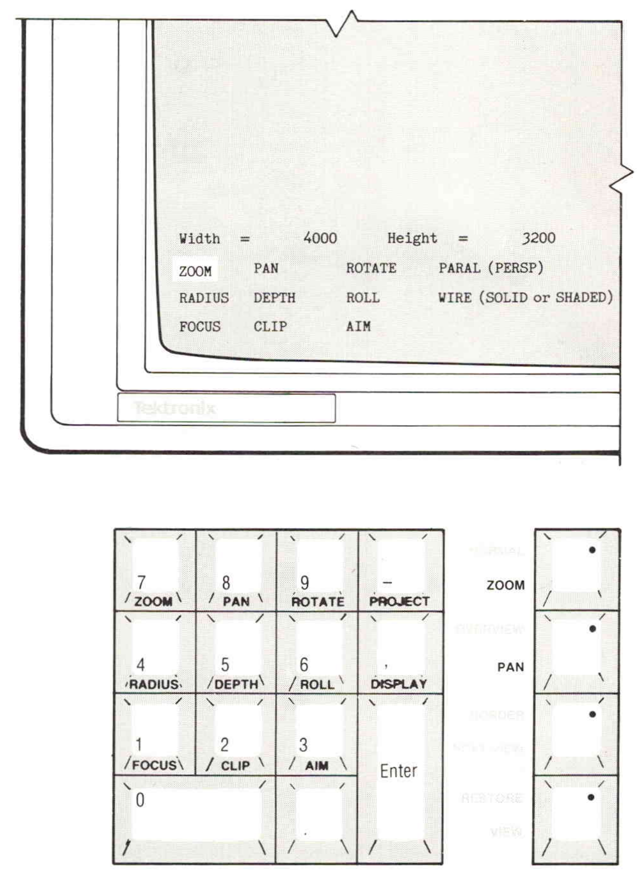


Figure 3.

Zoom

One way of obtaining an overview of an object is to change the size of the image. For example, an architect might want to view a building in architect's scale (1/4 inch to 1 foot) or dollhouse scale (1 inch to 1 foot). Or a designer may be concerned with an image in the center of the display and want it to fill the entire screen. The ZOOM function can accomplish this. Just as in 2D displays, where ZOOM increases or decreases a window, a 3D ZOOM increases or decreases the entire viewing volume. A small volume displays less 3D space, but more detail. So to move from an architect's scale to dollhouse size, the user would ZOOM in, to a smaller viewing volume.

The ZOOM function changes the viewing volume via another component, the *UV window*. This window is defined by the horizontal (U) and vertical (V) axes and the *view reference point*. The view reference point is on the line of sight and defines the *UV plane*, or screen, upon which everything is projected. Because zooming in or out shrinks or expands the UV window, this window controls the horizontal and vertical size of the viewing volume.

When ZOOM is pressed, a rectangle, with crop marks corresponding to the UV window, appears on the display. The thumbwheels are used to shrink or enlarge the rectangle. Figure 4 shows zooming in and its effect on the UV window and the viewing volume.

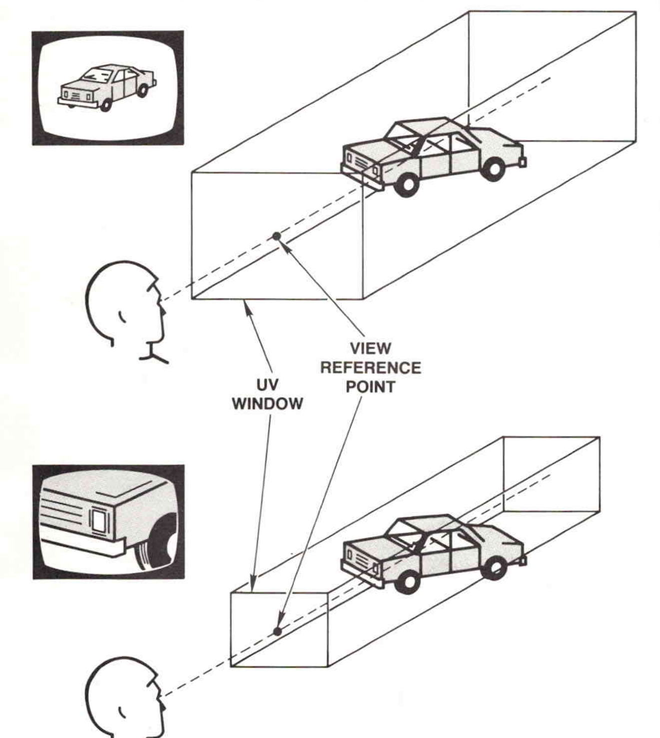


Figure 4.

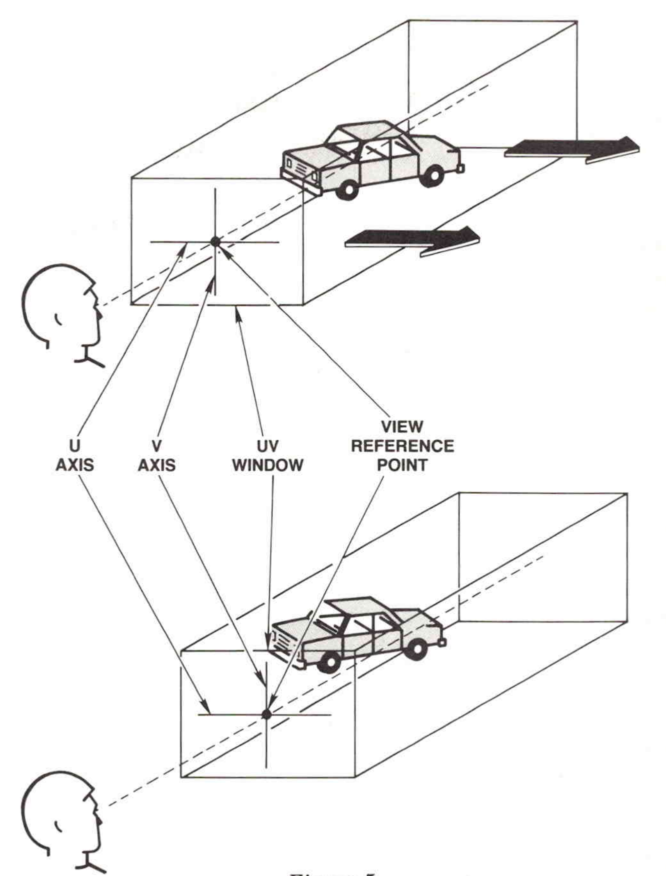


Figure 5.

Pan

Another way to vary the view is to move sideways or up and down with PAN. PAN moves the viewing volume right and left on the U axis (horizontal), or up and down on the V axis (vertical).

When PAN is pressed, a rectangle, with a plus sign corresponding to the UV axes, appears on the display. The thumbwheels move the plus sign, whose center corresponds to the new center of the viewing volume. When VIEW is pressed, a new view reference point is defined, affecting the line of sight, the UV window, and the viewing volume. Figure 5 shows panning and its effect on the UV axes and the viewing volume.

Rotate

The ROTATE function is analogous to "going around" the object to view it from all angles. This function moves the UV window and viewing volume around a fixed point using a fixed radius. This is like moving the the UV window around a sphere, known as the *rotate sphere*. The rotate sphere is tangent to the UV window at the view reference point (like a ball touching a pane of glass). The fixed center point of the rotate sphere is the *view motion center*, usually defined by host software to be near the center of the object.

When ROTATE is pressed, a symbol appears on the screen, corresponding to the view reference point and the radius of the rotate sphere. Moving the bottom thumbwheel causes rotation left or right; the top thumbwheel rotates over and under. Figure 6 shows a rotation and its effect on the image and the viewing volume.

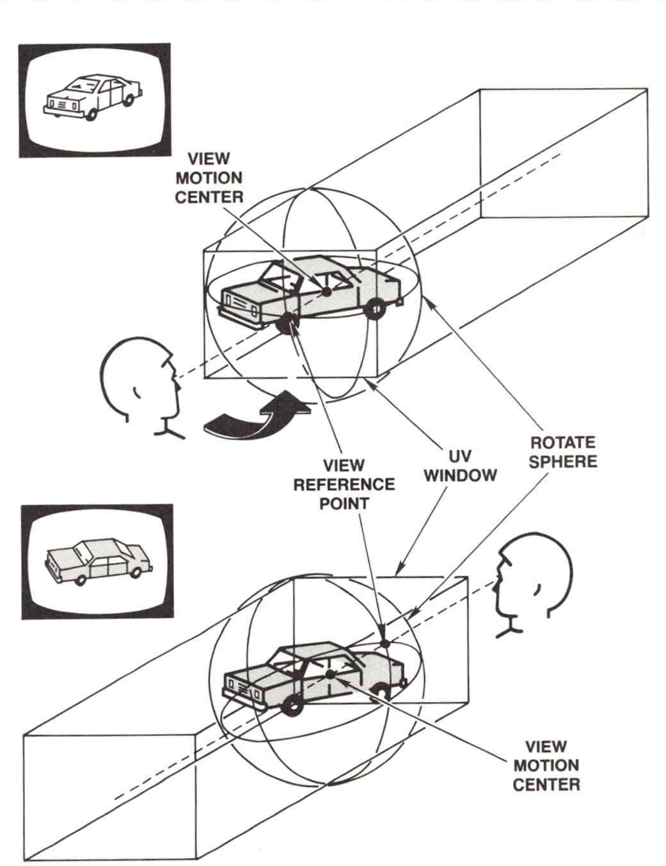


Figure 6.

Roll

The ROLL function (Figure 7) turns the viewing volume around the W axis (line of sight) clockwise or counterclockwise. When ROLL is pressed, a rectangle with a sliding bar appears on the display. Either thumbwheel controls the roll.

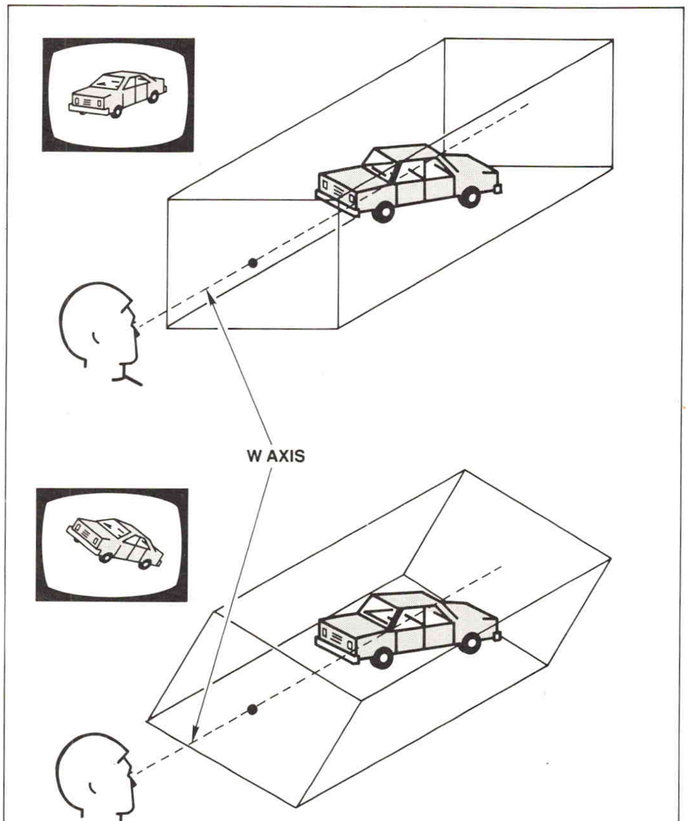


Figure 7.

Radius

The RADIUS function allows the designer to move in closer to or away from the object. RADIUS changes the radius of the rotate sphere; the center remains the same. RADIUS also moves the view reference point (and thus the entire viewing volume) toward or away from the eye along the line of sight. Moving to a smaller radius can show more detail in the image (Figure 8) in a perspective projection, or can be used to extend the viewing volume in a parallel projection.

Clip

The CLIP function allows the designer to view "slices" of the object, if data about the object's interior has been sent from the host. CLIP controls the *front plane* and *back plane*—the ends of the shoebox in a parallel projection, or the apex and base of the pyramid in a perspective projection. A thin slice involves moving the two planes closer together, and vice versa for a thick slice.

When CLIP is pressed, a rectangle with two boxes appears on the display. The outer box represents the back plane, and the inner box represents the front plane. Each box enlarges when the plane approaches the eye and shrinks as the plane moves away from the eye (Figure 9).

The second group of three functions (AIM, FOCUS and DEPTH) are used less often, and the designer can learn them as needed.

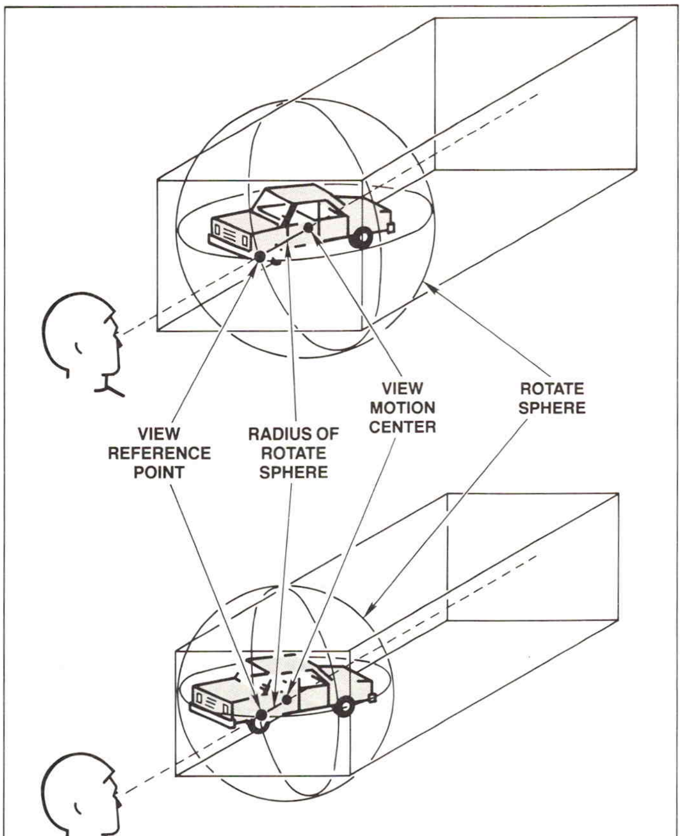


Figure 8.

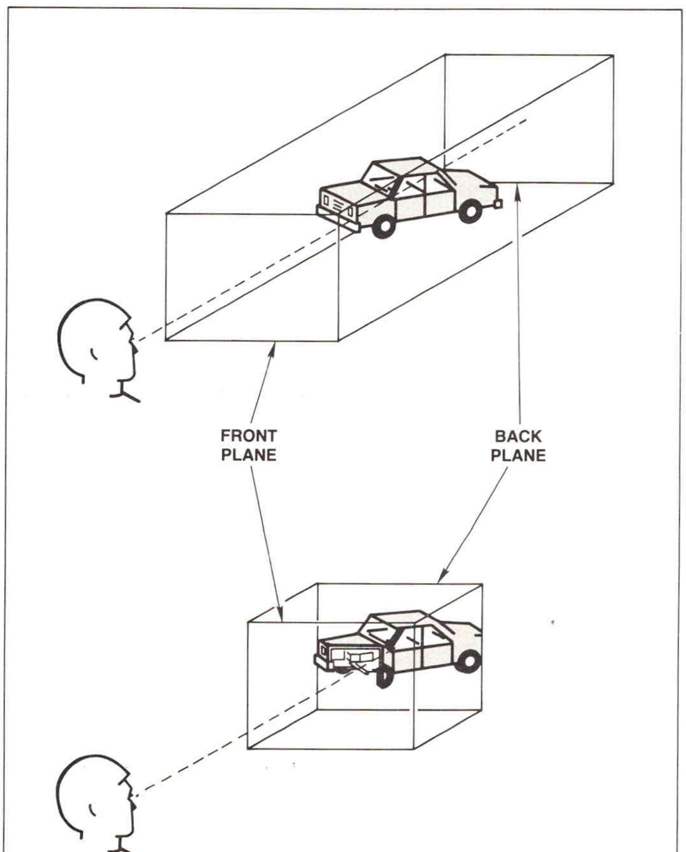


Figure 9.

Aim

The AIM function is analogous to moving your head from side to side. AIM pivots the entire viewing volume left or right, up or down, on a single point, the view reference point. With this viewing transformation, the viewing volume can move away from the object, so that the object is no longer visible. When AIM is pressed, the thumbwheels are used to control the amount of pivoting.

Focus

In real life, you might move away from or toward an object. The FOCUS function imitates this action, as if the eye were moving away from or closer to the displayed object. The name derives from its similarity to a double lens system (as in a camera).

FOCUS moves the eye position relative to the view reference point in the UV window. When FOCUS is pressed, a rectangle with an "F" appears on the display. The "F" gets smaller as the "eye" moves closer to the UV window.

In a parallel projection, the size of the viewing volume remains the same and changing the focus has little effect. With a perspective projection, however, the view changes shape and size, because the pyramid shape of the viewing volume in a perspective projection is based on the eye position. As the eye moves farther and farther back from the view reference point, the lines of projection come closer and closer, finally approaching a parallel projection.

Depth

Another way to change views "close in" or "far away" is with DEPTH. DEPTH moves the view motion center along the line of sight. The view reference point remains the same, so the radius of the rotate sphere changes.

The DEPTH function allows the designer to "fly by" or "fly through" 3D space. This is analogous to tacking with a sailboat: a sailboat tacks by sailing forward, turning a little, sailing forward again, turning again, and so on. The designer can imitate this motion by using DEPTH, then ROTATE, then repeating DEPTH and ROTATE until the desired position is reached.

When the VIEW key is pressed after the DEPTH function, the view motion center changes; but because the view reference point has not moved, the rotate sphere radius also changes.

When DEPTH is pressed, a rectangle with a stylized "D" appears on the screen. As the viewing volume moves away from the object along the line of sight, the "D" gets larger.

Restoring Views

Finally, the 4128 and 4129 offer another function besides the different display forms, projection styles, and viewing transformation functions. The RESTORE function allows the designer to retrieve either the original view of the object as received from the host at the beginning of the work session, or to retrieve any one of the last three views displayed with the VIEW key. This encourages the designer to play "what if" while searching for effective views; changing to a newer view doesn't mean losing a previous good view.

Free to Experiment

In summary, the local viewing system in the 4128 and 4129 encourages designers to experiment. Because of the ease in choosing new display forms, projection types, and viewing transformations; the speed in displaying new views; and the ease with which a previous view can be restored, the 4128 and 4129 give designers flexibility and interactivity in viewing displayed objects.

One-Pass Tiling Engine Boosts 4129 Performance

by Dave Swenson
Graphics Workstations Div. Marketing
Tektronix, Inc.
Wilsonville, OR

The 4129 Color Graphics Workstation has been enhanced with a new tiling engine that allows tiling of 3-D shaded objects in a single pass.

Initially, the 4129 used an optimized four-pass tiling procedure, calculating and displaying shaded tiles of 3-D polygons in four distinct passes, with perceptible pauses during each intermediate calculation. New 4129s and 4100F59 upgrade kits calculate the entire image in one pass, then display it in one smooth operation. Factory testing with ME/CAD images from our Solution Vendor Program shows a performance increase ranging from 50 to 100 percent.

Changes to the tiling memory board set include memory chips with four times as much memory as the initial boards, and subsystem self-test changes that will test this extended memory automatically without any change in the main firmware of the 4129.

All 4129 color workstations and 4100F59 field upgrade kits shipped since February 7, 1986 have this new one-pass tiling engine.

High-Speed Interfaces for the 4120 Series: Understanding Options 3A, 3C, 3E and 3Q

by Tim Brown
Product Support Analyst
Tektronix, Inc.
Philadelphia, PA

High-speed interfaces, when used in conjunction with the RS-232-C interface, offer certain applications a dramatic increase in performance over the exclusive use of RS-232-C. This article describes the high-speed options available for Tek's 4120 Series Graphics Workstations, provides an overview of their capabilities, and looks at the type of performance you can expect from them. In addition, this article attempts to clear up issues that are often misunderstood regarding the high-speed interfaces. If you are evaluating the high-speed interfaces for integration into an application, this article will help you understand the interfaces and the type of performance that can be expected from them.

The High-Speed Interface Options

Tek offers four high-speed interfaces for the 4120 Series:

- Option 3A is a parallel, high-speed DMA (Direct Memory Access) interface to the DEC® PDP-11™ and VAX™ series computers. This option includes both the hardware for the 4120 workstation and the hardware for the host computer (VAX). The interface cable is purchased from Tektronix as a separate item because of the varying lengths of cable offered. This option supports high-speed transfers over a maximum distance of 1000 feet.
- Option 3C is a high-speed serial RS-422 interface used to connect the 4120 workstation with the Tektronix 6130 Intelligent Graphics Workstation. The Option 3C interface is standard on the 4111 Computer Graphics Terminal.
- Option 3E is identical to Option 3A except that it does not provide the hardware for the VAX. This option is designed for system integrators who wish to build their own interface to a computer.
- Option 3Q, the newest offering, allows the 4120 to be connected via a parallel DMA interface to the DEC MicroVAX™ II. Unlike the other options, Option 3Q uses TTL signal levels instead of RS-422. This option provides only the hardware for the 4120; DEC supplies the hardware for the MicroVAX II, the three-meter interface cable and an I/O software driver. (Option 3Q can also be purchased from DEC.)

DMA and RS-422: What Are They?

Let's clarify two terms that are often misunderstood: RS-422 and DMA. To understand RS-422, let's compare it with the more common RS-232-C. RS-232-C is an ANSI standard for a serial computer interface which defines signal voltage levels, the connectors, the pin configuration of the connectors, and the signal definitions. RS-232-C does not define data formats (i.e., 7-bit versus 8-bit) or protocols (i.e., ASCII versus EBCDIC). RS-422 is also an ANSI standard, but it defines only the signal voltage levels and states for a single signal line. RS-422 does not define the cable, the type of connector, the number of pins, the data formats or protocol. Nor does RS-422 identify the interface as parallel or serial. This means that, unlike RS-232-C, RS-422 leaves room for a variety of interpretations by hardware manufacturers. For example, Options 3A and 3C both use RS-422 signal levels as prescribed by the ANSI standard, yet both differ in implementation. Option 3A is a parallel interface using three 40-wire ribbon cables sandwiched together. In contrast, Option 3C is a serial interface with 14 wires and different connectors at each end of the cable.

DMA is another term that is often equated with high-speed parallel processes. However, the terms DMA and high-speed parallel are not synonymous. Direct Memory Access (DMA) is the process of writing directly in a computer's memory without the assistance of the controlling processor. The use of the term DMA for Options 3A and 3Q refers to the VAX's perspective on the interface. The hardware that plugs into the VAX's bus is a DMA device. This hardware also controls the passing of data back and forth over the parallel interface.

It is best to think of Options 3A, 3C, 3E and 3Q as "high-speed" interfaces to the 4120 rather than as "DMA" or "RS-422" options. As we will see later, the type of data transferred to the 4120 is generally instructions (op-codes) directed to a special processor and not DMA writes to the 4120's memory. In this article, we'll use the term "high-speed interface" generically for all of the optional interfaces (Options 3A, 3C, 3E, and 3Q).

Performance Implications

There is a common misconception that increasing the speed of communications via a high-speed interface automatically increases the performance of an application. This is true only if the performance bottleneck is caused by the data transfer rate to the display device. If you are considering implementing a high-speed interface, the first question to ask is, "Where is the bottleneck?" If you increase the communications speed to the 4120 via a high-speed interface, the next question is, "Where will the bottleneck be now?"

The 4120's central processor (an Intel 80286) is able to keep up with alphanumeric data and "simple" graphics (i.e., vectors) at a data communications rate of 38.4K baud (3,840 8-bit CPS-characters per second). This means that the 4120 can process 2D vector data as fast as 38.4K baud. If the graphics are more complex (i.e., panels, segments, 3D graphics) then the 4120 will process it at a slower rate. The user may not notice this since the amount of graphical data may be relatively small (as is the case during most interactive graphics applications). Performance becomes an issue when larger amounts of graphical data are being sent to the 4120 and the user has to wait for an unacceptable period of time while the graphics draw on the screen. Rather than implementing a high-speed interface to increase performance, you might be able to increase your application's performance sufficiently by using the local intelligence of the 4120. For example, you can gain performance by manipulating the graphical data locally rather than repeatedly recreating it from the host. If the volume of the graphical data is very large, however, RS-232-C even at 38.4K baud can result in slow graphics performance, and the 4120's high-speed interface may present a solution.

To benefit from the high-speed interface, the host application must assume some of the work that the 4120's processor does for RS-232-C graphical data. If the host application sends the low-level op-code instructions, the application can see a dramatic increase in speed. The tradeoff, however, is the amount of programming resources and expertise required to modify the application to send the low-level op-codes, in addition to the resulting increase in CPU workload.

4120 Series Architecture

To understand what is required from a programming standpoint, let's first look at the architecture of the 4120s. Figure 1 shows a block diagram of the 4120. The central processor serves several functions, including that of "command interpreter" for incoming RS-232-C commands and data. It controls the internal bus, writes to the terminal's segment memory, builds the low-level instructions out of RS-232-C commands (which ultimately cause the graphics to be displayed), handles all of the RS-232-C and keyboard I/O, and packages reports and messages going to the host and to the display. The Picture Processor is a bit-sliced processor that is responsible for displaying the graphics. Its instruction set (display list op-codes) is received from the command interpreter or read from segment memory, and it in turn writes to the 4120's frame buffer. Segment memory is an area of random access memory (RAM) where low-level graphics instructions are stored for use by the 4120. The frame buffer is the memory (the bit planes) containing the pixel data used to refresh the 4120's screen. The Alpha Overlay (also shown in Figure 1) is a processor that controls the dialog areas and the ANSI X3.64 instructions. The timing controller coordinates the reading of the frame buffer and the Alpha Overlay. The digital information in the frame buffer is combined with the color map and converted to red, green, and blue video signals at the 60Hz refresh rate. (For a more detailed discussion of the 4120 Series architecture, see "The Architecture of a High Performance Color Graphics Workstation" in the Fall, 1985 issue of *Teknīques*.)

When the 4120 receives graphical data over the RS-232-C line, for example, the command interpreter checks incoming data for any command character (i.e., Escape, GS, etc.) that precedes a command. The command interpreter then checks the syntax of the command, creates the appropriate optimized Picture Processor instruction(s) (the op-codes), and passes the instruction(s) on to the Picture Processor. If the data is to be a retained segment, the command interpreter also places these instructions in RAM.

Not for Every Application

To see an increase in graphics throughput, a high-speed interface must eliminate some of the work that the command interpreter normally does. Ideally, an application that uses the high-speed interface mimics the process of the command interpreter; the application does the syntax checking, and creates the appropriate Picture Processor op-code for the best throughput. This is similar to writing a program in a low-level language such as Assembler, versus writing the same program in a high-level language like FORTRAN. In directing instructions to one of the low-level logical "devices," the application that uses the high-speed interface is responsible for ensuring that the instructions are valid, and equally important, that they are optimized. If the data is not properly optimized, it is very possible that the performance will be slower than using the normal RS-232-C interface. If the application does an intelligent job of building the command stream from the low-level instruction set, however, the improved performance can be quite attractive. For example, sending the image shown in Figure 2 via RS-232-C at 38.4K baud takes 7.9 seconds to complete. The same display sent via the Option 3Q interface to a MicroVAX II in display list op-code format displays the graphical results in 750 msec!

In addition to the extra work required by the application to communicate to the 4120 via the op-code instruction set, there is also a good deal of overhead in actually using the high-speed interface. This overhead makes it in general undesirable to transmit small amounts of data (1K bytes or less).[1] The transfer of large amounts of data will be very effective. The optimum amounts are in excess of 8K bytes for a single buffer of op-code instructions. This means that the most interactive applications are best suited for the standard RS-232-C interface. Applications that send high volumes of graphical data with little or no interactivity will benefit from using a high-speed interface. The use of the high-speed interface does not exclude the use of the standard RS-232-C interface. In fact, currently the RS-232-C interface is required to initiate and direct the transfer of graphical data over the high-speed interface. This means that an application will use both interfaces and can use the appropriate interface for the amount and type of data being handled. For example, it might be appropriate to download initial graphics via the high-speed interface, interactively modify it using the RS-232-C interface, and then save the results back to the host via the high-speed interface.

Editor's Note

An interface, according to computer science dictionaries, is a common boundary between two systems or devices. The verb "to interface" refers to the process of interconnecting devices or systems that have different characteristics.

As desk-top systems proliferate and application requirements grow more complex, methods of interfacing—of moving data between systems and devices—become increasingly important. In this issue, *Tekniques* looks at three types of interfaces used in some of Tek's graphics products. Tim Brown, of Tek's Philadelphia field office, explores the high-speed interfaces (often referred to as DMA interfaces) available for moving graphics data between Tek's 4111 and 4120 Series workstations and a host such as a 6130 intelligent workstation or a DEC® VAX™. As an example of software interfacing, Chris Rotvig, product line manager for PLOT 10® TekniCAD, describes CADDport, a TekniCAD option that provides an interface between Tek's drafting software and a number of CAD packages. Finally, Al Paylor, of the Phoenix field office, and Clarence Peckham, engineering manager of the 6130 LAN team, collaborated to produce the article on local area networks—an essential component of a successful workstation and a standard feature of the 6130.

Sources and Destinations

The high-speed interfaces support bidirectional communications, allowing the host application to obtain a "snapshot" of the graphics in the 4120. In the remainder of the article, we'll look at the destination and source devices that can be used by an application over the high-speed interface.

There are four different low-level destination logical (pseudo) "devices" to which data can be directed over the high-speed interface. In addition, data can be sent over the high-speed interface to the command interpreter. Which device you should use depends on the type of graphics required and what is to be done with the graphics once they are sent to the 4120.

The destination devices are *DM:*, *SG:*, *DS:*, *PX:* and *CM:*.[2] Each destination requires a unique data format and offers different rates of data throughput.

The Simple Method

The easiest way to access the high-speed interface is to send a command stream of RS-232-C style graphics commands to the command interpreter over the high-speed interface. The advantage of this method is that little programming change is required; in addition, these changes can be done as a post-processing function for non-interactive graphics applications.

This method requires, however, that the command interpreter process the data in the same manner as standard RS-232-C communications. Therefore, the performance you can expect will be that of the maximum throughput of the command interpreter. In general, the throughput rate of graphics will not exceed the maximum RS-232-C throughput rate of 38.4K baud. Although this allows communications at higher baud rates than what is typically supported in most host installations, significantly faster throughput performance can be achieved when the host offloads some of the work that the command interpreter traditionally performs.

Achieving higher throughput requires routing a special data stream to a specific low-level pseudo device. The following describes those devices, the type of graphical data required and the expected performance.

Non-Retained Graphical Data

DS: is a "pseudo" device; that is, it does not correspond with an actual physical device. *DS:* indicates that the data is to be displayed by the 4120 and not retained in segment display-list memory. This device offers high-order graphics primitives (i.e., move, draw, rectangle, panel, text, etc.). These primitives will not be stored as segments; they are simply processed into screen graphics. (See Appendix F of the *4110/4120 Command Reference Manual* for a complete list of the primitives.) The commands are passed as binary (8-bit) instructions. The throughput that can be achieved for the "average" graphics is approximately 70K bytes/second. (By "average" graphics, I mean that it contains various op-code instructions typical of a "real-life" application, as opposed to just one type of instruction,

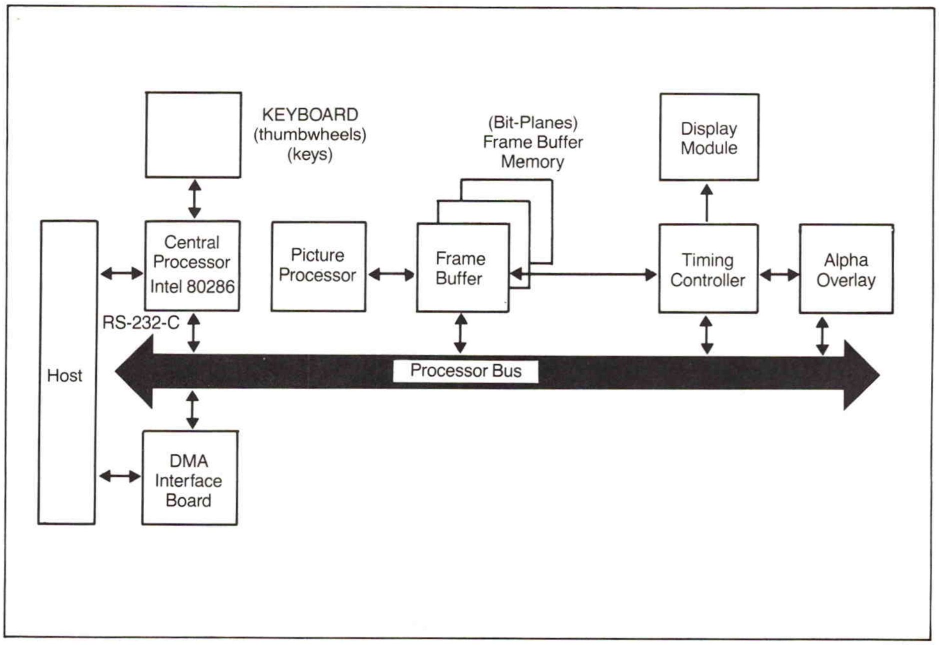


Figure 1. 4120 System Block Diagram.

which would give a "best-case" scenario. In addition to using "average" graphics, all tests for this article are based on 2D data; 3D data would give different results.)

Segment Data

The *SG:* pseudo device is used for the retained segment display list. This device is similar to the *DS:* device. Data directed to *SG:* is written to the 4120's RAM and then displayed. The display list of Picture Processor op-codes is identical to those used by *DS:*; however, additional information is used to define the segment properly. Once the segment is created, it can be manipulated in the same manner as a segment created via RS-232-C. The throughput performance is less than that of data going to *DS:* since there is an additional operation of storing the data in the 4120's RAM. The advantages of segment data versus non-segment data more than compensate for the differences, though. The performance of "average" graphics to the *SG:* device is around 40K bytes/second.

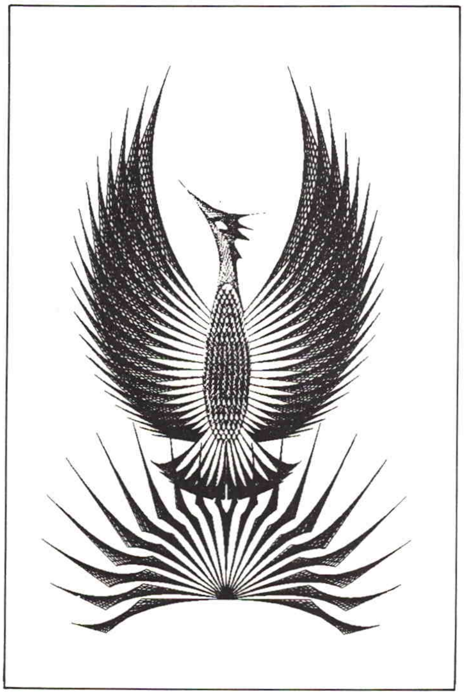


Figure 2.

Pixel Data

The pixel pseudo device, *PX:*, directs pixel data to the lowest level device. *PX:* data, comprised of binary pixel values, offers the highest data throughput, approximately 180K bytes/second. This equates to writing an entire screen of pixels in about 7 seconds. Since each pixel is simply a bit pattern stored in the 4120's frame buffer, this operation requires the most work from a host-based application and the least work from the 4120 Graphics Workstation.

Color Map Data

CM: is the destination device for data going to the color map pesudo device. Since the amount of data is relatively small (768 bytes defines all 256 colors) and no additional work is required once the color map is received, the effective throughput is instantaneous.

Data Going to the Host

Data can be copied from the 4120 to the host in various forms. *CM:* copies the binary color map information back to the host. *PX:* copies back a pixel rendition of the visible screen. *SG:* copies back the binary display list for all retained segments corresponding to the "current" dimensionality of the 4120. (For example, if the 4120 is in 2D mode it copies only 2D segment display list information. To copy 3D data, the 4120 must be in 3D coordinate mode.) It is not possible to get data from the *DS:* pseudo device because this device is used for non-retained graphics.

When *DM:* is used as a source for the copy, it indicates that the transfer will occur via the high-speed interface going to the stated destination device.

Numbers Don't Tell the Whole Story

The Option 3A and 3Q data sheets state that transfer rates of 1 MB/second are possible; Option 3C quotes a 1 Mbit/second transfer rate. Why do these numbers differ from the average graphics throughput figures given in this article?

The figures in the data sheets are based on the speed of transferring the data across the interface cable and are not measures of the possible throughput. This means that the speed of the transfer is faster than the data can be handled at each end of the interface cable. Thus the bottleneck can be the processing of the data, rather than the rate of the data transfer. The difference between the 1 MByte/second transfer rate on Options 3A and 3Q versus the 1 Mbit/second transfer rate for Option 3C is due to the fact that 3A and 3Q are parallel interfaces (passing a byte at a time) and Option 3C is a serial interface (passing data bit by bit).[3]

The throughput of the specified "devices" (*SG:*, *DS:*, etc.) is given in "bytes/second" figures. What does this mean to the user of the 4120? As a point of reference, the 4120 can redraw 50,000 short vectors per second. This is the maximum rate at which the Picture Processor can display 2D data. It is safe to say that the high-speed interface will not exceed this internal redraw rate. How close can you get? That depends on what type of data is being sent (i.e., which pseudo device is the destination). The numbers given above quote the data going to *PX:* as the fastest. With a throughput rate of 180K bytes/second, you should get the best performance, but this is not necessarily the case. It takes one byte to define a single pixel. The 4120s have a pixel resolution of 1280 by 1024, which means that there are 1.3 million pixels. To write to all pixel locations will take about 7 seconds. Additionally, the host is now burdened with building a pixel image. This may be desirable for certain types of graphics applications such as "imaging" or highly complicated solids modeling applications (using techniques such as ray tracing). However, if the application is a vector-oriented application, defining pixels may be a burden to the host; it also does not give the user the throughput or the advantage of working with the 4120's local intelligence.

An example of drawing a vector diagonally across the screen will demonstrate this issue. Let's assume that the vector is 1000 pixels in length. At one byte/pixel, the data going to the *PX:* device would be 1000 bytes long just to define the illuminated pixels. Additional pixel data is required to "move" the pixel beam to create the diagonal vector. Alternatively, data going to the *DS:* device would consist of two commands to draw the same vector (AMOVE16 and DRAW16). Both commands require five bytes, giving a total of ten bytes for the vector. The reduction in data more than compensates for the difference in speed that the *PX:* device has over the *DS:* device. However, the application may want to direct the data to *SG:*, thereby gaining a retained segment that the 4120 can then manipulate locally during the graphics session.

How Fast is Fast?

What does this mean in vector performance? Again, the answer depends on the data being sent. Vector data is sent to either the *DS:* device or to the *SG:* device. In examining the list of display list op-codes, there are several varieties of move and draw commands that allow the application to choose between absolute or relative addressing, and to choose the most efficient command for the size of the move or draw. The best-case performance will be the shortest move or draw. The DRAW7 op-code, for example, takes two bytes to encode and allows seven bits to define the relative distance to be traversed. This means that the distance must not exceed 64 addressable points (2 to the 7th power=64, which is about 0.6 of an inch on the 4120's screen). Sending this data to the 4120 exclusively would result in the best-case performance. The drawing rate of DRAW7s to the *DS:* pseudo device is approximately 35K vectors/second—about 70% of the internal Picture Processor drawing rate of the 4120. Directing DRAW7 data to *SG:* as a single segment gives a drawing rate of 25K vectors/second, or 50% of the 4120's redraw rate. These figures are approximations, and will differ from actual application performance based on how the application uses various length move and draw op-codes, as well as other instructions.

The draw rate for any given application is affected by the amount of time the application takes to compute the optimized instructions to the 4120. It is possible to achieve throughput rates slower than the standard RS-232-C interface by not efficiently building the op-code instruction set. Performance will also vary based on the volume of data being sent, block size of data transfers, and host computer usage. These issues will be the subject of a future article.

In Summary

Using the high-speed interface effectively can provide a dramatic increase in 4120 performance for certain applications. Just how much improvement an application will experience depends on the variables we've discussed here. If you are a user of an application that supports one of the high-speed interfaces, all the decisions have been made for you. You will be able to reap the benefits of a tremendous increase in the graphics throughput. If you are interested in integrating one of the high-speed interfaces into an application, you can find additional information in the 4110/4120 Series Command Reference and Host Programmers manuals, as well as in the installation and service documentation for the various high-speed options.

[1] An exception to this is data going to *CM:*, the color map pseudo-device, which is by definition small.

[2] *DM:* is a physical device that refers to the high-speed interface itself. When it is used as the destination, it indicates that the transfer will go to the host via the high-speed interface. See "Data Going To The Host."

[3] The throughput figures quoted for the various pseudo devices are based on random data types running from a MicroVAX II to a 4125 via Option 3Q. You can expect different throughput rates as these factors change. An additional assumption is that the host has already computed the graphical data. These figures are the rate that an op-code display list on a VAX can be transferred via Option 3Q to the 4120. In other words, this is the rate at which the 4120 can process data to a specified pseudo device. The total volume of data as well as the size of the blocks transferred via the high-speed interface will also affect performance.

Software interface option allows data exchange between PLOT 10 TekniCAD and a variety of CAD design packages and adds flexibility in use of computing resources. Both IDES-based and custom interfaces are available.

CADDport—A Smooth, Powerful Pathway to the World of CAD

by Christopher Rotvik
Graphics Software Products Marketing
Tektronix, Inc.
Wilsonville, OR

A major aerospace firm purchases a powerful CAD system with two workstations at a cost of over $200,000. Soon, the system is fully saturated, and the firm's engineering management begins making plans to allocate funds to purchase a second system of the same type. However, before making the purchase, they attend a Tektronix seminar on PLOT 10® TekniCAD. After analyzing the workload on the CAD system, it becomes apparent that a better use of their resources would keep all the designers on the presently owned CAD system without having to purchase a second one.

The solution: offload the drafting tasks to a TekniCAD system with CADDport for about 1/3 the cost of a new CAD system.

In another instance, a small subcontractor to one of the Big-3 American automobile manufacturers in Detroit needs to be able to transfer blueprint data electronically to its client. This subcontractor is a long way from fully automating its engineering, but it wants to start at the most familiar function—computer-aided drafting.

The solution: TekniCAD with CADDport, a cost-effective entry point into engineering automation. CADDport lets the company transfer data electronically to the client while giving them the flexibility to grow into a larger CAD system as their needs dictate.

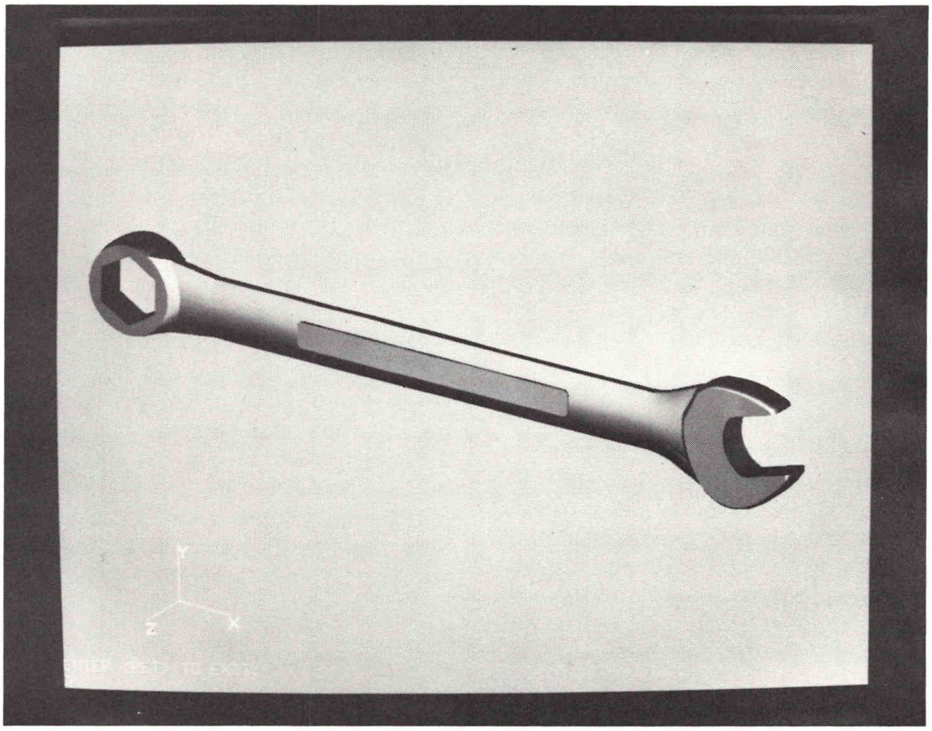


Figure 1. Sample of CAD prototype. (Data courtesy of PDA Engineering.)

These examples illustrate some of the ways in which TekniCAD and its companion product CADDport are providing effective solutions to CAD data translation problems. By coupling computer-aided drafting with computer-aided design, CADDport offers a smooth pathway to the world of CAD, both at entry-level and through making better use of existing CAD resources.

In this article, we'll examine CADDport—what it is, how it relates to TekniCAD, and how the two work together to help users realize the full potential of today's computer-aided engineering systems.

A Link Between Design and TekniCAD

Computer-aided design systems provide design engineers with a sophisticated set of tools used to interactively model and analyze a computerized prototype of a product (Figure 1). Once complete, the design must then be documented in order to be efficiently carried to production.

TekniCAD provides that important design documentation, and CADDport is the software interface through which the design is transferred from the CAD system to TekniCAD. CADDport translates the 3-D design information from the CAD system into the TekniCAD database format. Unlike most CAD systems, TekniCAD produces the design documentation without the overhead inherent in 3-D, that is, in a more cost-effective manner.

CADDport Supports IGES Interfacing Standard

CADDport allows the user to choose between two interface methods: the industry-standard interface, and custom interfaces tailored to a specific CAD system.

CADDport supports Initial Graphics Exchange Specification (IGES) format files, the most universally supported means of translating graphical design information between dissimilar systems. Figure 2 is a block diagram of an IGES-based CADDport system. In such systems, the CAD system's postprocessor first transforms the 3-dimensional output files into an intermediate, IGES-formatted file. Then CADDport takes over and translates the IGES-formatted files into TekniCAD's internal, 2-dimensional file structure.

Today, many CAD software packages support the IGES standard, including CADAM™ (Cadam, Inc.), SDRC GEOMOD™ (GE CAE, International), and ANVIL-4000® (Manufacturing and Consulting Services, Inc.). In addition, turnkey CAD systems vendors such as Computervision are further supporting the IGES standard.

The IGES standard began in late 1979 as an effort to create a graphics exchange specification to solve the problem of exchanging data between dissimilar systems. At that time, it was officially a project of the Air Force Integrated Computer-Aided Manufacturing (ICAM) program and drew heavily on previous efforts by both Boeing Corporation and General Electric.

The first version of the IGES specification was published in January, 1980. Later that year, the American National Standards Institute (ANSI) recommended IGES to be part of a proposed American National Standard. Today, version 2.0 of the IGES standard has been adopted, and Version 3.0 is under development.

By design, IGES is a flexible and expandable format which accommodates most of the data constructs native to most CAD systems. While this flexibility is needed to allow a wide range of vendors to translate their files using IGES, it sometimes complicates the processing of IGES files. A recent study from Sandia National Laboratories cites four areas that can cause problems with IGES exchanges: incomplete processor implementations, system differences, poor mapping choices, and user conventions.[1]

Still, IGES provides a good first step toward making different CAD systems compatible with one another. It can at least make 70–90% of the data from one system available outside the system without the user's having to access the usually proprietary native database.

Tektronix's implementation of IGES interfaces has been recognized as being one of the best in the industry. It overcomes the four problem areas cited in the Sandia study by tailoring its IGES translator to the CAD packages of leading targeted vendors. In many ways, the IGES-formatted interfaces from Tektronix offer the benefits of a system-specific interface within the confines of the IGES standard.

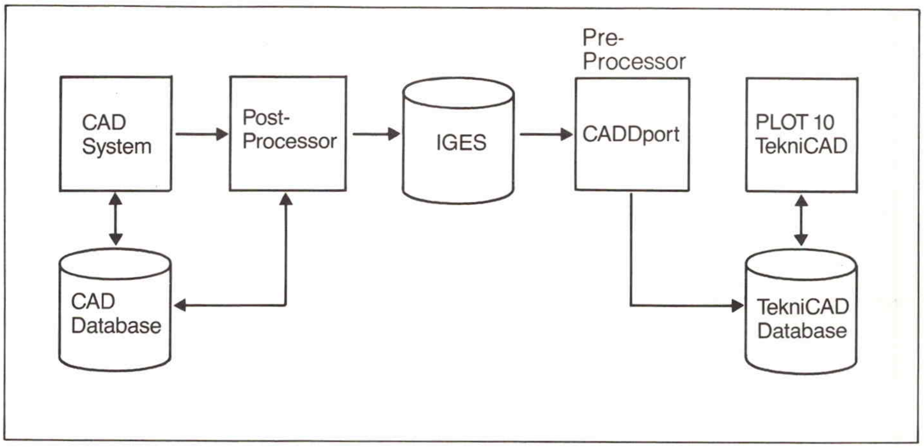


Figure 2. CAD to TekniCAD via IGES. A number of CAD packages support the IGES standard, including CADAM, GEOMOD and ANVIL-4000. The data flow is bidirectional, allowing TekniCAD information to be shipped to the design package.

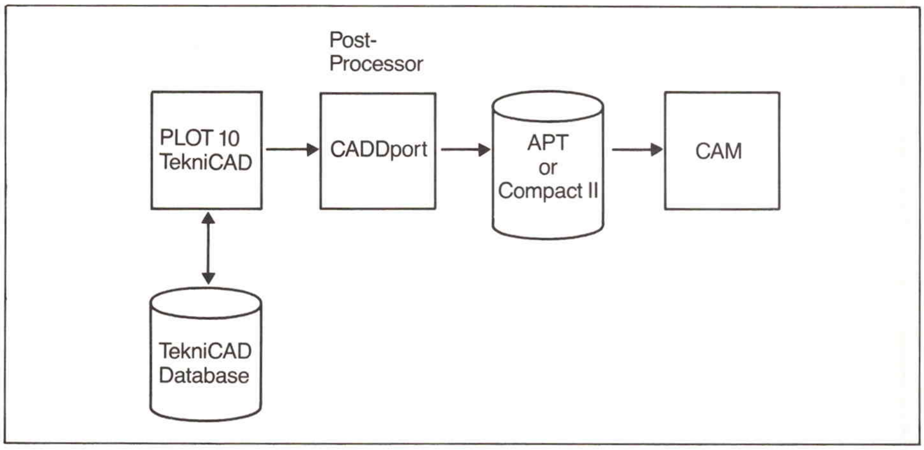


Figure 3. For output to numerical control (NC) machines, CADDport functions as a postprocessor, translating drafting data into APT CL-file or COMPACT II format.

System-Specific Interfaces, Too

Tektronix also offers several system-specific interfaces that translate directly from the design system database to the TekniCAD database.

The CADDport interface to PATRAN® (PDA Engineering, Santa Ana, CA) provides an example of a system-specific interface. PATRAN is a versatile software program that is used for solid geometry modeling, analysis modeling, and results evaluation for finite element analysis. Since PATRAN lacks drafting capability, TekniCAD provides the drafting function. CADDport's conversion program converts PATRAN files directly to TekniCAD files.

CADDport also supports a number of other system-specific interfaces including those from Intergraph™, Inc. The Intergraph interface is interesting because it is a system-specific interface that uses a data file much like IGES. Intergraph chose to call its data file the Standard Interchange Format (SIF). Since Intergraph has its own internal format, it must go through a translator to create a SIF file that is then read into TekniCAD via CADDport.

Within the realm of computer-aided manufacturing, CADDport provides output for numerical control equipment. In this application, CADDport takes the numerical control output from TekniCAD geometry and translates it into either of the two most commonly used parts programming languages—APT CL-file or COMPACT II (Figure 3).

Bidirectional Data Flow

Although the primary emphasis of CADDport is to translate design data from a

CAD system to TekniCAD for purposes of documentation, there are some cases where the user needs to pull data from TekniCAD and bring it back into the CAD system. This might occur if a user is migrating from an old system to a new system. Or perhaps a 2-D data file in the TekniCAD system needs to be pulled up into the CAD system for 3-dimensional modeling. With the bidirectional feature of CADDport, these kinds of data transfers are easily accomplished.

A Better Use of CAD System Resources

As in our first example, the aerospace firm, companies the world over are finding out the benefits of coupling TekniCAD with their CAD systems. By offloading the drafting functions to a dedicated drafting tool like TekniCAD, the costly resources of the CAD system are reserved for the complex design tasks for which they were intended.

In addition, since TekniCAD produces design documentation without the overhead of a 3-dimensional database, it contributes to an overall increase in CAD system throughput.

TekniCAD also makes a better use of system resources through its use of the extensive graphics intelligence found in Tektronix terminals and workstations. By employing these local graphic functions, TekniCAD users can accomplish the drafting tasks with only modest use of the system resources.

Paving the Way to Engineering Automation

Like the automotive subcontractor in our second case history, many small companies are discovering how the TekniCAD system plus CADDport can provide an entry point into engineering automation. Because of TekniCAD's compatibility with major CAD systems down the line, the initial investment of these companies in engineering hardware and software is protected.

A less obvious but equally important advantage is protection of human resources. All drawings that have been stored in TekniCAD may be moved up into the CAD system when the time comes. There, they can be used as a starting point for building the new CAD system's 3-dimensional design database.

For companies just beginning to automate their engineering functions, TekniCAD plus CADDport allows the creation of high quality drawings without extensive training or knowledge of "computerese." It is an easy system to learn to use, with a flexible menu-driven user interface that uses standard drafting terminology. Users have the option of control keys, thumbwheels or a joydisk. At the same time, more sophisticated features, such as macros, control functions, and direct commands, are available for the experienced user.

Trend Toward Integration

Many industry experts feel that integration of all phases of product manufacture, from initial design to factory production, is the trend of the future. This will include linking the design process with many stations along the production line, including manufacturing test, quality control, numerical control, etc.

Using CADDport to link CAD systems to TekniCAD is another example of appropriate factory integration. For increasing numbers of companies, CADDport represents a small but important step toward the ultimate goal of complete factory integration.

[1] Arlo F. Ames, Sharon K. Fletcher, "Making IGES Work for CAD/CAM Data Exchange," Society of Manufacturing Engineers Technical Paper MS85-1110, Dearborn, Michigan, 1985.

PLOT 10® GKS is First to Pass Certification Tests

In February, 1986, Tek's PLOT 10 Graphical Kernel System (GKS) became the first commercially available GKS package to be tested and certified as a conforming implementation of the GKS International Standard for graphics software development systems.

"The intention of graphics standards is to provide device independence to the users of computer graphics. Certification provides assurance that the promise of standards will be delivered," says David Straayer, a senior Tek engineer and participant on both the American National Standards Institute (ANSI) and International Standards Organization (ISO) committees. "We at Tektronix are proud that PLOT 10 GKS is the first to be certified."

Testing was performed by the Gesellschaft fur Mathematik und Datenverarbeitung (GMD) MBH BONN, a leading German research institution and testing laboratory. GMD is the first officially recognized testing laboratory for GKS. The suite of test programs was developed with funding provided by the European Common Market.

GKS was adopted by ANSI as X3.124-1985 in June, 1985, and by ISO as International Standard 7942 on July 15, 1985. GKS was pioneered in West Germany as Deutsche Institut fur Normung (DIN) standard 66 252, and is the product of international cooperation to provide standards for computers and data processing.

Although the GMD certification applies directly to only the DIN standard, the functional equivalence of national and international standards based on GKS means that application software developers in the United States, West Germany, the United Kingdom, or any other nation that has adopted GKS can now have confidence that PLOT 10 GKS is a conforming implementation of the standard.

The whys and hows of local area networking, with an emphasis on the LAN implementation of the 6130 Intelligent Graphics Workstation.

6130 LAN: OSI-Standard Networking with Value-Added Services

by Al Paylor
Senior Systems Analyst
Tektronix, Inc.
Phoenix, AZ
and
Clarence Peckham
Graphics Workstation Div. Engineering
Tektronix, Inc.
Wilsonville, OR

In the past few years, design teams have turned increasingly to workstations as an answer to some of the problems inherent in using large centralized computers. The system cost and maintenance, plus unpredictable turnaround time and the expense of down-time and lost engineering hours—all can have a negative impact on a company's profitability. Stand-alone workstations have their own potential drawbacks, however. Working as a team requires that users be able to communicate easily, sharing files and testing results among group members. The solution to this dilemma—of needing dedicated workstations without giving up the communication among users of large systems—is to connect the workstations with a local area network (LAN).

With a LAN, users have the benefits of personal control over their computing environment as well as easy communication and data exchange with other users. In addition, a LAN makes it possible to efficiently share system peripherals and files. Sharing information through a local network has a significant impact on productivity. There are two obvious advantages of sharing peripherals: to distribute the cost of expensive or seldom-used peripherals and to benefit from a smaller, quieter workstation area. These advantages make a local area network an essential component of a successful workstation (Figure 1).

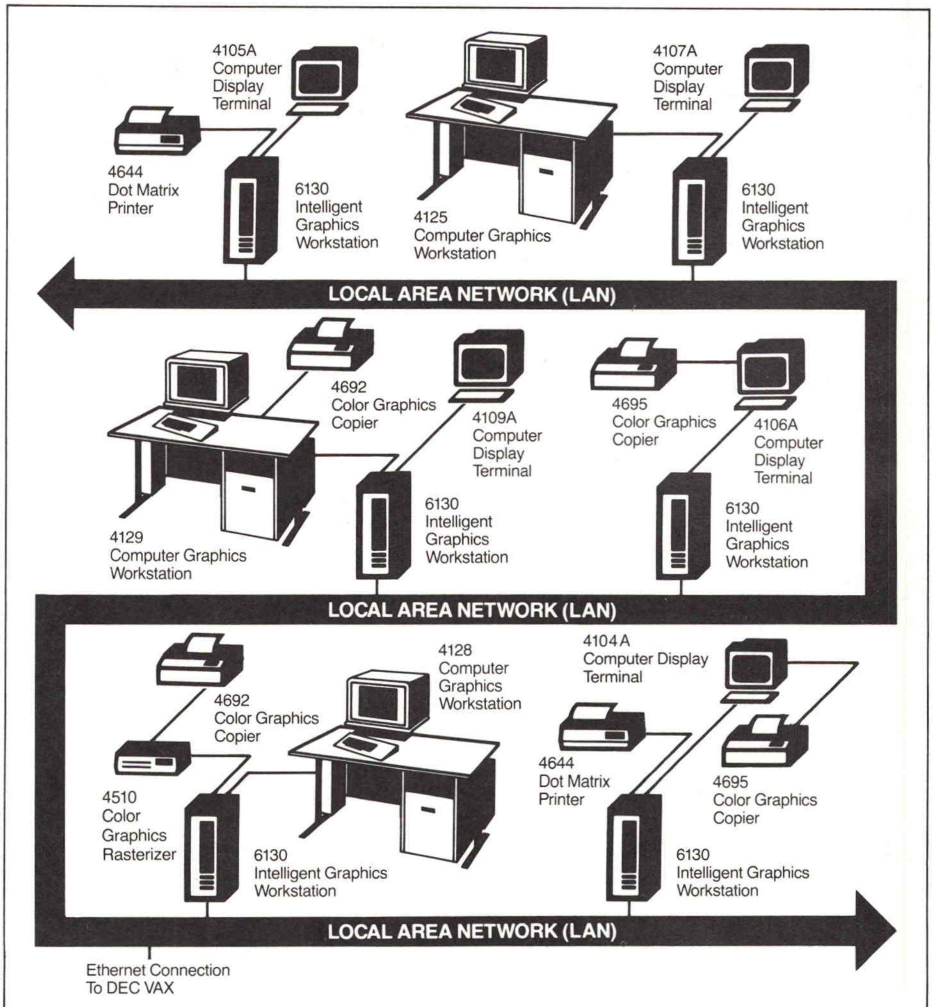


Figure 1

This article provides some general background information on LANs and some specifics on the LAN implementation of Tek's 6130 Intelligent Graphics Workstation.

Stars, Rings and Buses

There are four popular ways to configure a LAN. Figure 2 depicts these configurations and a common application for each.

The *star* network design optimizes transmission between the central node and its peripheral nodes, but compromises peripheral-to-peripheral communications. Another disadvantage is that the entire network may fail if the central node fails.

The *ring* network has advantages in easy and highly reliable node-to-neighbor-node communication. And unlike the star configuration, it does not burden any one node with the entire network maintenance task. However, transmission becomes slower and more difficult as more nodes are added to the network.

Many large corporations have *hybrid* networks, which can accommodate their large computers and which enjoy a high degree of reliability. Typically, these networks have evolved slowly and are customized to fit the users' particular needs. The greatest disadvantage to these

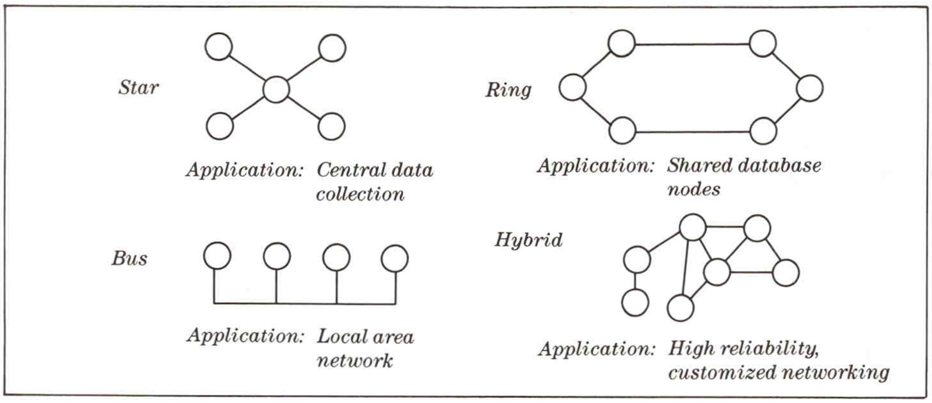


Figure 2. Common network interfaces

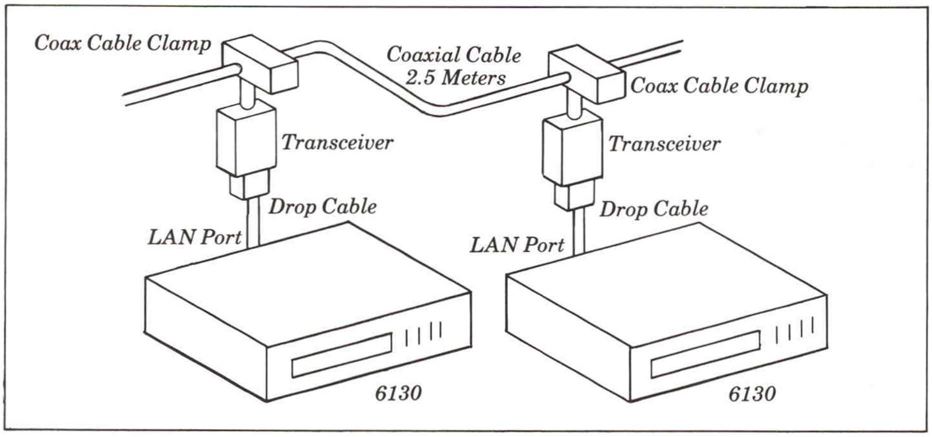


Figure 3. 6130 LAN connection.

networks is the high cost of construction.

Simplest in implementation and most common for today's LAN is the *bus* structure. It allows very high raw data rates and easy access in adding new nodes. In addition, a single node failure crashes only that particular node.

ISO's Open Systems Interconnection Model

In 1979, the International Standards Organization (ISO), as a step toward international standardization, proposed a model for computer network architecture. This model, called the Open Systems Interconnection (OSI) model, is shown in Table 1. The model describes both the physical and logical connection in networking communication systems, and has become the most widely accepted model for local networks.

The seven-layer structure of the OSI model defines functions and requirements for each layer. Different networks may implement different portions of the layers using various techniques. Typically, layers 1 and 2 are physical and are implemented in hardware. Layers 3 through 6 are implemented in software. Layer 7, also software, may not even recognize that the system it is working on is a network.

Ethernet™

Ethernet is a bus-oriented LAN system that solves the basic problem of equipment interconnection as addressed in layers 1 and 2 of the OSI model, but leaves the specification of layers 3–6 to the designer. Originally designed by Xerox Corporation in the early 1970s, Ethernet served in the late 1970s as the model for the IEEE 802.3 standard. The 6130 LAN is based on Ethernet and conforms to the OSI model. Figure 3 shows the 6130 Ethernet connection.

The physical layer of the OSI model, layer 1, defines the physical characteristics of the transmission. It first receives the data in binary form from the data link layer (layer 2), changes it into a phase-encoded signal, and transmits it on the coaxial cable.

In addition to transmitting and receiving data, the physical layer also transmits a carrier signal on the network. The presence of a carrier signal indicates that the network is in use. Each node monitors the network and refrains from transmitting if a carrier signal is present.

If more than one node transmits at the same time, a *collision* results. The physical layer detects the collision and notifies the data link layer. The data link layer then steps through a sequence to retransmit the data without contention.

The data link layer of the OSI model, layer 2, is also performed by Ethernet. This second layer builds on the physical layer and performs the following functions:

- Data framing (dividing the data into blocks called *frames*)
- Generation and decoding of network addresses (specifying the destination and source addresses of the data)
- Error detection (using a 4-byte cyclic redundancy check [CRC] value in each frame to detect errors)
- Network channel allocation
- Collision resolution

When data is received, the data link layer removes the framing information before passing the data to higher layers. Before data is transmitted, the data link layer waits for a signal from the physical layer indicating that the network is not busy, then passes the frame to the physical layer.

If more than one node transmits data simultaneously, the data link layer resolves the collision by following the channel-access protocol known as Carrier-Sense Multiple-Access with Collision-Detect (CSMA/CD). The CSMA/CD protocol dictates that if a simultaneous transmission occurs, each sending station waits a random amount of time and then retransmits. This is analogous to normal human communication in which no one speaks until there is quiet, and if two or more people start to speak simultaneously, random individual hesitations are enough to resolve the "collision."

Networking on the Tektronix 6130

The Tektronix 6130 workstation is an example of a workstation that uses the Ethernet to accomplish the functions of the lowest two layers of the ISO model, then provides a complete set of services for the rest of the layers of the model. These LAN services are standard with the 6130, giving users the power of a large system while still maintaining the benefits of local control and dedicated computing power. The 6130 LAN provides several utilities to aid in administering and using the network. A Distributed File System (DFS), as well as standard Unix™

4.2 bsd networking features such as remote login (rlogin), remote file copy (rcp), and remote command execution (rsh), allow the user to access other nodes to utilize more fully the power inherent in networking. Users can log on to "remote" workstations, execute commands, view and copy files, and send mail to other users.

Ease of Use

Three 6130 utilities aid in administering the network: *Nettest*, *Netconfig*, and *Nameserver*.

The *Nettest* utility allows the system administrator to verify that the network is usable. *Nettest* performs several tests, including echoing data between workstations and performing a time domain reflectometry test both to verify that the coaxial cable is usable and to isolate faulty taps on the cable. *Nettest* thus provides a quick method of analyzing the Ethernet cable without requiring external test equipment.

Netconfig performs all the steps necessary to install a workstation on the network. Some network implementations require the user to edit several files and build command scripts in order to install a new workstation. With *Netconfig*, however, the system administrator or user can add or remove a workstation on the network in a matter of minutes. The only knowledge required of the user is the desired host name of the workstation and the network number of the Ethernet network.

The *Nameserver* utility automatically maintains an up-to-date list of all host machines on the network, so that users can designate workstations by name rather than by their physical address on the network. For example, consider the remote copy command:

rcp hostname:fname fname

In this example the file is copied from the remote host *hostname* to the local workstation. The *Nameserver* utility provides the translation from *hostname* to workstation address.

Table 1. The OSI Model

Layer	Services Provided
7. Application	Systems utilities and application software *What do I want to do with the information?*
6. Presentation	Data representations and formats *How do I get the data into the correct format for my system?*
5. Session	Data management *Once I receive all the data, how do I put it back together again?*
4. Transport	End-to-end error control *How are errors handled in the flow of data from one node to another?*
3. Network	Data routing and switching *How does data get from one node to another?*
2. Link	Link access control *Who talks when?*
1. Physical	Voltage signals and levels *How are 1s and 0s passed?*

The Distributed File System

The Distributed File System (which was discussed more thoroughly in the Winter, 1986, issue of *Tekniques*) integrates networking into Unix. With conventional Unix, files can be shared, but are logically and physically connected to individual workstations. With the DFS, disk files remain physically connected to the individual workstation but are logically the same file system from a user's viewpoint

The DFS lets users easily utilize the computer power of their own workstations on files that exist on remote workstations. This is especially useful for large projects that share common files or for applications that require access to the same set of files. The DFS also supports teams of users. Often, several organizations within a company will share the same LAN; however, they may not want to share the same Distributed File System. With the DFS, multiple and distinct distributed file systems can co-exist on the same LAN. For increased flexibility, the file systems can intersect; thus, files can be made available to certain groups or users but not to others.

Other Application-Layer Services

Besides the DFS and the networking utilities, the 6130 LAN provides application services (at the seventh layer of the ISO model) to facilitate transfers between workstations and systems on the network that are not Unix 4.2 bsd compatible—including DEC™ VAX™ hosts and the Department of Defense protocols in the ARPANET network. These 6130 LAN services include:

- **File Transfer Protocol** (FTP) provides the ability to transfer files reliably to and from other nodes with different operating systems.
- **Simple Mail Transfer Protocol** (SMTP) facilitates sending electronic mail between 6130 workstations and other systems on the network, including a VAX.
- **Virtual Terminal** (Telnet) protocol makes it possible to log on to and operate a remote host outside the 6130 as if the user's local workstation were a terminal.
- **Transport Control Protocol Internetworking Protocol** (TCP/IP) handles communications between a user program and other processes executing at the same workstation, at a different workstation on the LAN, or on a different network.

The need to share information among team members and to use system resources efficiently make a local area network a key requirement for any workstation product line. With Tek's 6130 LAN, users have the reliability and portability of the Ethernet protocol for the physical transferring of data, the sophistication of the Distributed File System and the 6130 LAN utilities to allow networking to appear transparent, and the power of the application-level protocols to give each workstation user the information and resources of an entire network.

An Intelligent Electronic Calendar: A Smalltalk-80™ Research Application

by Jeff Staley
Computer Research Laboratory
Tektronix, Inc.
Beaverton, OR

Conventional paper-and-pencil calendars are ripe for automation. They present a number of problems, many of which would be alleviated by automation. For instance, paper is not a permanent media; it is often lost or ruined. Paper calendars are difficult to maintain; changing information is often messy or impossible. They are also difficult to share. And they are inflexible in terms of time spans; hence several calendars are often required to completely meet the needs of a professional.

In a research project at Tek's Computer Research Lab's Artificial Intelligence Department, we are resolving many of these problems with a computerized calendar system that keeps a permanent record of a person's schedule. Changing the information on an electronic calendar is simple. A single calendar can cover different time spans: a day, a week, a month, or a year or more. An electronic calendar can share its information among a group of users, while respecting individual privacy.

An intelligent electronic calendar can also solve many of the problems associated with the usual methods for scheduling a meeting of several individuals. In most cases, the ordinary methods do not select the optimum time for the meeting. A meeting may be scheduled by making only one pass through the attendees, but, except for trivial cases, several passes would find a better solution. Or, when every attendee could not be contacted, a meeting might be scheduled at a default time. An intelligent electronic calendar can automate much of a person's mundane scheduling and can automatically keep track of and schedule repetitive meetings. To schedule a meeting, a group of intelligent calendars can communicate to determine the most appropriate time for the meeting.

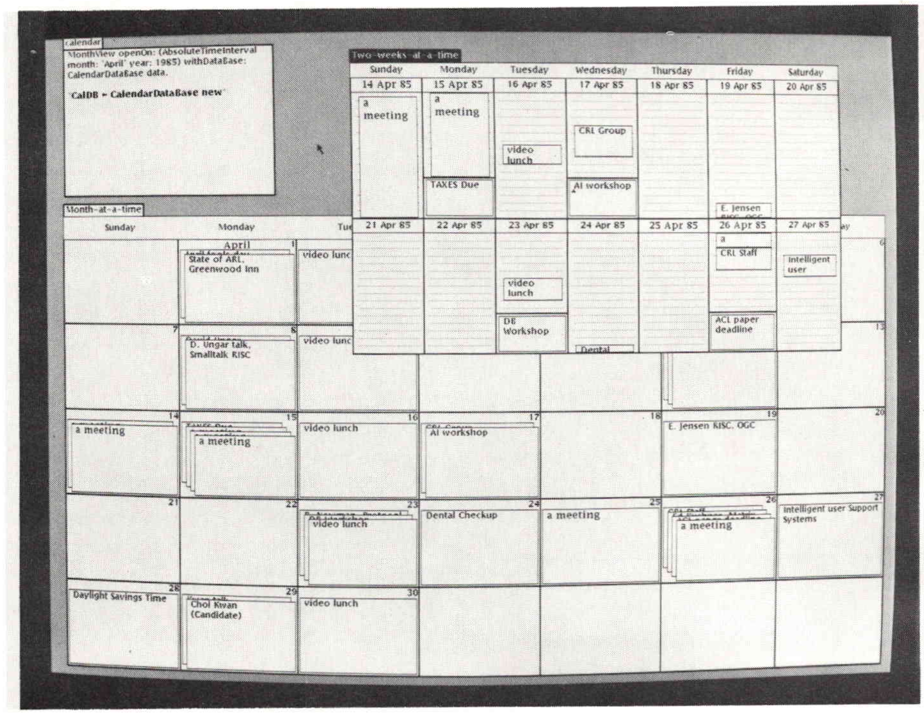


Figure 1. Calendars can be viewed on different scales, and on each scale will display only the appropriate events. The monthly calendar shows monthly meetings, vacation, etc., but no reminders or daily meetings.

Intelligent Project Support

Our Intelligent Electronic Calendar (IEC) is part of an Intelligent Support Environment, an evolving experimental environment designed to support the communication and information needs of project teams. In addition to the calendar, the support environment includes two major modules:

- The Project Encyclopedia, an information base that contains the history and state of the project and a schedule of events. The project encyclopedia has conceptual and temporal dimensions to capture the what and when of project evolution.
- The Elapsed Time Communication module, an enhanced electronic mail system that has the ability to evaluate rules, which allows it to filter and generate messages.

The IEC provides facilities to access the project encyclopedia, view multiple schedules, summarize a schedule, automatically schedule repetitive meetings, and automatically negotiate a meeting time.

We have built a prototype of the Intelligent Support Environment that runs on the 4400 Series Artificial Intelligence Systems and is implemented in the object-oriented language Smalltalk. Smalltalk is more than just a programming language for our system; its philosophy overtly influences our design. The Smalltalk environment provides approximations to many of the user interfaces we are developing and we have created interfaces largely by modifying the Smalltalk system.

IEC Facilities

Most people use a paper calendar to help organize and manage their schedule. By annotating a paper calendar, they can schedule meetings, note events, and scribble reminders to themselves. Paper calendars cover many time spans: a month, a week, a day, etc. The tradeoff one makes in choosing one scale over another is the amount of detail available versus one's ability to overview. For example, a daily calendar is appropriate for planning the events of a day, but is poor for quickly indicating how busy the next month will be.

One solution is to maintain several calendars, each at a different scale. Then one has the problem of having to enter some meetings on some calendars but not others. And worse is the problem of maintaining consistency among the calendars.

The IEC is similar to a paper calendar; meetings can be scheduled, events can be noted and reminders can be entered (see Figure 1). In addition, any time-dependent item can be entered onto the calendar. For example, if a user wants to send electronic mail one week from today, he can enter the piece of mail for the day it is to be sent. The calendar automatically sends the mail at the correct time.

Unlike a paper calendar, an intelligent electronic calendar can be viewed on different scales. On each scale, the IEC will display only those events that are appropriate. Thus, when the IEC is viewed at the day scale all of the items for that day are shown in full. However, changing the scale changes the items that are seen. For example, when a month is viewed, monthly meetings, milestone meetings, and vacations would be displayed, but no reminders or daily meetings. Thus, with one calendar many time spans can be covered whether they be for one or two days to a year or more.

Accessing Other Calendars

A person working at a remote site is often frustrated because he or she cannot access information that would be readily available at the office. This tends to be community information, since this is less likely to be moved to a remote site. It could be the manager's schedule, the availability of a conference room, or the availability of shared equipment. For example, suppose a group member is at home in the evening, adding some last minute touches to a presentation for the following day. She decides to add some 35 mm transparencies to the talk, necessitating a room change. It is unlikely that she can determine from home whether another room is available. However, with the room's calendar on-line, it is easy to determine that the room is available and then schedule it. Thus, not just people, but all time-restricted resources—meeting rooms, audio-visual equipment, etc.—can be managed with an IEC.

All intelligent calendars are available for perusal from anywhere within the system. However, access to a calendar should not be unconstrained. Even though anyone can communicate with any calendar, a calendar can restrict what may be seen. A room's calendar, for example, might be fairly cavalier with its permissions. A personal calendar, on the other hand, would place some restrictions on who can view it and modify it. For example, a personal calendar might allow a third party to view only meetings that are common to the viewer and the owner of the calendar. Or it might allow the viewer to see all meetings in full with the exception of personal meetings which, although displayed, are not annotated. In spite of the restrictions, a viewer would still be able to get a feel for the person's schedule.

It is important to realize that each intelligent calendar can be personalized. Whatever security restrictions the owner wishes must be codified. This is an important feature. Studies have found that personal concerns about privacy vary greatly; some individuals are closely guarded, while others are quite free.

Accessing the Encyclopedia via the Calendar

The repository of information for the system is the project encyclopedia. The IEC stores its knowledge of schedules, meetings, dependencies, and rules for behavior there. The encyclopedia can be viewed through the project browser, which presents the contents in a topical fashion. The IEC can provide a temporal view onto the information contained within the encyclopedia.

This interface is useful for an information-seeker who remembers approximately when the information was entered. For example, he might be interested in the minutes of a meeting that occurred towards the end of last month. He can scroll back to that time interval and inspect the meetings there. This interface is also an archiving device, allowing one to use the information for planning.

Repetitive Scheduling

Many meetings have a repetitive schedule. They might occur daily, every other week, or every second Monday. It would be unreasonable to expect the user to have to enter them individually. Hence, the IEC allows the meetings to be specified once (called a generator) with a description of when they will occur. The IEC will then create repeated instances of the meeting. Each instance inherits the properties of its generator, but can have its own annotation.

Consider a weekly group meeting to discuss the progress of a project. A single generator can describe the general properties of this weekly meeting—where it is held, the group members who typically attend, and that the purpose is to discuss project progress. Once this generator is entered in the calendar, multiple instances would be created, each occurring a week apart. (These instances are created as needed, not *ad infinitum*!) Every instance inherits identical attributes from the generator. With time, some of these generated meetings might become individualized because of changes to them. For instance, a meeting's location might be changed or its purpose might be enlarged.

Meetings that can be automatically generated fall into two groups. For most, the cycle is based on time, as in the above example. A second group is irregular, triggered by particular events in the world. For example, a follow-up group meeting might need to occur one week following an occasional departmental meeting. When the IEC detects the appropriate departmental meeting, it will automatically schedule an instance of the follow-up meeting for the following week.

Multi-Person Scheduling

One of the most difficult problems of scheduling a meeting is determining a mutually agreeable time for all of the attendees. When more than four or five persons are involved, it can become so difficult that a default time is often selected, which is seldom satisfactory to all. Moreover, the problem often becomes intractable when some of the attendees are not available for consultation. We plan for the IEC to have the ability to automatically schedule a meeting among a group of fellow IEC users.

To schedule a meeting, a user initiates the automatic scheduling by creating a meeting node on the calendar. In addition to the usual description of the meeting (title, purpose, attendees, agenda, location, and so forth), he would include a general description of when the meeting should be scheduled, such as "beginning of next month," or "end of next week." He would indicate this time interval by graphically selecting the region on his calendar. With this information, his IEC begins the process of negotiating with the other attendees' IECs with the goal of finding a mutually agreeable time. If no common free slot is found, compromises will have to be made. Someone will have to reschedule another meeting, or one of the personal constraints will have to be waived.

A user's calendar can not negotiate in a vacuum. If it is to schedule events for a user satisfactorily, it must have explicit guidelines as to how the user wants his schedule to be arranged. It must have knowledge about when the user likes to come and leave work, how long and how important lunch is, what meetings are important, how busy particular weeks and/or days are, how important a balanced day and week are, and much other information particular to the user. These constraints are coded as rules in the system.

Integration with Other Parts of the Project Support System

The calendar is not an autonomous system. It shares its world with the project encyclopedia and the communication system. The entire support environment strives towards a level of integration that encourages synergism. In fact, much of the apparent intelligence comes about because the modules can communicate freely. The mailer modifies its behavior when a user is on a lengthy vacation. Only by communicating with the calendar system can the mailer determine when a person is scheduled to leave and return so that appropriate actions, such as redirecting mail, should occur.

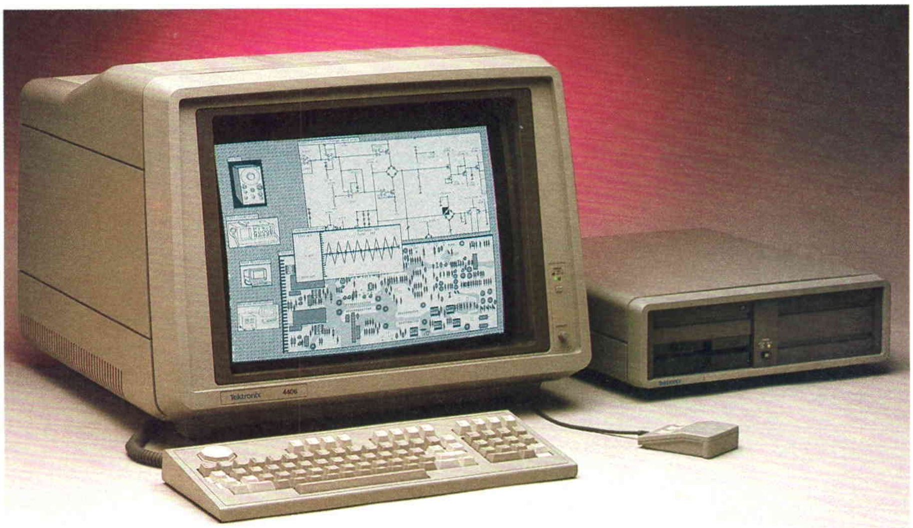
We feel that this integration has the desirable effect of encouraging a trend towards more information being stored. By allowing convenient migration, information tends to be retained in its full form.

Conclusion

In our research project, the use of the 4400 Series Smalltalk environment has allowed us to quickly prototype an environment that supports a project team. The 4404's high-resolution, bit-mapped screen, coupled with Smalltalk's powerful windowing facilities, has allowed us to design interfaces with ease. Moreover, Smalltalk's object orientation has supported our attempt to integrate three complex software systems. The IEC is just an example of what can be accomplished when the proper environment is available.

4405/06 Remove OOPS Constraint

by **Jeff McKenna**
AIM Marketing
Tektronix, Inc.
Wilsonville, OR



In an object-oriented programming system such as Smalltalk-80™, the number of possible objects is a fundamental limitation to a program and its data. In current versions of the Smalltalk-80 system, the number of possible objects is determined by the number of Object Pointers, otherwise known as OOPS. As specified in the standard Smalltalk reference, Adele Goldberg and David Robson's *Smalltalk-80: The Language and Its Implementation*, 32, 767 objects are possible. To extend Smalltalk's ability to handle significant, real-world applications development, Tektronix' new 4405 and 4406 Artificial Intelligence Systems include a new Smalltalk interpreter and virtual image that effectively remove the OOPS limitation as well as improving the Smalltalk speed.

With this new large-object-space (LOS) Smalltalk system, programmers can develop very large applications with the full confidence that the number of objects available will always be sufficient. With the object space limitation removed, the effective constraint becomes the system virtual memory size. In the 4405 and the 4406 systems, virtual memory size is 32 megabytes—ample for serious applications.

Applications that have been developed on the 4404 system using the previous Smalltalk system can be run on the 4405 and 4406 systems without modification. These applications will run faster due to the faster basic speed of the hardware. To take advantage of the LOS interpreter, existing Smalltalk applications can be easily converted to run using LOS.

AddSys-3000™: Graphic Software Tools for Advanced Device Functionality

by Jim Fitzpatrick
Dick Linville
Precision Visuals, Inc.
Boulder, Colorado

Applications programmers who use the device-independent, DI-3000® extended graphics software tools package from Precision Visuals, Inc., have always been able to capitalize on the powerful device features of the Tektronix 4107, 4109, 4115, and 4125 Computer Graphics Terminals. A newly introduced PVI product family, AddSys-3000, provides FORTRAN program access to special hardware features of these Tektronix display terminals.

Newly supported features include full segmentation support, dynamic segment dragging, image transformations, 32-bit coordinate mode, GIN rubberbanding and stroke input with time and distance filtering.

AddSys-3000 makes use of advanced device functionality, facilitating easier development of application programs that execute faster with reduced demand on host CPU and memory resources.

Just as DI-3000 manages the intricacies of drawing graphics primitives, maintains the segment display list and other graphics operations, AddSys-3000 additionally manages the details of many Tek-specific terminal functions, such as:

- Device-resident display list management
- Multiple views
- Segment editing
- Segment list operations
- Pixel operations

Special subroutines selectively disconnect the normal DI-3000 Software Segment Storage utility as well as allow bulk data transfers. AddSys-3000 incorporates a new, advanced device-driven architecture that increases information throughput and decreases CPU time for graphics input.

To help programmers learn to use Tektronix capabilities more quickly, support documentation includes a concepts and examples section on how to utilize the built-in Tektronix terminal functions.

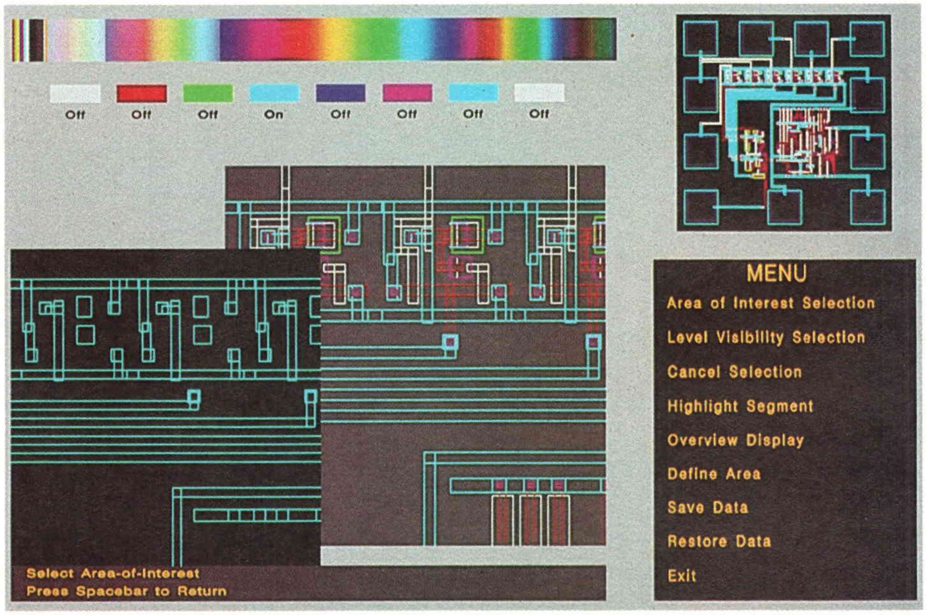


Figure 1. Multiple views and interactive menus give users better tools to maximize accuracy and productivity. Real-time pan and zoom, user-defined views, and dialog areas for communicating to the application program increase efficiency of developers' time.

Multiple Views

AddSys-3000 takes advantage of the Tektronix terminals' ability to maintain multiple views, transferring numerous viewing functions to hardware and relieving the computing load on the host (see Figure 1). The FORTRAN interface enables the programmer to partition the terminal screen into as many as 64 individual views to tailor the display to suit the application. Graphic information may be directed selectively to different views, enabling side-by-side or overlapping comparisons of related information, to facilitate comprehension. Various projections of an architectural rendering, or a series of graphs analyzing trends or seasonal factors are just two ways in which multiple views can be used.

Because the application can control the terminal's viewing system for window and viewport operations, the need for the software window and viewport transformation is eliminated. If the graphics consists of retained display list segments, the operator may use the local viewing system on the Tek terminals to view a selected area more closely. With the push of a button, the operator can redisplay the segments in any view with a new window. The terminal clips, maintains, and displays a unique list of segments for each view in the correct viewport on the screen.

Because of the Tek terminals' ability to "cluster" views, the application can apply a certain operation to any one view in the cluster and it will also affect the remaining views in that cluster. For instance, all views belonging to a cluster could have their window changed or updated with a single command.

Segment Instancing

AddSys-3000 allows the application program to store a segment definition in the terminal, then reference that segment within the description of another segment. To create four star symbols on a map, for example, the program would define a segment consisting of the symbol. Then, instead of storing an entirely new, independent segment for each of the next stars, the program would reference the original segment each time to create an "instance" of the star as needed.

Segment instancing reduces the amount of terminal memory required to store copies of segments. An added benefit is that any segment-attribute operation made on the "parent" segment will be reflected in the "child" instances contained in that segment. For example, rotating the segment containing instances of the star would cause the instances to rotate identically. This hierarchical display list structure is very useful in many sophisticated applications, such as cartography, architectural design, genetic engineering, VLSI, and molecular modeling.

Segment Editing

By accessing the terminals' segment editing capabilities, AddSys-3000 allows for modification of previously created retained segments. The segments stored in the terminals' memory may be changed and then redisplayed locally in their modified form. To redefine the segment, the host program need only send the editing commands and any new primitive data to the terminal, rather than the entire sequence of commands. The application benefits from reduced terminal communication and enhanced image refresh speed. In the case of the star symbol on the map, the application could modify the original description of the star and all instances of the star symbol would be changed.

The editing commands available allow applications to delete part of a segment, insert primitives into a segment, or replace a portion of a retained segment. Segment editing provides advanced graphics functionality while reducing host overhead (Figure 2).

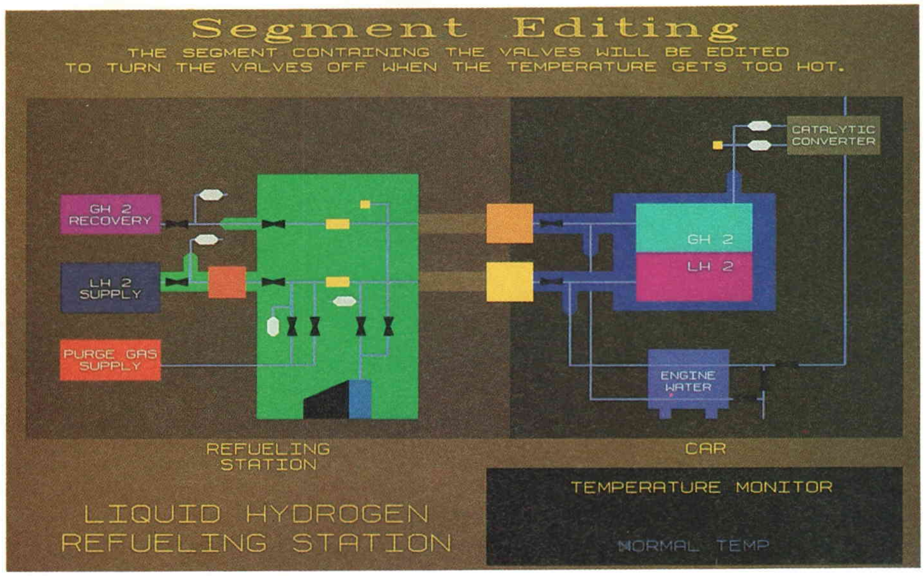


Figure 2. In this process control application, which monitors dynamic temperatures in a liquid hydrogen refueling station, segment editing allows the contents of a retained segment to be altered at the device level, after the segment has been closed.

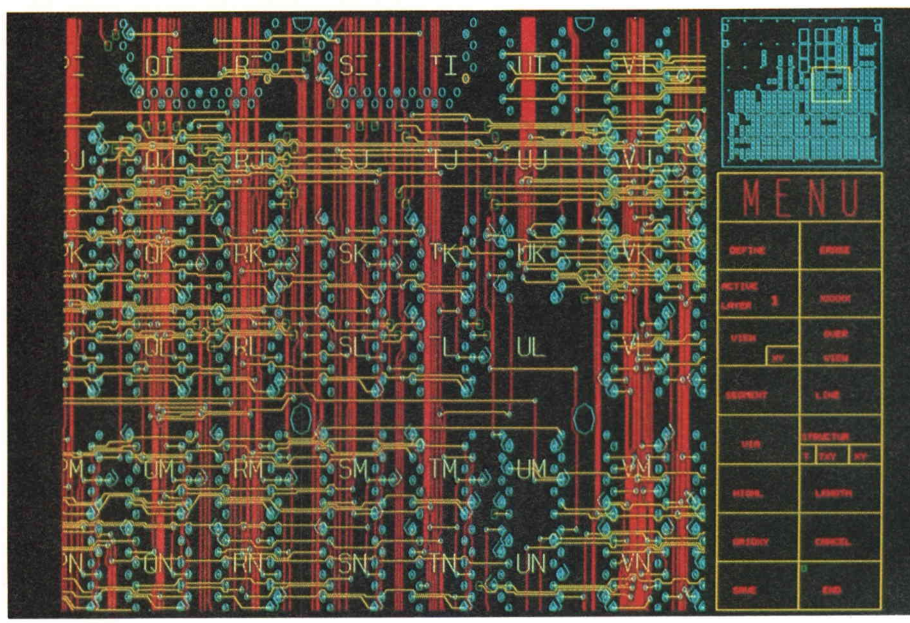


Figure 3. AddSys-3000's advanced device-driver methodology and optional host-resident segment storage provide segment instancing and segment list operations to help designers create complex and accurate images.

Segment List Operations

Locally stored picture segments make a major contribution to more efficient graphics. Special routines in AddSys-3000 promote further efficiency and convenience by allowing the application programmer to set or modify the attributes of a group of segments with a single command. Segments meeting certain criteria can be grouped together in an array. Then, with one command, a particular list of segments can be acted upon. For example, segments defining resistors could be grouped according to their value, tolerance, or wattage. One command could be issued to make visible, or highlight, or delete, a particular group of segments.

Pixel Operations

Programmers who are processing pictorial information or using pop-up menus will appreciate AddSys-3000's pixel routines, which give control over individual pixels or groups of pixels on the display screen. For example, some application images that will benefit from the pixel operations include medical scan techniques, satellite and aerial photography. In most cases these images are composed of mosaic pieces too small to be adequately described by panels and impossible to describe with vectors.

Pixel regions can be copied to other areas of the screen, or processed with one another through a selection of logical and arithmetic functions. AddSys-3000 handles the device-explicit commands and encoding of numeric parameters; the programmer is free to concentrate on the imaging task.

Bulk Data Transfer

A special subroutine in AddSys-3000 transmits data directly to the graphics memory of the Tektronix terminal, bypassing unnecessary, intermediate levels of processing. The data may be a word-per-pixel representation or run-length encoded for more economical data transfers. The image is quickly displayed on the screen.

Image processing applications can be integrated into existing DI-3000 applications. Or, new imaging applications can be created. Bulk data transfer is especially useful when combined with DI-3000 graphics. The host processes the image, determining the pixel-by-pixel display, then transmits the pixel data to the terminal's image memory. The application can then draw graphic primitives such as lines, circles, polygons, or markers which overlay the image. For example, the bulk data could be a processed Landsat image; the overlaying graphics could display the political boundaries.

Host-Resident Segment Storage

Included in AddSys-3000 is a subroutine that controls the use of DI-3000 Host-Resident Segment Storage (HRSS). Retained segments can be created and sent directly to the Tektronix terminal without being simultaneously preserved in HRSS. In applications where performance is critical, HRSS can be disabled. Host processing and memory usage are reduced, and terminal response is improved (Figure 3).

Alternatively, when segments are to be saved for later use on the host or for display on other devices, HRSS may then be enabled. This HRSS control gives the programmer greater flexibility and latitude in designing applications.

Independence With Performance

Over the years, DI-3000 has provided the software tools for application programmers whose specialized needs require in-house development of software. These software tools have assisted programming efforts by managing the intricacies of graphics operations and graphics display hardware.

To achieve portability of software, DI-3000 conforms to the SIGGRAPH Core proposal. However, device independence often sacrifices optimal use of advanced features. Graphics display hardware often makes technological advances while leaving graphics standards behind. Hardware vendors continue to include more sophisticated graphics operations in their designs only to have software tools packages not making use of the latest developments.

DI-3000 and AddSys-3000 strive to provide the optimum balance between independence and performance. The programmer can retain the device-independent routines developed in DI-3000, or use AddSys-3000 to take advantage of the device-specific functions. The Tektronix device offloads the host by performing specialized graphics functions on the terminal's local display list.

AddSys-3000 simplifies the integration of new, more productive display technologies into established applications programs that also access the modular device independence of DI-3000.

Photography for this article courtesy of PVI.

IDG Customer Training Workshops

Tektronix Information Display Group's Customer Training Workshops for June through mid-November, 1986, are listed below.

This schedule includes the addition of the 4120 Series Workshop, which covers the 4125, 4128 3-D wireframe, and 4129 3-D shaded surfaces. The workshop consists of lectures, reinforced with extensive structured labs.

The IDG Customer Training Workshops cost $800 per student, or $8,000 for an on-site class; the maximum class size for any of the workshops, on-site or at a Tektronix location, is 12 (3 students per workstation maximum). The fee does not include food, lodging or transportation. Register by calling the Customer Training Registrar at (503) 685-3412 (**New Phone Number**).

4120 Series User Training (5 Day)

Aug. 18–22	Gaithersburg, MD
Nov. 10–14	Santa Clara, CA

Introduction to UTEK (5 Day)

Jun. 16–20	Boston
Aug. 4–8	Irvine, CA
Oct. 6–10	Gaithersburg, MD

Computer-Aided Drafting (5 Day)

Jun. 16–20	Santa Clara, CA
Jul. 21–25	Gaithersburg, MD
Aug. 11–15	Dallas
Sep. 22–26	Boston
Oct. 13–17	Santa Clara, CA
Nov. 10–14	Gaithersburg, MD

GKS (4 Day)

Jul. 14–17	Irvine, CA
Sep. 8–11	Gaithersburg, MD
Oct. 13–16	Dallas

IGL (4 Day)

Sep. 8–11	Santa Clara, CA

Local Programmability (5 Day)

Contact the IDG Customer Training Registrar at (503) 685-3412 for further information. This workshop is flex-schedule (date and location determined when sufficient attendees are registered).

Smalltalk-80™ (4 Day)

Classes in beginning and advanced Smalltalk-80 are offered through Artificial Intelligence Machines Marketing. The introductory Smalltalk class introduces the student to object-oriented programming and the Smalltalk-80 programming language and environment. Students will develop their own application, featuring an interactive graphical display. The advanced class covers topics such as Model, View and Controller, team programming with Smalltalk, and multi-process Smalltalk programming. Students enrolling in the introductory class should have previous experience with a high-level programming language; enrollees in the advanced class should have three to six months experience with Smalltalk. Customer in-house classes can also be arranged.

For a schedule of Smalltalk classes, contact Sandi Unger at (503) 685-2941. To arrange an in-house Smalltalk class, contact AIM Marketing at (503) 685-2947.

The new 4100 graphics standard builds on 4010-style graphics, incorporating state-of-the-art capabilities while maintaining compatibility with previously developed 4010 applications.

4100-Style Graphics: Extending the 4010 Standard

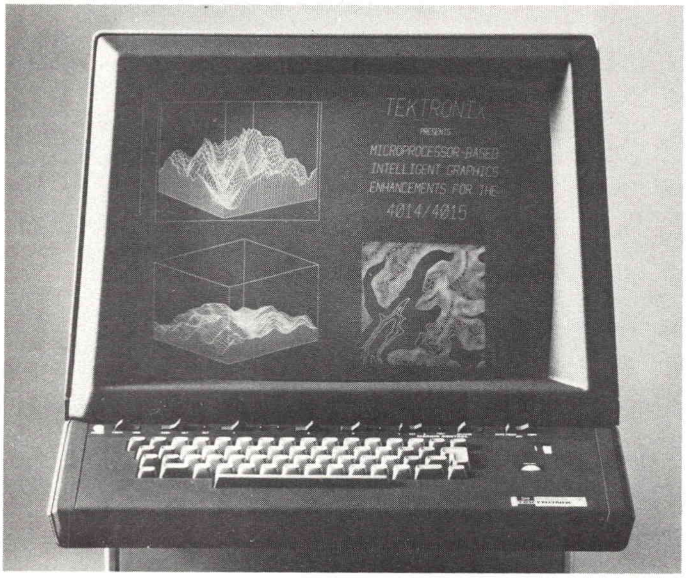


Figure 1. The DVST-based 4014, introduced in 1973, became a de facto standard and is still emulated by many software packages today.

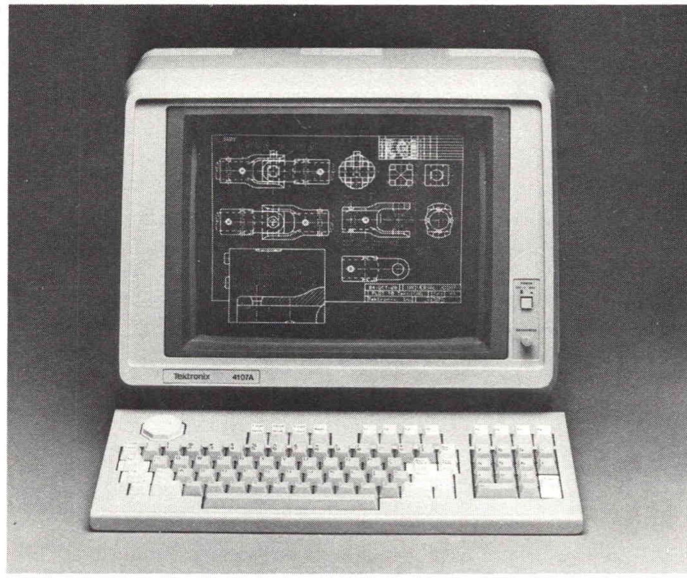


Figure 2. The new 4100-style graphics are 4010 compatible but offer extensive additional graphics capabilities, including color, segments, true zoom and pan, windows, viewports, and a wide range of enhanced graphics primitives.

by Merle Smith
Terminals Division Marketing
Tektronix, Inc.
Wilsonville, OR

For years, the Tektronix 4010 family of DVST terminals set industry standards for cost and performance in graphics terminals. Thousands of applications have been built around 4010 features and protocol, and a number of graphics vendors advertise "4010 compatibility." In the last three years, however, Tektronix has moved considerably beyond 4010 graphics to a new standard, not abandoning 4010 graphics but extending it to reflect current technology, standards, and market demand.

Because so many existing applications use the older 4010-style graphics, it's important to understand how the new "4100-style" graphics differ from 4010 graphics, and how these changes came about. Let's look first at 4010-style graphics. The most successful 4010 terminal was the 19-inch 4014 (Figure 1). To most people, "4010 graphics" means 4014 graphics. The main features of the 4014 were:

- **Alpha and Graph modes of operation.** Incoming data was interpreted as alphanumerics or graphics, depending on the terminal mode.
- **MOVEs and DRAWs**, with hardware implementation of dashed and dotted line styles. Vector definition of text also allowed multiple character sizes.
- **Encoded coordinates.** Encoded character strings were decoded into graphics commands by terminal circuitry. The "HiX, LoX" encoding scheme allowed parts of a graphics coordinate to be selectively omitted, further increasing graphics speed.
- **Single-point graphics input.** By pressing a key, the operator could send the crosshair position back to the host, allowing **interactive graphics** applications to be developed.

These features, at the 4014's resolution and price, created quite a stir in the mid-1970s. However, the industry has matured considerably in the last ten years. Integrated circuit performance has grown exponentially, while IC and memory prices have dropped, making local graphics storage and command processing relatively inexpensive. As a result, raster devices have largely replaced DVST devices as graphics displays. Raster devices are superior to DVST in two primary ways. First, because a raster picture is continually refreshed out of terminal memory, the picture can be **selectively updated**. A DVST display, by contrast, must be completely redrawn. Secondly, raster devices can display color, and color is now a standard requirement for many routine graphing applications, as well as in research and design environments.

In response to these changing conditions, Tektronix developed entirely new families of products, the 4100 Series and 4120 Series (Figure 2). These products are based solidly on the new color raster technology. They extend greatly the graphics capabilities available to users and to graphics applications programmers, while demonstrating a commitment to 4010 compatibility. Thus, coordinate encoding, addressing schemes, and escape sequences for MOVEs and DRAWs are compatible, to ensure that existing 4010 software need not be rewritten to run on the newer terminals.

Let's look at the major features of the new "4100-style" graphics standard:

- **Color.** From 4 to 256 colors can be displayed simultaneously, selected from a total color palette ranging from 64 to over 16 million color choices.
- **ROM-based instruction sets**, to off-load more computations from the host to the terminal.
- **Separate dialog and graphics areas,** to keep conversations with the computer or commands to the terminal separate from the graphics display.
- **Enhanced graphics primitives**—panels (enclosed areas), pre-defined fill patterns and colors, graphtext, markers for point plotting.
- **Segments.** The ANSI proposed Core standard and the ISO Graphical Kernel System (GKS) standard both provide for hierarchical graphical constructs called **segments**. Segments allow a program to take advantage of natural groupings, repetitions and hierarchies in a picture. Tektronix 4100, 4110 and 4120 Series products support **local segments**, off-loading burdened computers and minimizing data transmission time.
- **Windows and viewports.** Windows, along with local segment support, allow local zoom and pan operations, with local redraws performed much more quickly than if graphics are retransmitted from the host. Viewports allow multiple pictures to be displayed in different screen areas.
- **Enhanced graphic input (GIN)**, including input of multiple points (stroke mode operation) with one GIN command, and segment pick operations to facilitate the work of creating and editing interactive graphics.
- **Peripheral interface options**, with output to a variety of devices, including Tektronix 4690 Series Color Graphics Copiers.
- **Coaxial connection** into IBM® host environments.

In addition to these basic features, the 4120 Series includes firmware instruction sets to support 2D, 3D wireframe and 3D shaded surface displays.

What do all these new features mean to you as a user or a graphics programmer? They mean three things:

1. Software developed for the 4010 terminals will run on the 4100, 4110 and 4120 Series terminals, and most software will run unmodified. You don't have to spend time rewriting software that's already working for you.
2. Existing 4010 applications can be updated to take advantage of the new features. For example, you can add color commands to a 4010 application, thus enhancing the display with a minimum of effort.
3. Finally, the new products have been designed to be "upward compatible." Applications written for one member of the family will run on more full-featured members of the family. This gives you a range of compatible terminals and workstations that can be matched to the price/performance needs of each application, from the most simple to the most complex.

In this issue of *Tekniques*, we begin a series of articles dealing with the graphics capabilities of the new 4100 graphics standard. The series begins with an article on local segments, one of the most powerful features of 4100 graphics.

4100-Style Graphics

Segments—For Improved Graphics Interactivity

by Tom Ireland
Terminals Division Marketing
Tektronix, Inc.
Wilsonville, OR

Segments promote programming efficiency and application interactivity. Graphics commands are stored in compact form in the terminal, allowing redraws and zoom and pan operations to be performed locally.

Graphics data streams tend to be very large, and the time required to transmit them to the terminal can decrease the interactivity of graphics applications. Local graphics segments, a key feature of Tek 4100-style graphics, are one way to greatly reduce the amount of data transmitted to the terminal and thus increase interactivity.

What Is a Segment?

Before we define what a segment is, let's back up a bit. Most graphics terminals (including the Tektronix 4100 and 4120 Series) use a *vector graphics* or *graphics primitives* approach to creating pictures on the screen. The terminal recognizes a set of *graphics primitive commands*; when these commands are sent from the host computer, the terminal decodes them and uses the transmitted data to create a pattern of pixels on the display.

Vector graphics makes it possible to define a graphics display in terms of its intrinsic geometry. For example, a DRAW VECTOR command lets you draw a vector—a straight line. Other commands are used to draw arcs or circles, define closed areas (called *panels*) and fill them with colors or fill patterns, put text (graphtext) into pictures, and so on. (Non-vector-based approaches can be useful for specialized needs. For example, *pixel commands*, which let you define a picture in terms of the specific pixels that must be lighted on the screen, can be more suitable for some applications that require extreme realism.)

A segment is simply a collection of one or more graphics primitives that can be treated as a single object. Segments are part of both the ANSI Core standard and the ISO Graphical Kernel System (GKS) standard. Segments can be implemented at the host level or at the local terminal level. In this article, we'll be discussing local segments.

Segments greatly simplify the work of creating graphics displays, particularly when the display includes multiple instances of a shape. Segments are a *hierarchical graphical construct* that can be nested and called as subroutines. For instance, a tree could be made from three segments: a leaf, a branch with many leaves included or called, and a trunk with many branches included or called.

By building displays from segments, you can easily control how each object is displayed, and can modify the drawing, perhaps changing the size, color, location, and so forth, then redisplay it in its new form. Segments can also be grouped into classes; for example, you can make an entire segment class blink, rather than telling each individual segment to blink.

In addition to the time savings from reusing code, segments provide a number of other advantages. First, as we've already mentioned, segments reduce data transmission time, since modifications, scaling and so forth can be done by the terminal. Local redraws and zoom and pan operations can be done quickly, without communicating with the host at all. And local segments reduce the host's computation load, because the terminal can actually make copies of changes to local segments.

Segment vs. Non-Segment Graphics

Let's look further at the difference between non-segment graphics and local segment graphics. When a terminal lacks local segment support, graphics are handled according to the general scheme shown in Figure 2a. The terminal receives commands from some external source (host, keyboard or storage device), parses (or interprets) these commands and converts them into specific data stored in the terminal's graphics memory. (The graphics memory in a color terminal is composed of bit planes, with the number of bit planes determining the number of colors that the terminal can display at one time.)

Thus, what you see on a non-segment display is a direct result of the information stored in graphics memory. When the graphics memory changes, the display

```
*sgpivot 3400,1500
*sgopen 1
*lineindex 3
*sgpickid 2
*move 2500,1000
*draw 2000,1000
*draw 2000,2000
*draw 2500,2000
*curve arc 3000,1500 2500,1000
*move 3400,1500
*draw 3200,1500
*curve arc 3000,1500 3200,1500
*move 1600,1200
*draw 1800,1200
*curve arc 2000,1200 1800,1200
*move 1600,1800
*draw 1800,1800
*curve arc 2000,1800 1800,1800
*sgclose
*
```

Commands used to create segment in Figure 1A.

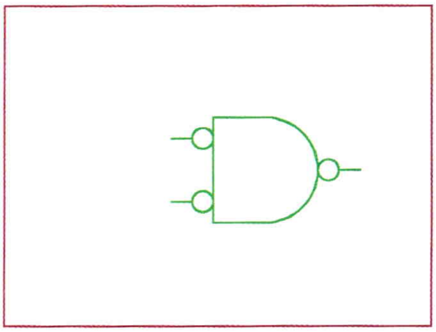


Figure 1A

```
*sgtransform 1 1,-1 1,-1 0,0 3000,2050
*
*sgpivot 3000,2050
*sgopen 3
*sginclude 1
*sgclose
*sgposition 3 3000,800
*renew
*
*sgpivot 1800,950
*sgopen 4
*sgcall 1 1800,950 none
*sgclose
*
```

Transforming and copying the segment via segment includes and segment calls (Figure 1B).

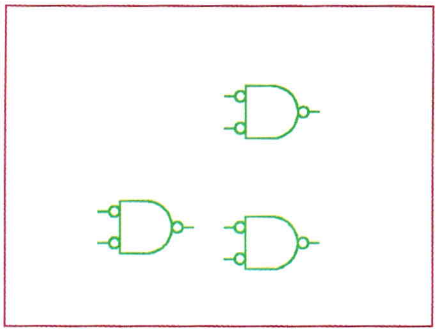
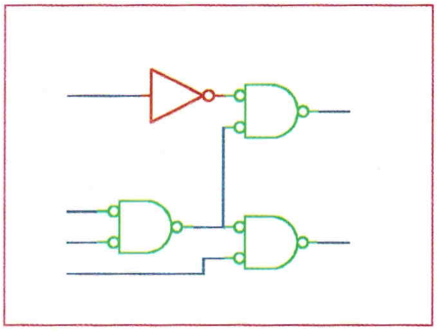


Figure 1B



The completed image.

```
*sgedit both
*sgreplace 1 1 1
*lineindex 5
*sgclose
*renew
*
```

Segment editing (Figure 1D). The called segment changes color; the included segment does not.

```
*sgmatchingclass 1
*sgclass 1,,1
*sgclass 3,,1
*sgclass 4,,1
*sgvisibility -3 no
*
```

Using segment classes to turn off the display visibility of the OR gates (Figure 1E).

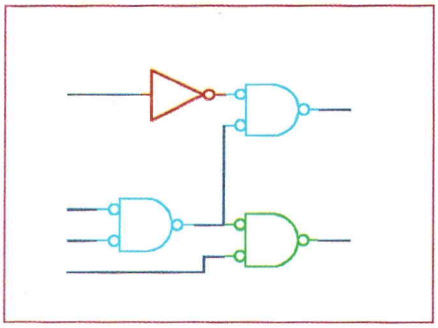


Figure 1D

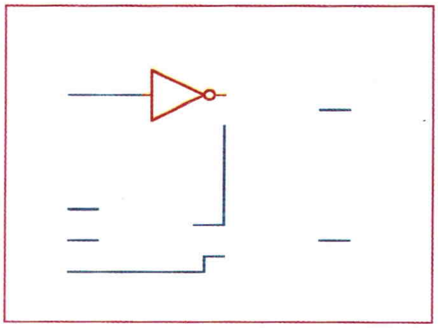


Figure 1E

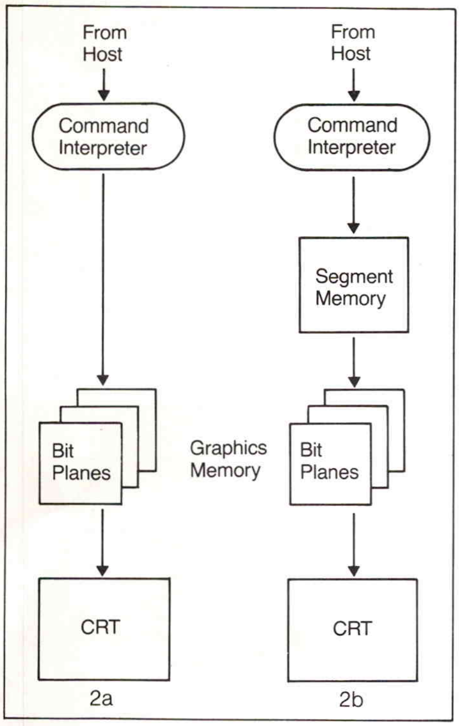


Figure 2. When a terminal lacks local segment support (2a), commands are decoded into pixel data and lost after execution. With Tek's 4100-style terminals, segment commands are stored locally in segment memory in a compact, display-list format (Figure 2b). Local segment storage allows the display to be updated without host interaction.

changes. In a non-segment terminal, this happens whenever the command interpreter receives a new command. There is no local storage of graphics commands, only of the pixel data in the bit-plane graphics memory. Once the graphics commands are parsed and executed, they are lost; once the screen is erased, the only way to recreate the display is to retransmit the entire sequence of commands from the host.

A terminal with local segment capabilities also has graphics memory that is scanned to create the pixel pattern on the display. However, when this terminal receives a segment command, it stores the command in a special memory called the segment memory (Figure 2b). The command is stored in a compact, *display list* format. Periodically, as determined by the terminal's *fixup level* parameter, the processor reexecutes the commands in the segment memory and reads the new display instructions into graphics memory to update the display.

In a terminal with local graphics segment support, then, the differences are twofold:

- Graphics commands, not just display patterns, are stored locally.
- The display is not modified until the commands from the segment memory are executed.

Defining a Segment

To create a local segment, you simply:

1. Issue a command to open the segment.
2. Issue the graphics primitive commands to draw the segment.
3. Issue a command to close the segment.

For example, the following sequence of commands would create a square as segment number 1. (For readability, we'll use the descriptive form of the commands.)

```
BEGIN SEGMENT 3
SELECT FILL PATTERN 37
BEGIN PANEL BOUNDARY 1000 1000 1
DRAW 2000 1000
DRAW 2000 2000
DRAW 1000 2000
END PANEL
END SEGMENT
```

When you define a segment, you can also set segment *attributes*. There are two types of attributes. *Static* attributes cannot be changed once the segment has been defined. Static attributes include the segment's *pivot point* (the point around which a segment can be rotated, scaled, etc.), as well as line style, line color, and so forth. On the 4128 and 4129, a segment's dimensionality (2D or 3D) is also a static attribute. *Dynamic* attributes include items such as position, rotation, scale and visibility, and can be modified after segment definition.

What Can You Do with a Segment?

Local segment support opens the door to all kinds of possibilities. For example, since the graphics display information is stored locally, you can clear the screen and locally redraw all the graphics stored in segments as often as you like. You can also move or *transform* a segment locally, without redefining the segment. For example, you can move a segment (or a class of segments) to a different screen location, scale them, rotate them, blank them from the display, or make them blink. You can control how a segment is displayed relative to other segments, that is, when they overlap.

Segments allow true zoom and pan to be performed locally. In fact, the terminal handles them as segment transformations. To zoom, you simply draw a box (or *set a window*) around the area to be enlarged; the terminal then automatically creates a segment definition that includes the items contained in the window. This segment is then scaled as a unit, maintaining correct proportions as the size is changed. The terminal recalculates the drawing, giving greater detail, rather than simply "blowing up" the image and giving a grainy appearance.

Panning is handled similarly, with the entire drawing defined as a segment. For existing 4010 applications, this means that you can turn a 4010 picture into a segment by simply inserting a BEGIN SEGMENT command at the beginning and an END SEGMENT command at the end. You can now display it on a 4100 series terminal, and zoom and pan locally!

When you're editing a complex display, local segments allow the terminal to save up several minor editing changes, then redraw the corrected display all at once. Time is saved, since you don't have to redraw the display after every change.

You can use the SET GIN CURSOR COMMAND to select a segment as the graphics input cursor, and move it across the screen with the terminal's joydisk or thumbwheels. For example, you might create a menu of graphics objects, with each object defined as a segment. You could pick items off the menu, move them across the screen and place a copy of the segment at any location, and then, using graphics input techniques, scale and rotate the object to a different size and position.

You can also specify *pick groups* within a segment, and later use these groups for segment editing or during graphic input pick operations.

For Further Information

Local graphics segments are one of the most useful features of Tek's 4100-style graphics. This article has discussed the benefits of local graphics segments and provided an introduction to using them. For detailed information on creating and using segments, refer to your terminal's programming manual.

Tektronix
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Group
Tekniques
Mail Stop 63-635
P.O. Box 1000
Wilsonville, Oregon 97070

BULK RATE
U.S. POSTAGE
PAID
TEKTRONIX, INC.

LI-FO

MICHAEL R OLIVERI

TEKNIQUES

Address Correction Requested—Forwarding and Return Postage Guaranteed.

15W-6307