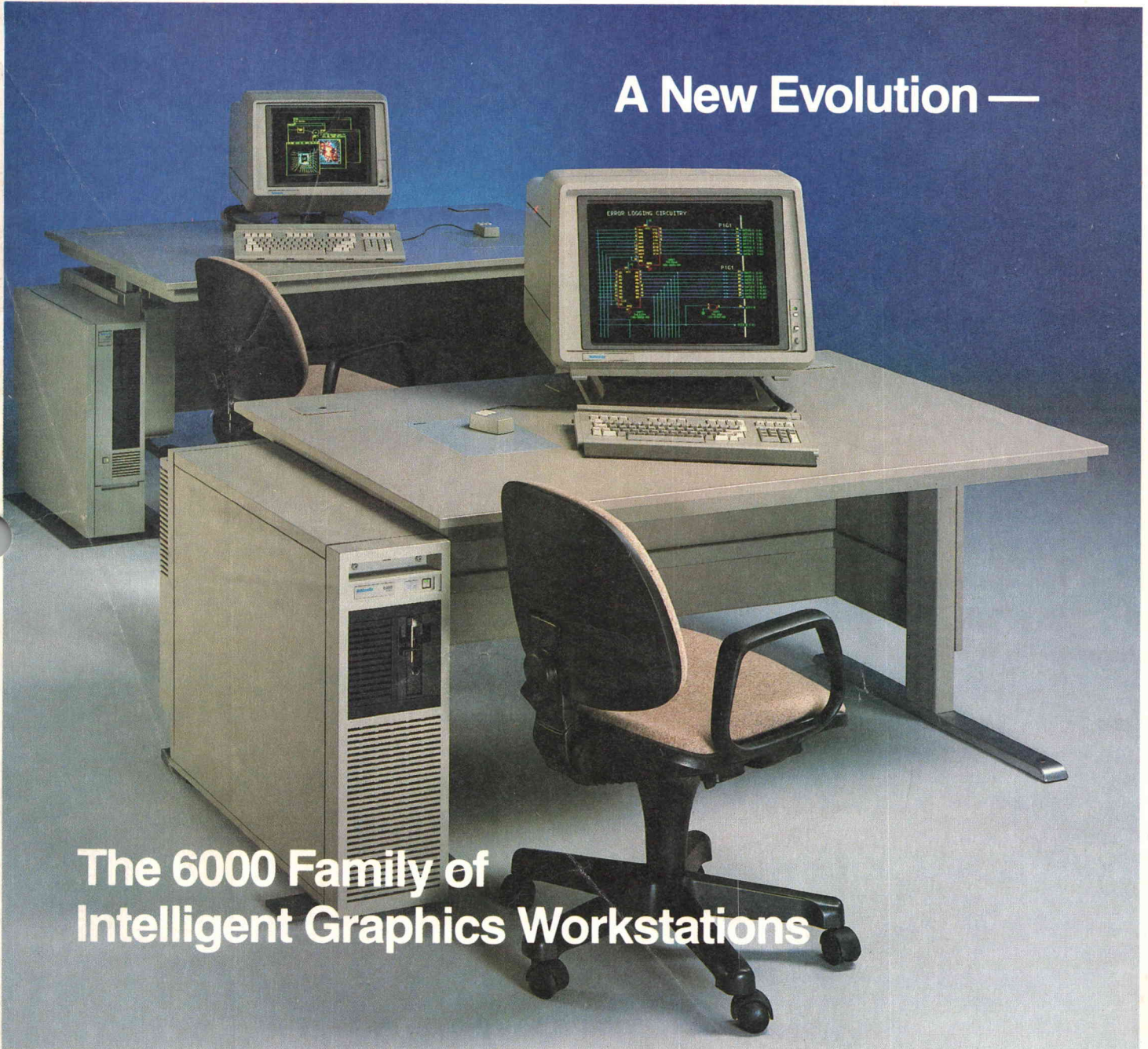


Tekniques

A New Evolution —



The 6000 Family of
Intelligent Graphics Workstations

Tektronix®
COMMITTED TO EXCELLENCE

Introducing the 6000 Family of Intelligent Graphics Workstations

A new family of products has been added to Tektronix' evolving line – the 6000 Family of Intelligent Graphics Workstations. From process and instrument control to the most complex engineering tasks, the 6000 Family provides the necessary computing capabilities, graphics technology and software support. Designed with an open-ended architecture, the workstations easily interface with existing systems and accommodate system expansion and reconfiguration when needs change.

The 6000 Family is made up of two distinct but software compatible product groups – the 6100 and 6200 Series. The products can be configured for a variety of applications, ranging from a satellite unit for instrument and process control to a powerful computer-aided engineering (CAE) workstation that supports multiple CPUs, multiple users and

On the cover:

The 6200 Series workstation pictured in the foreground and the 6100 Series unit shown in the background represent the 6000 Family's range of computing power and graphics capabilities. With six different configurations available, the family accommodates the requirements of a wide range of applications. Designed for easy expansion and integration with existing computing resources, all family members are based on operating system and programming language standards.

multiple displays per user for mechanical or electronic engineering design. The Intelligent Graphics Workstations can be used as stand-alone systems, networked together with standard interfaces or linked with a mainframe.

The 6000 Family supports a variety of displays – from the 4010, 4100 and 4110 Series terminals to a new line of highly-interactive displays. Combining the speed and interactivity of bit-blt technology with the superior resolution and information content of display list processing, these new displays feature window management, fast vector performance, smooth text scrolling, multiple fonts, variable cursors and area fill. Multiple, active windows, pop-up menus and a three-button mouse take advantage of bit-blt characteristics. (See related story on 6000 Family Display Subsystem.)

The 6000 Family employs National semiconductor's Series NS32000* microprocessors featuring a mainframe-like architecture and an instruction set designed for efficient execution of high-level language programs. (See related story on 6000 Family Workstation Architecture.)

All members of the 6000 Family are object code compatible to insure system compatibility. The task of porting software is simplified because an application that works on one member of the family will run on

other similarly-configured 6000 Family workstations. It also means that users can move from one workstation to another without having to learn new commands and protocols.

Software support for the workstations includes a variety of application programs, professional productivity packages, graphics libraries for application development, a UNIX*-based operating system and standard programming languages.

Application software from several independent software suppliers will be ported and available to customers. The established Tektronix Solution Vendor Program will be used to communicate the availability of these applications to customers. Solution Vendor companies announcing intent to port their software product to the 6000 Family are General Electric CAE International, Inc. (SDRC GEOMOD and SDRC SUPER-TAB); Manufacturing and Consulting Services, Inc. (ANVIL-4000); The MacNeal-Schwendler Corporation (MSC/GRASP); PDA Engineering (PATRAN); Precision Visuals, Inc. (DI-3000); and Swanson Analysis Systems, Inc. (ANSYS).

"The Environment," a group of integrated professional productivity software products developed for the workstation family, includes a graphics editor, document processor

TEKniques, the IDG Applications Newsletter, is published in the Information Display Group of Tektronix, Inc., Mail Stop 63-635, P.O. Box 1000, Wilsonville, Oregon 97070. It is distributed to TEKTRONIX computer and terminal users.

To submit articles to *TEKniques* write to the above address. Changes of address should be sent to the IDG Program Exchange serving your area (see Program Exchange addresses).

In This Issue

6000 Family

Introducing the 6000 Family of Intelligent Graphics Workstations	2
6000 Family Display Subsystem: Combining Graphics Worlds	5
The 6000 Family Workstation Architecture:	
Consistent Flexibility	9
A User Interface Management System For Workstation Productivity	14

4991S1 Graphics Input Workstation

Graphics Input Workstation Systemizes Hardcopy to Database Conversion	17
---	----

Regular Features

Programming Tips	19
Program Exchange	20
Tekniques Reply Card	21
Publications Update	23

Copyright © 1984, Tektronix, Inc.
All rights reserved.

and project management system. "The Environment" takes advantage of the 6000 Family integrated, bit-mapped display by utilizing a Casual User Interface (a menu-driven alternate to the UNIX operating system interface) and a User Interface Management System. (See related story on User Interface Management.) Additional general-purpose software support is offered by a relational database manager and Teknical Tools - packages that run on both 4000 and 6000 Family displays. The 6000 Family also runs other UNIX-compatible programs available in the marketplace. The 6120 and 6130 workstations can also run software compatible with the MS-DOS* operating system.

Application development tools supported by the 6000 Family include Tektronix graphics libraries: PLOT 10 Terminal Control System (TCS), PLOT 10 Interactive Graphics Library (IGL), and PLOT 10 TekniCAD Computer-Aided Drafting Software. In addition, an optimized version of GKS has been implemented for the new 6000 Family displays. Future Tektronix developed CAD/EE applications will also be based on these workstations.

UTek, the enhanced version of UNIX implemented by Tektronix includes the major features of Berkeley 4.2 and System V. UTek efficiently manages the local area network (LAN) and the processor's virtual memory and demand paging capabilities so that very large programs can be executed on the workstations. Language support includes high-performance compilers for ANSI FORTRAN 77, C, ISO Pascal and the proposed ANSI BASIC, which integrates graphics, program segmentation, file processing and structured programming concepts.

The 6110 Instrument Controller, a process and instrument control satellite unit, runs a real-time operating system (RTOS). Object code programs generated by other 6000 Family workstations in BASIC, C, Pascal or FORTRAN-77 can be downloaded to the 6110 via RS-232, IEEE 488 or flexible disk.

The 6110 features a full-function GPIB port that enables talk, listen and control functions. It also comes standard with 256KB of memory, dual RS-232 ports, and a 360KB flexible disk. A high-speed GPIB port with single character end-of-message detection and cache memory transfers is optionally available. Other options include an additional 512KB of memory, a 10 or 20MB Winchester disk, additional ports and a MultiBus* connector.



The 6130 Intelligent Graphics Workstation is designed for computer-aided engineering applications. The 6130 is shown here with a 6000 Family display (in the foreground), a 4107 computer display terminal (in the background) and a 4695 Color Graphics Copier.

The 6120 Intelligent Graphics Workstation addresses the needs of the data analysis market segment for scientific, engineering and instrument control applications. Proposed ANSI BASIC provides extensive graphics functions, extended I/O capabilities and interactive editing.

For increased power and capability, a subset of the UTek operating system is included with the BASIC system and can be accessed directly by the user. This creates an environment with powerful text-processing, communication, and file-handling tools.

The 6120 is offered standard with 1MB of memory, a floating point processor, a 360KB, 5 1/4-inch flexible disk and a 10MB Winchester disk. Interfacing is provided by dual RS-232, as well as GPIB and LAN ports.

Optionally available are a 13-inch monochrome display, a 13-inch, 16-color (from a palette of 64) display, a PC (personal computer) co-processor, expansion memory, an external Multibus board set, disk storage options and additional interfacing options. In addition, C, FORTRAN-77 and ISO Pascal compilers can be added (these require additional disk capacity and memory).

The 6130 Intelligent Graphics Workstation is designed for data analysis, software development and CAE applications. Standard 6130 features include the UTek* operating system,

a 20MB 5 1/4-inch Winchester disk, 360KB flexible disk, 1MB of memory, and a floating point processor. A 13-inch, 640-by-480 pixel, bit-mapped color display and keyboard are optional. For flexible interfacing, dual RS-232 ports, a GPIB and a LAN port are included in the base unit.

A PC co-processor option provides compatibility with the MS-DOS* operating system. Also available are expansion memory disk options, additional interface options, GKS level object buffer, a 15-inch, 640-by-480 pixel, bit-mapped color display and a 15-inch, 640-by-480 pixel, bit-mapped monochrome display.

With large memory capacity, independent 32-bit I/O processing and global bus architecture, the 6205 Intelligent Graphics Workstation is well suited for moderately demanding computer-aided design and computer-aided engineering applications. To deliver the necessary processing power, the 6205's central processor, an NS32016, is aided by a 32016 I/O Processor and a 32081 Floating Point Processor. Standard 6205 features include 1MB of error correcting code (ECC) RAM and a 40MB, 5 1/4-inch Winchester disk. An additional 9MB of memory can be added as well as a 40 or 80MB Winchester disk, 40MB cartridge tape, 9-track tape and 360KB flexible disk drive. Optional monochrome and color, high-resolution displays, each having a 32-bit display processor and bit-slice engine are supported. The 19-inch

color display features a 1024-by-768 pixel screen with 256 displayable colors from a palette of 16.8 million.

Flexible hardware interfacing is provided by the following standard ports:

- A LAN port
- An 8-bit, Centronics-compatible hardcopy port
- Four high-speed RS-232-C ports (19.2K baud), one of which can be configured as a high-speed RS-422 serial port.

The 6205 can be configured to act as a special-purpose node such as a file and peripheral server to the other workstations on a Local Area Network. It can be used, for example, as a network node through which other workstations can share peripherals, rather than having the peripherals attached to an individual user's workstation.

Similarly, such a node could act as a server for file administration or file access from any workstation. When configured as a server node, the 6205 can support two Multibus-interface peripherals, such as a 9-track tape drive or an SMD disk. At least one workstation on a Local Area Network must include a cartridge tape drive.

The 6205 can be upgraded to a 6210 or 6212 configuration by integrating faster 32032-based processing. Seven board slots are available on the system's global bus and peripheral interfacing capacity is expandable.


The 6210 Intelligent Graphics Workstation combines 32-bit processing power, mass storage, a Local Area Network (LAN) interface and optional high-resolution displays, enabling it to serve in multiuser environments and to take on demanding engineering tasks. The 6210 is well suited for mechanical engineering design and analysis as well as large-scale circuit board design, custom VLSI design or gate array design. Terminal emulation windows and host communications capabilities permit distributed processing in conjunction with existing host applications.

The 6210 is offered standard with a single 32032 applications processor, 32016 I/O Processor, 32081 Floating Point Processor, 1MB RAM and 40MB disk capacity. An additional 9MB of memory can be added, as well as an 80MB Winchester disk, 40MB cartridge tape, 9-track tape and 300MB disks. Optional monochrome and color high-resolution displays, each having a 32-bit, display processor and bit-slice engine, are supported.

The 6210 Intelligent Graphics Workstation can also be configured as a file and peripheral server for workstations on a LAN. It

supports large SMD disk drives, 9-track tape drives and hardcopy devices. The 6210 allows the sharing of these peripherals without impacting an individual workstation's performance.

The 6212 Intelligent Graphics Workstation features two 32-bit applications processors and is intended for concurrently executing highly-interactive tasks (such as editing engineering drawings) and compute-intensive tasks (such as circuit simulation). Separate tasks can run simultaneously on the separate CPUs and on separate windows on the display screen. The 6212 is actually a preconfigured version of the 6210, and a 6210 can be upgraded to a 6212 in the field.

In addition to the two 32032 applications processors, the 6212 is offered standard with a 32016 I/O Processor, a 32081 Floating Point Processor, 4MB RAM and 80MB disk capacity. An additional 16MB of memory, an 80MB Winchester disk, a 40MB cartridge tape, a 9-track tape and 300MB disks can be added to the configuration. 

*UNIX is a trademark of AT&T Bell Laboratories, Inc.

Centronics is a trademark of Centronics Corporation.

MULTIBUS is a trademark of Intel Corporation. VT102, VAX and VMS are trademarks of Digital Equipment Corporation.

Series NS32000 is a trademark of National Semiconductor.

MS-DOS is a trademark of Microsoft, Inc. SDRG GEOMOD and SDRG SUPERTAB are trademarks of Structural Dynamics Research Corporation.

ANVIL-4000 is a trademark of Manufacturing and Consulting Services, Inc.

PATRAN is a trademark of PDA Engineering.

DI-3000 is a trademark of Precision Visuals, Inc.

ANSYS is a trademark of Swanson Analysis Systems, Inc.

Smalltalk is a trademark of Xerox Corporation.

MSC/GRASP is a trademark of MacNeal Schwendler Corporation



The 6200 Series Intelligent Graphics Workstations combine substantial computer power, mass storage and a Local Area Network (LAN) interface, making them suitable for ME design and analysis as well as large-scale ECB, custom VLSI or gate array design. Shown here is the Tektronix 6212 connected to the 6000 Family high performance display, 4115B and 4107 Computer Display Terminal .

6000 Family Display Subsystem Combining Graphics Worlds

“Does it do windows?” is a critical question for many engineering workstation users. A windowed computer environment allows several processing tasks to be monitored on the screen at the same time, with a different screen area or “window” associated with each task. Windows provide a responsive interface between the computing system and the workstation user; however, workstations designed for engineers must also offer sophisticated graphics manipulation, transmission and storage capabilities. For this reason, displays built for the 6000 Family of Intelligent Graphics Workstations offer two different graphics styles – the Bitblt raster technique and GKS-style graphics primitives. Display engine software and hardware is carefully integrated to utilize each style to its greatest advantage.

Bitblt Plus GKS

Put simply, windows are discernible blocks of screen pixels addressed by different segments of computer code. Bitblt Graphics, a technology pioneered by researchers at Xerox PARC for the Smalltalk* programming environment, is a graphics protocol optimized for the type of pixel manipulations used to create windows. With this technology, a display is created by mapping one memory bit (1 or 0) to either on or off illumination of a selected raster pixel.

Bitblt technology offers the display speed and simplicity appropriate for creating menus and icons (graphics representations of command actions) as well as windows. Because these features are key to effective user interaction, the designers of the 6000 Family displays integrated Bitblt graphics into the workstation display system.

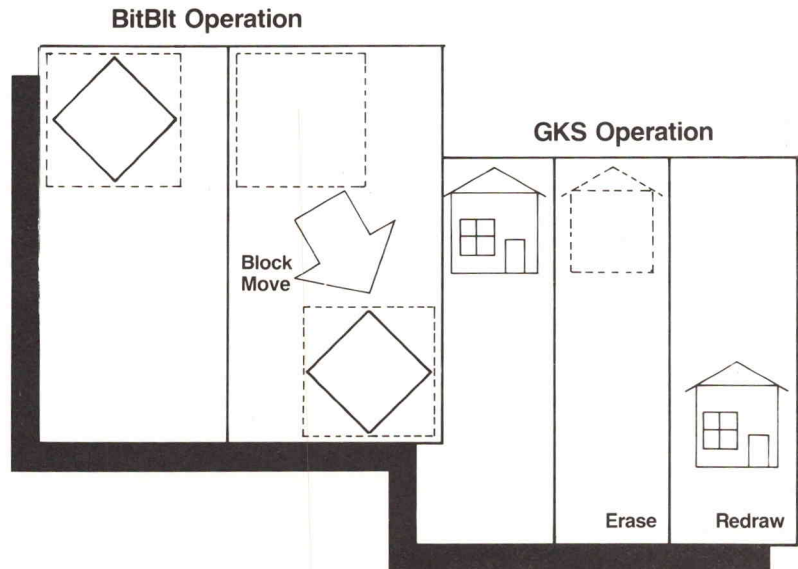


Figure 1. Bitblt graphics operations are performed by addressing individual screen pixels, whereas GKS-style graphics allow the manipulation of entire objects made up of graphics primitives, lines, arcs, circles, etc.

However, a simple pixel-oriented graphics style such as Bitblt is inappropriate for applications that demand a high level of screen resolution and display complexity such as circuit board design, mechanical part drawings, solids modeling, etc. For this reason, the 6000 Family display subsystem also incorporates GKS-style graphics, a higher-level, device-independent protocol formalized by the GKS (Graphical Kernel System) standard.

GKS-style graphics information is manipulated, stored and transmitted as graphics primitives – lines (vectors), circles, polygons, etc. – rather than as pixels (Figure 1). This graphics technique is better suited for translating, scaling and rotating detailed CAD/CAM displays. And, because a display image can be expressed more concisely with graphics primitives than with bit-mapped pixel data, they require a smaller bandwidth when transmitted and less memory when stored.

In addition to Bitblt-style interactivity and GKS-style graphics primitives, the 6000 Family workstation user has the advantage of both regular alphanumeric terminal capabilities and a variety of text display capabilities.

A full spectrum of text output is possible, ranging from simple raster fonts for high performance to proportionally spaced, arbitrarily stroked fonts for special applications.

Software Support for Display Processing

UNIX*, the basis of the 6000 Family's operating system, is noted for its ability to handle many processing tasks and many users at once. However, it's not well-suited for real-time tasks, and the unmodified UNIX kernel is not capable of handling graphics user input in the highly responsive manner that characterizes the 6000 Family user interface.

To remedy this situation – and to allow the workstations' primary central processing unit to remain dedicated to the execution of application programs – user interface tasks are handled primarily by the display subsystem. Also two special-purpose software components, a display subsystem TTY device driver and a display subsystem server, are added to the UNIX-based operating system.

The display subsystem TTY device driver offers a standard ANSI terminal interface to the display subsystem's console/login window. All input and output directed by the

operating system filters through this window. The TTY device driver also offers application programs access to other terminal emulator windows, such as Tektronix 4014 and DEC VT102 emulator windows. With this mechanism, 6000 Family displays are able to offer windows to processing activities taking place on remote workstations or host mainframes.

The display subsystem server handles requests from the display subsystem and from any subroutines accessing display functions. For example, it initiates the display subsystem; initiates the display subsystem TTY device driver; manages process information for the display subsystem; and creates and offers access to windows.

Display System Processes

When discussing applications handled by a UNIX operating system, any group of subroutines dedicated to performing a certain task is called a "process." The 6000 Family's display system software integrates three types of processes (Figure 2):

- The **screen process** maintains control over the entire screen for the display subsystem. The application program requests window management functions from the screen process when creating, moving, burying, deleting or collapsing windows.

- **Window frame processes** exist for each window frame on the screen. These processes display the window frames and handle any I/O to and from the window frames.

- **Window processes** exist for each window or subwindow in a window frame. They handle any I/O to and from the windows.

Another software component within the display subsystem is the **display operating system**. It performs services for the display processes such as providing access to file services from the workstation operating system. Files are accessed for tasks including loading text fonts and nonresident window processes.

Display Lists

Screen output is the result of building and executing display list structures. A display list structure is a simple framework that arranges graphics primitives in order of their correct display. Display subsystem processes - screen processes, window frame processes or window processes - create display lists by retrieving graphics primitives from a general-purpose GKS library as called for by input from an application program.

Each window on the workstation screen can have one or more display list structures associated with it. The display list structure used by the 6000 Family has been designed to

facilitate GKS-style organization of graphics primitives. However, the design also accommodates Bitblt-type graphics operations. In fact, GKS output primitives and Bitblt operations can be mixed in the same display list.

Each display list has a data structure called a "display state" associated with it. The display state maintains a list of environmental parameters within which the display list graphic and text primitives are interpreted. For example, a GKS polyline function has a line type attribute that describes it. The line type information, whether dashed, long dashed, solid, etc., is stored in the display state structure. Similarly, text characters are stored with font type, size, position, orientation, etc.

To understand the flow of a display system request, consider what happens when an application program calls a GKS polyline routine. First, the GKS environment is initialized and a GKS logical workstation is activated. GKS internal state lists and display state structures are then created by the compute engine. To open and activate a GKS window process, a request is made through the display subsystem server. Once this is done, the GKS polyline routine causes the creation of a display list that contains the appropriate coordinate points of the shape to be drawn. When the display list is complete, the display engine executes the instructions (Figure 3).

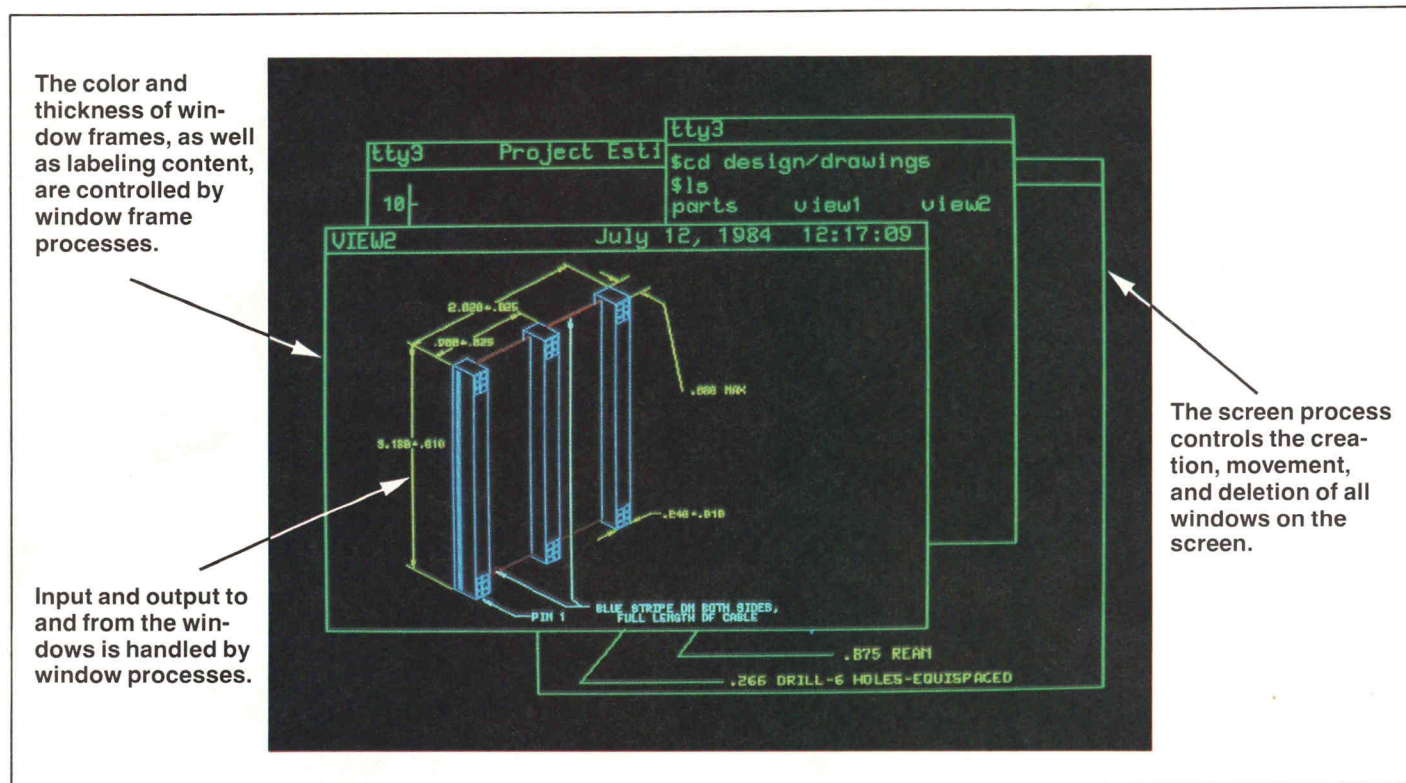
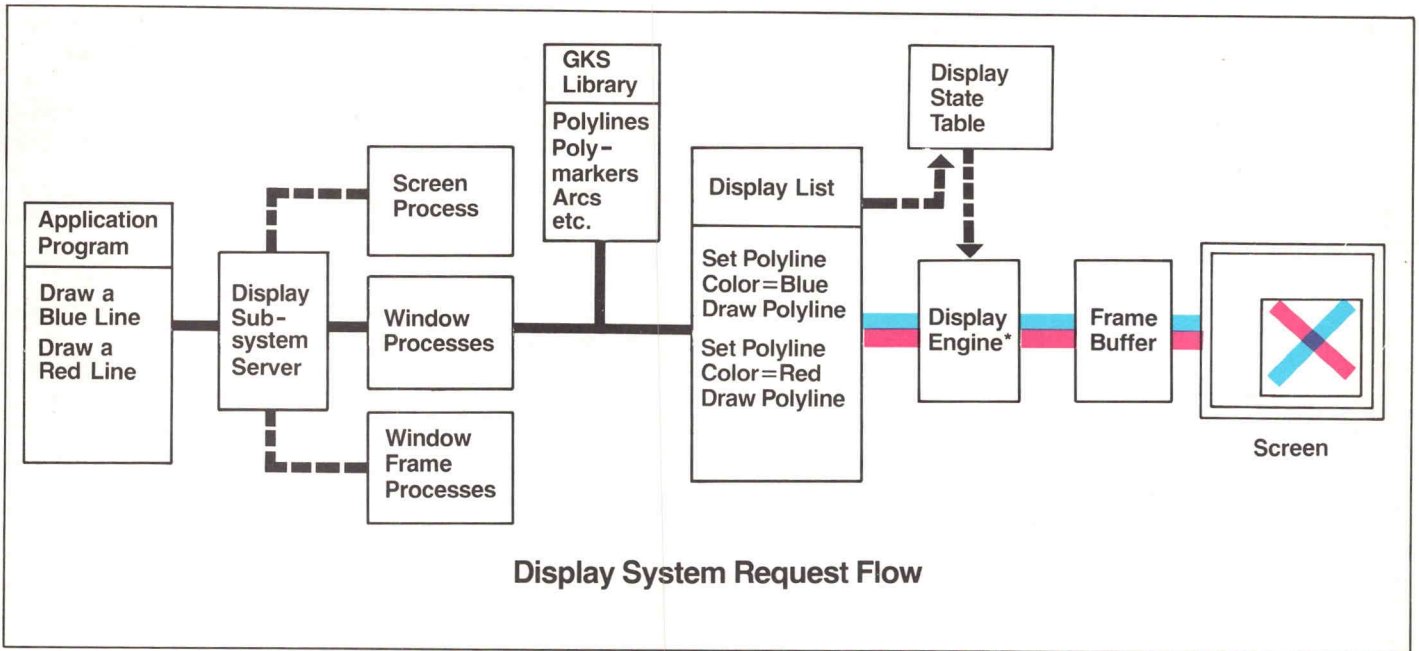


Figure 2. 6000 Family displays perform imaging tasks using three types of operating system processes: screen processes, window frame processes and window processes.



Display System Request Flow

Figure 3. The logical flow of an application program's display system request is the same for 6100 Series and 6200 Series display system architectures. However, for the 6100 Series, the Display Engine that converts display list commands into pixel data is a software-implemented display list processor. The 6200 Series displays perform the same task in hardware with an AM29116 bit-slice microprocessor.

Why Windows?

A windowed computer environment is often compared to a desktop; different stacks of paper related to different tasks can be placed simultaneously on top of the desk. Without window management, the computer system is more like a clipboard; only one piece of paper can be on top of the clipboard at one time. In order to refer to another piece of paper, it must be found and moved to the top of the stack of papers on the clipboard.

An environment that provides both windows and menu selection of commands allows the system to tell the user a great deal about itself. A traditional interface, in contrast, often limits the system's communication to command prompts and error messages. With a windowed interface, a portion of the display area can be devoted to displaying system-related facts that help the user run an application program, while a portion maintains a display of the actual application program. For example, rather than remembering the names of signals that appear on a schematic, a drafter working in a windowed environment can choose from a menu of signal names displayed in another window.

Windows also permit the engineer to better monitor the activities of the system. For instance, a software developer who formerly compiled program modules in the "background" and created a file to capture error messages, can now direct the compiling activity to a separate window and continue on with other system activities, perhaps debugging or coding a separate program module. Compiler messages can then be observed as they are issued. When the window system offers scrolling features, he/she can also scroll back through the compiler messages.

Windows created by the 6000 Family workstation displays act as "virtual terminals," each operating independently of the others. For this reason, our software developer no longer has to use background mode to have separate processes going on at once; he simply opens another window. With the workstations' virtual terminal capability, windows can simultaneously be open to another node on a Local Area Network (LAN) or even to a mainframe host.

Thus, if our drafter's library of signal names resides at a remote workstation along the LAN, he just opens a window

to that workstation's network node and displays the file. The virtual terminal capability also means that one window might contain an applications program with 4014-style graphics manipulations, while another window runs a conventional operating system utility and yet another window runs a text editing applications program.

Windows offer increased flexibility to system developers as well as users. For OEMs, window management provides an opportunity to create a separate and unique environment for application programs that have special requirements. For users, a windowed interface comes another step closer to approximating their own normal thought processes, shortening learning curves and making workstation use more natural and enjoyable.

6100/6200: Display Processing Differences

Both product groups within the 6000 Family share the same logical display subsystem hardware architecture and display system software. The overall operation of the 6100 Series and 6200 Series display engines differ only at a relatively low level – at the point where display list commands are interpreted and sent to the display hardware. The 6200 displays execute repetitive processes using dedicated hardware, whereas the 6100 displays perform the same tasks in software. The difference noticeable by the user is that the 6200 Series displays are capable of faster response and higher resolution.

In the 6100 Series, a major design goal was to keep the display hardware cost low while delivering good performance. Thus, the work of interpreting the display list commands and converting them to pixel data for the frame buffer is performed in software by the 6100 Series display list processor. This enables display circuitry to be contained on one physical board.

The primary workstation CPU communicates with the display subsystem via the display subsystem's common RAM. In this common RAM are the display subsystem software, as well as interprocess communication information and display list data structures, control and status information, and user input. External bus circuitry buffers information and arbitrates between the external bus and the workstation CPU.


Both color and monochrome versions of the 6100 Series displays are available. A frame buffer RAM stores the 640-by-480 pixel bit-mapped display data. The monochrome display uses 64K bytes for one plane of pixel data, whereas the color display uses four times 64K for four planes of pixel data. An index generator circuit enables a monochrome pixel pattern to be converted to color using a color index value prestored by the display processor.

In contrast to the 6100 Series single board display processor, the 6200 Series features a two-board display architecture. Based on a National Semiconductor 32000 Series processor, one compute engine board acts as a dedicated display processor unit (DPU). The second board contains a display list micro-engine based on an Advanced Micro Devices AM29116 microprocessor. This bit-slice processor converts the display list into raster information and then sends the data to either monochrome or color frame buffers.

Because 6200 Series workstations distribute the processing load among general dedicated processing units, multiple display subsystems can be supported by a single workstation CPU. Also, more processing power is available to accommodate the 1024-by-768 pixel resolution of the 6200 Series displays.

Customer's Choice

When workstation display needs involve monitoring a variety of simultaneous tasks or accommodating several users at different

skill levels, the window-managed 6000 Family display is a useful option in a 6100 or 6200 Series workstation configuration. Other display options include the 4000 Family, 4010 Series, 4100 Series and 4110 Series Computer Display Terminals, each offering a price/performance advantage depending upon the application. 

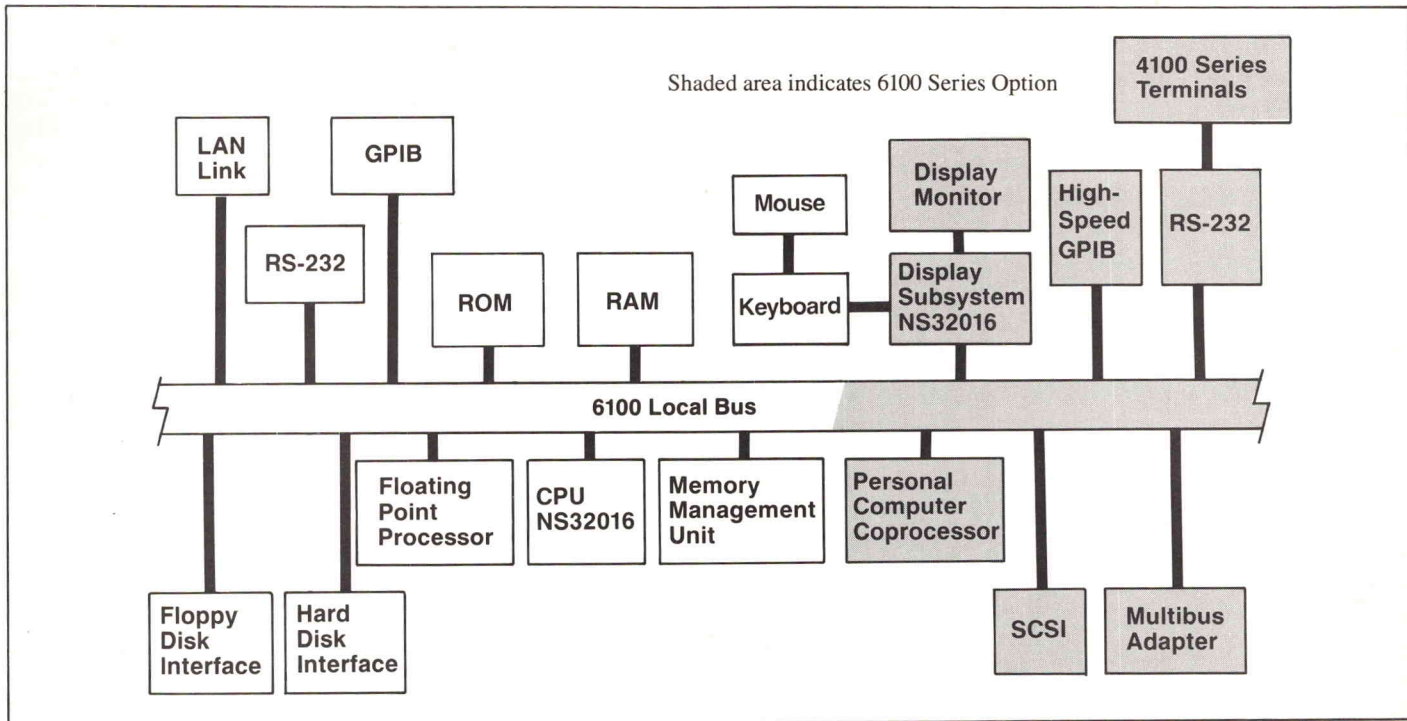


Figure 1. Standard 6100 Series architecture is based upon a single central processor and a local bus.

The 6000 Family Workstation Architecture: Consistent Flexibility

In today's engineering environments, workstations are used for tasks as diverse as designing a product and preparing a manual for that product. While workstation users want the appropriate level of processing power for their particular task, they also want to be able to freely share information with other users and to avoid any extra data input and programming steps.

To meet these goals, the designers of the 6000 Family of Intelligent Graphics Workstations based the two product groups - the 6100 Series and 6200 Series - on different modular system architectures, while establishing strong architectural links between family members. In addition, each member of the 6000 Family offers a UNIX-based operating system (with the exception of the 6110 Instrument Controller which features its own real-time operating system), a relational database manager, a Local Area Network (LAN), high-level language support, standard interfacing options and a wide range of productivity software tools.

NS32000 Series CPUs

The factor that establishes the 6100 and 6200 Series workstations as an integrated family

rather than just a collection of workstations is the common use of National Semiconductor's 32000 Series processors. With their common CPU base, the workstations provide full binary (object-code) compatibility, despite divergent architectures. Thus, a program developed on one workstation can be run on other workstations in the family without modification, as long as the program involves only resources available to both workstations.

The central processor for all 6100 Series workstations and the 6205 is a National Semiconductor NS32016 CPU (Figure 1). National's NS32082 memory management unit and NS32081 floating point processor are present on the processor board to help carry the computational load. Additional processors for functions such as graphics generation are attached to an extension of the 32016's own bus. This standard bus can also be driven by devices and processors not in the NS32000 family. For example, an optional 8086-based PC (personal computer) coprocessor runs in a master/slave relationship with the 32016 CPU to provide compatibility with Microsoft's MS-DOS* operating system. The workstations also support IBM-

formatted flexible disks, enabling files to be exchanged with IBM personal computers.

The Compute Engine board used by 6210 and 6212 workstations contains a National Semiconductor NS32032 CPU (Figure 2). National's 32081 floating point unit, 32082 memory management unit, and other circuitry are also included on the board. For flexibility, the compute engine uses a partitioned design, with plug-in processor and memory module daughter cards. A dedicated I/O processor, based on an NS32016, off-loads the main compute engine CPU by handling all I/O activity. Multiple compute engine subsystems, integrated display subsystems and expansion memory can be added to the modular architecture of the 6200 Series as needed.

Another key factor in creating a workstation architecture that defies obsolescence is a bus that can easily adapt to new devices and new applications. Based on the recently developed IEEE 896 proposed standard, the global bus developed for the 6200 Family provides a high speed data link between the compute engine, the I/O processor and any display subsystems. The 32-bit data bus is designed to extend up to three feet in length and to support up to twenty-eight boards.

To permit interfacing flexibility, the communication protocol is separated from the bus hardware. In fact, the global bus exists in four layers:

- The bottom, "physical" layer provides mechanical and electrical specifications for

the transmission medium and the electrical devices connected to it.

- The next layer, the “transfer” layer implements the arbitration protocols used to obtain control of the bus, and the parallel bus protocols used to transfer data items.
- The “link” layer is imposed on top of the “transfer” layer. It decodes address and control information, routes data to the appropriate location, executes immediate operations and signals events.
- The fourth and highest layer is reserved for specifications from the application.

Communication along the global bus occurs between a master and one or more slaves; and a master-slave relationship only lasts for one transaction. Messages and interrupts exist as a protocol superimposed on the bus’s memory access mechanism.

Addresses on the bus are the addresses of physical resources and are assigned to the subsystems as a function of their position on the backplane. Using an asynchronous protocol, address and data are multiplexed at a bandwidth of 40 megabytes per second. The 32-bit data path supports 8-bit, 16-bit, and 32-bit transfers.

A subsystem connected to the global bus can perform any defined function in any manner as long as an ID ROM is included to present identifying status information to the bus. Also, the subsystem must generate a 32-bit address on the bus and respect global bus resource sharing and message protocols.

In addition to the global bus, each multiple-board subsystem has backplane connections available that it can use to uniquely define its own subsystem local bus. Thus, the compute engine and the display subsystem have their own unique local buses, which are daisy-chained as required between the backplane board slots within a subsystem. For example, the compute engine local bus is used to connect multiple memory expansion boards.

A software mechanism, the Inter-Subsystem Communication Protocol (ISCP), establishes logical channels and communicates requests, responses and control information between subsystems. By providing a uniform level of abstraction to all the resources in the system, it allows the various subsystems to operate in a parallel manner, yet still be very responsive.

The architecture of the NS3200 Series chip also integrates virtual memory, a memory management technique that allows the system to execute programs whose size exceeds the capacity of the available physical memory. A virtual memory system only maps the currently active part of a program’s logical address space to main memory. It

marks the rest of the logical address space as unavailable so that references to it by the program can be detected.

Virtual memory improves system throughput by efficiently using the memory resource. Also, a workstation CPU with virtual memory can handle more tasks concurrently and can run sophisticated multi-megabyte software applications like the large computer-aided engineering programs traditionally run on mainframe computers.

The Demand-Paged Virtual Memory capability built into the NS32000 family includes the critical capabilities of page fault detection and instruction restart. Demand paging is the most efficient method of implementing virtual memory and has been proven in most major mainframe systems from the IBM 370 to the Digital VAX. This technique automatically moves fixed sized pages of program virtual memory in and out of physical memory as they are needed.

The page fault detection mechanism alerts the CPU when a required page is not resident in physical memory. Page faults are always detected while an instruction is being executed, since the effective memory address is not known until that time. When the fault is recognized, the instruction is aborted. Then action is taken to bring the required page into memory.

With instruction restart, when a CPU receives an abort signal from the memory management unit, it backs up to the state it was in before it started the aborted instruction. Instruction restart requires that a certain amount of CPU memory be dedicated to remembering the previous state; but it eliminates the need to save midinstruction state information in main memory. The ability of the CPU to abort an instruction and retry it later, enables it to bring a program into memory, one page at a time. Each time a page fault occurs, another page of physical memory is allocated and loaded from disk memory.

Since demand-paged virtual memory alleviates concern over program and data structure size, source code can be transported across NS32000-based virtual memory systems without special modifications to account for differences in physical memory size. Consequently, virtual memory contributes to code compatibility among members of the 6000 Family.

Operating System and Software Support

The operating system and high-level languages supported by the 6000 Family are industry standards – a factor that simplifies the tasks of software porting and program development (Figure 3). Also, because operating system and software support is consis-

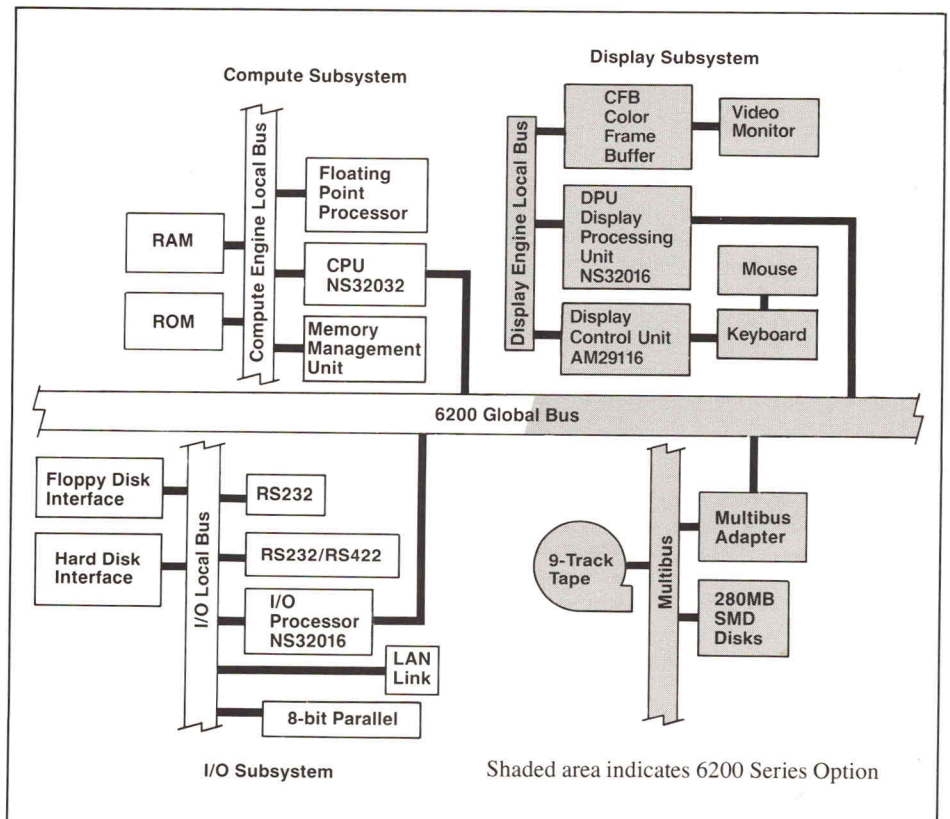


Figure 2. Standard 6200 Series architecture includes an NS32032 central processor, an NS32016 I/O processor and a 32-bit global bus.

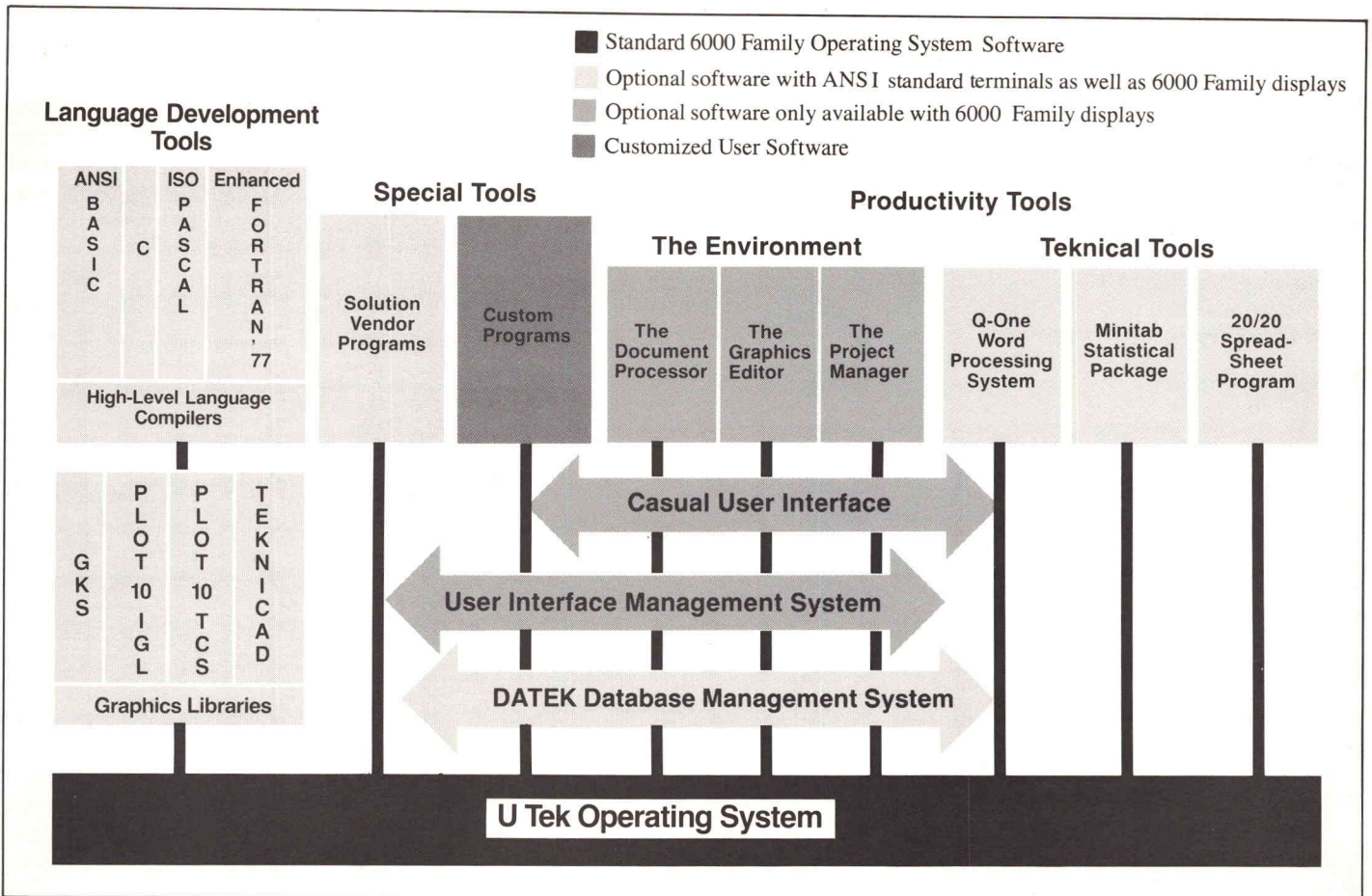


Figure 3. 6000 Family modular software architecture offers tools for program development, database management, user interface management and personal productivity, as well as facilities for integrating custom-developed and outside-vendor applications programs.

tent throughout the 6000 Family, users can apply their knowledge of system commands and protocols to any member workstation.

UTek, the operating system common to all 6000 Family workstations (except the 6110 Instrument Controller), is an enhanced version of Berkeley 4.2bsd UNIX. An additional library provides a high degree of System V source-code compatibility. UNIX, an operating system originally designed for software development, is the logical choice for workstation environments for several reasons:

- Interactivity – A response is initiated as soon as a command is issued.
- Multitasking – Workstation users can initiate new processing tasks without waiting for current tasks to be completed.
- Multiusers – More than one user can be supported simultaneously, which promotes information sharing between workstations within a UNIX-based network.

The architecture of the NS32000 CPU Series is patterned after Digital Equipment Corporation's VAX mainframe technology – the development standard for Berkeley UNIX. For this reason, only minor modifications to

the UNIX kernel were needed to port the operating system to the 6000 Family. Also, because addressing modes, register usage and byte order are similar to the VAX standard, applications programs developed on other systems are easily transported to the workstations.

Along with its reputation for power and efficiency, the UNIX operating system is recognized as a more difficult operating system for new or occasional users to master. To aid the 6000 Family user, operating system tutorials and "help" information can be called on-screen at any time. On-line documentation includes "manual pages" that offer thorough descriptions and examples for each operating system command.

To make information contained in the manual pages more accessible, the system responds to a variety of specific user requests. For example, it will scan manual pages for all references to a command's syntax or flags when only that information is desired. Also, online explanations of system error messages are always available.

To further aid users less familiar with UNIX, an optional Casual User Interface (CUI) pro-

vides an alternate interface to the operating system. Taking advantage of the 6000 Family's integrated display systems, the CUI enables users to operate the workstation by manipulating a mouse input device to select menu items. Users with the CUI option can easily utilize workstation application programs without a thorough understanding of operating system command names and syntax. Even the tasks of system and network administration are supported by interactive menus, simplifying activities such as adding new nodes to a network, maintaining the file system and creating user accounts.

Another 6000 Family software option that helps to unify a workstation network is DaTek*, a relational database management system. Local database management promotes information sharing by making data generated by any workstation user available to all workstation users within a network.

The relational database model is a significant improvement over earlier hierarchical methods because it offers greater flexibility in the use of data. Data is stored in independent files rather than locking it into specific programs. DaTek enables data sharing between network users despite differences in

languages, applications and procedures. Access to a common database assures each user that his/her information is current, accurate and complete.

Many DaTek functions as well as standard UNIX communications services are carried out by the 6000 Family Local Area Network (LAN). The LAN promotes information sharing by enabling users to log onto a remote workstation and to transfer files between workstations and to and from VAX computers running compatible versions of UNIX (Figure 4).

Based on the Ethernet IEEE 802.3 standard, the LAN supported by all 6000 Family members features a Distributed File System (DFS) that makes network functions transparent to the user. The DFS protocol consists of three layers of software:

- The top layer, the interface to the DFS, is standard 4.2bsd UNIX socket layer code.
- The middle "transport" layer features a message-based protocol.
- The bottom "link" layer processes the Ethernet packets.

The DFS minimizes the number of operating system calls needed to perform standard UNIX communications functions and makes it unnecessary to modify or recompile any programs that use the communications facilities before porting them to a 6000 Family workstation.

Though several different organizations within a company share the same Ethernet network, they may not want to share the same Distributed File System. The 6000 Family allows multiple and distinct distributed file systems to coexist on the same LAN by establishing each one as a domain. When several workstations on a LAN are divided into separate DFS domains, traditional UNIX remote copy utilities will move files between them.

The 6000 Family LAN further simplifies network administration via three special-purpose utilities:

- The Nameserver utility automatically maintains an up-to-date list of the names of all machines on the network so that a workstation can be designated by name rather than by its physical address on the network.

A language translation utility is available to assist conversion of Tektronix 4050 Series Desktop Computer BASIC programs to the 6000 Family workstations.

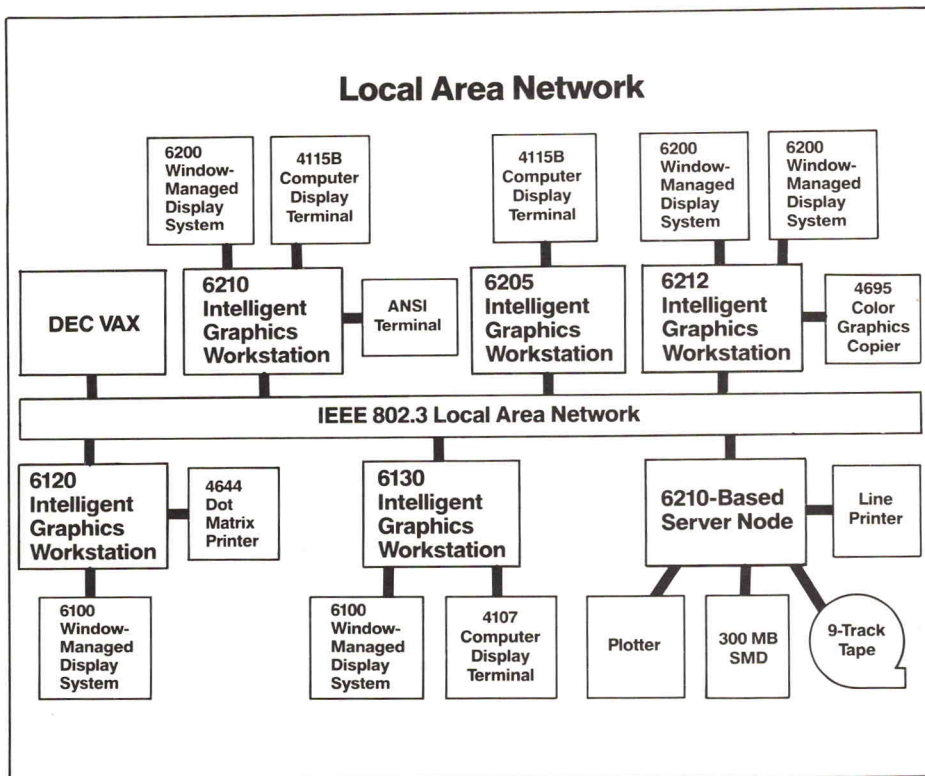


Figure 4. Communication between 6100 Series workstations, 6200 Series workstations and host computers is conducted over an Ethernet IEEE 802.3 Standard Local Area Network using TCP/IP Protocol.

- "Nettest" verifies that the network is usable by performing several tests, including echoing data between workstations and analyzing the Ethernet coaxial cable.
- "Netconfig" automates workstation installation steps, enabling a workstation to be added to the network by simply assigning it a unique name and network number.

Each member of the 6000 Family supports software written in high-level languages including the proposed ANSI BASIC, enhanced FORTRAN 77, ISO Pascal and C languages. The designers of the 6000 Family anticipated that the majority of code executed on the workstations would be compiler-generated. For this reason, the creation of efficient high-level language compilers was a primary design consideration.

An architectural feature of the 6200 Series NS32032 CPU that promotes efficient high-level execution is its set of general and dedicated registers. The eight homogeneous general registers easily accommodate high-level language code because they all have the same structure and they operate identically with all instructions. The eight dedicated registers incorporate special code-generation capabilities. For example, high-level languages typically have a stack which is composed of multiple activation frames for dynamic storage allocation. NS32032 registers are specifically allocated as either stack registers, frame pointer registers, or

static base registers.

The NS32032 instruction set provides a rich set of addressing modes, and many high-level constructs are directly executable. The symmetry of the instruction set allows most instructions to employ all addressing modes and operate on all data types. The NS32032 directly supports array addressing in high-level languages with upper and lower array bounds-check instructions and an array index-calculation instruction for multidimensional arrays. The scaled index addressing mode allows quick reference to single dimension arrays or complex structures.

Reference to packed data can be one of the most difficult tasks of a compiler writer, and the resulting code is generally inefficient because it must be done with individual shift and mask instructions. The NS32032 provides field instructions that allow direct access to packed data formats. String operations are directly supported by the instruction set, as well as bit operations that enable accessing variable length fields anywhere in memory without byte boundary and alignment restrictions. The operating system makes heavy use of these features.


Interfacing Flexibility

To accommodate expanding memory and storage needs and to simplify the integration of a variety of peripherals and other equipment, 6000 Family workstations provide a

number of standard interface ports. The 6100 Series supports two RS-232-C interfaces and one General Purpose Interface Bus (GPIB IEEE-488). A Small Computer System Interface (SCSI), a high-speed GPIB, an eight-bit Centronics-compatible port and a Multibus adapter are additional interface options. The 6200 Series provides RS232/RS422 support as well as RS-232-C interface ports. In the future, the 6200 Series will support a high-speed serial, 24-bit General Purpose Interface Bus (GPIB) and an external Small Computer System Interface (SCSI) to speed I/O transactions.

A variety of printer and graphics output devices can be included in a 6000 Family network via the RS-232, Centronics-compatible and Multibus ports. For workstation output these include the Tektronix 4644 Dot Matrix Printer (which is capable of high-density dot-addressable graphics) and the Tektronix 4695 Color Graphics Copier. Device driver support for high-speed line printers (300-600 lpm) and letter-quality printers is also provided. The 6200 Series also has optional drivers and interfacing to support large format output devices such as E-size electrostatic printer/plotters.

Terminal emulation is available as well, enabling the workstation to function as a terminal to a host computer. The 6000 Family display system can emulate the DEC VT-52* terminal and the Tektronix 4014 with 4105 color commands. So, for example, host-based software that interacts with a Tektronix 4014 Computer Display Terminal can easily be ported to the 6100 and 6200 Series workstations and interact with the 6000 Family display system.

From the selection of a central processor to the design of an external bus, through the choice of supported languages and interface standards, 6000 Family developers wanted to create a system architecture with longevity. The resulting systems should easily adapt to workstation network needs including new users, diversified instrumentation and changing communication protocols. 

Workstation Software Tools

A number of tools are available to help workstation users improve productivity and build upon the 6000 Family's open software architecture.

"The Environment," a set of sophisticated productivity tools designed for use with 6000 Family display systems, includes the Casual User Interface (CUI) and the following software packages:

- The Document Processor – a sophisticated and easy-to-use word processing system designed to handle the complex documentation needs of engineers and scientists. The Document Processor is a modeless, "what you see is what you get" editor that simplifies the job of creating documents ranging from brief memos to specifications that include a table of contents, index, section headers, and super- and subscripts.
- The Graphics Editor – an interactive, visually-oriented tool for creating conceptual sketches, technical illustrations and presentation graphics, which can then be incorporated into a document created with the Document Processor. The Graphics Editor can also be used to post-edit pictures created in another environment.
- The Project Manager – a set of interactive, graphically-oriented tools for controlling technical project resources, costs and schedules. It uses familiar concepts of project management and a simple method of data entry, allowing a manager to quickly perform activities such as producing Gantt and PERT charts, calculating a project's critical path, and producing a variable calendar.

The user's interaction techniques are the same for all software within THE ENVIRONMENT, so the learning time spent on any one of the tools applies to the others as well.

Other 6000 Family productivity tools can be used with ANSI terminals, as well as with the 6000 Family integrated window-managed displays:

- Q-ONE* – a word processing system with an easy-to-use interface and powerful formatting capabilities.
- Minitab* – a general-purpose statistics package with built-in table-formatting and FORTRAN-formatted input and output.
- 20/20* – a spreadsheet program that combines graphics and data management capabilities to produce an integrated modeling environment. A spreadsheet can be up to 1000 x 1000 cells.

Q-One, Minitab and 20/20 are available as a package known as "Teknical Tools." This package includes additional electronic mail and calculator utilities, as well.

In addition, workstation facilities such as the Datek database management system, the user interface management system (See related story) and the Casual User Interface (CUI) can be integrated with applications software supplied by Tektronix Solution Vendors, as well as programs developed by the customer.

In our next issue of *TEKniques* the Datek database management system and other 6000 Family Software Tools will be discussed in greater depth.

A User Interface Management System For Workstation Productivity

David Schiferl
Software Engineer
Tektronix, ECS
Wilsonville, OR

Managing user interaction has posed a challenge to software developers since the advent of on-line computer processing. User interface functions such as presenting information to the user, obtaining input, offering feedback and supplying error messages are common to almost all interactive application programs. The need to alleviate the burden of constructing a user interface from scratch for every application program led to the concept of a User Interface Management System (UIMS) – a set of tools that aid in the design and implementation of interactive user interfaces.

The evolution of the UIMS can be compared to that of database management systems (DBMSs). Before DBMSs became common, all database application programs included a portion of code devoted to managing low-level storage details. The independently-developed databases were inconsistent and often inefficient. The DBMS was introduced to free the application programmer from low-level programming tasks and to enforce a consistency in data manipulation and storage.

Similar goals guided the development of the UIMS. These goals included:

- Simplifying the design, prototyping and implementation of user interfaces.
- Providing consistent interaction techniques across applications and within applications, allowing user interface standards to be established and promoted in a product, within a company or throughout an industry.
- Inviting the development of user interfaces by interaction designers – specialists trained in disciplines such as ergonomics or human psychology whose main concerns are the user's ease of learning and ease of use, rather than the computer system's efficiency.

Like the DBMS, the UIMS handles low-level input and output functions and certain other implementation details for the application programmer. Moreover, it separates the tasks of developing application programs and developing user interfaces. The UIMS allows application programmers to concentrate their efforts on application needs, and offers the user interface itself as a new focus for testing and improvement.

UIMS Structure

The UIMS developed for the 6000 Family of Intelligent Graphics Workstations includes two components – a Tool used to specify the user interface and a Run-Time System that executes the specifications. Together the components exist as a framework within which rules for communication between the application user and the application can be established.

Interfacing with UTek, the workstations' UNIX-based operating system, the UIMS filters all interaction between applications and the graphics system. The development of a variety of user interfaces and a number of diverse applications to UNIX is supported, including a graphics editor, a document processor, a project management system and a front-end window-oriented system.

The UIMS Tool

Tektronix' UIMS Tool utilizes graphical interaction techniques that shorten the interaction designer's learning curve and ease the user interface specification process. In most instances, applying this Tool is a simple matter of responding to menus and tailoring displayed templates to meet application needs.

The Tool is essentially comprised of two editors – a graphical editor or "Screen Formatter" and a control-flow editor called the "Dialog Definer." The Screen Formatter is

used to partition the display into screen areas and to specify the contents of those areas. Once screen areas have been established, the Dialog Definer is used to assign user interface and application functions to each area.

Defining Screen Areas

Screen areas are the interaction designer's basic building blocks when constructing a user interface. They can be sized, positioned, named and assigned attributes as individual units. The attribute defaults assigned to each screen area type by the UIMS promote easy recognition and reaction on the part of the user. However, if the application calls for some adjustment to these defaults, the interaction designer is always free to independently specify attributes such as the interior color and fill pattern of areas and the color, intensity and thickness of area borders.

Seven screen-area types can be defined with the Screen Formatter:

- Menus** are lists of user-selectable items that can be chosen by pointing with the workstation's mouse cursor. A number of menu types are available to the interaction designer: those that display menu choices one item at a time to minimize the amount of screen space needed, and those that present all available choices at once. Available highlighting techniques include varying the color of border or background of a menu item, or marking an item with a bullet, tick mark or asterisk.

The interaction designer can also specify "pop-up" menus. These menus appear when the application user presses a specified mouse key. As the cursor is moved, the current item under the cursor is highlighted. When the key is released, the item under the mouse cursor is selected and the menu disappears. If the mouse cursor is outside the pop-up menu when the key is released, the UIMS interprets the user's selection as null.

If the designer wants to simultaneously show the user an assortment of menus, a "property-list" menu area is defined. All menus included in a property-list will appear and disappear together. The user responds to each menu individually, and then enters all choices by selecting an ACCEPT icon. A CANCEL icon is available to nullify all current property-list menu selections.

- Scroll Areas** are designed to selectively display quantities of information. The user controls the scrolling process by mouse interaction within a scroll bar area. The scroll bar is tailored by the interaction designer to meet applications needs. Smooth scrolling, incremental scrolling, horizontal and vertical scrolling, and in

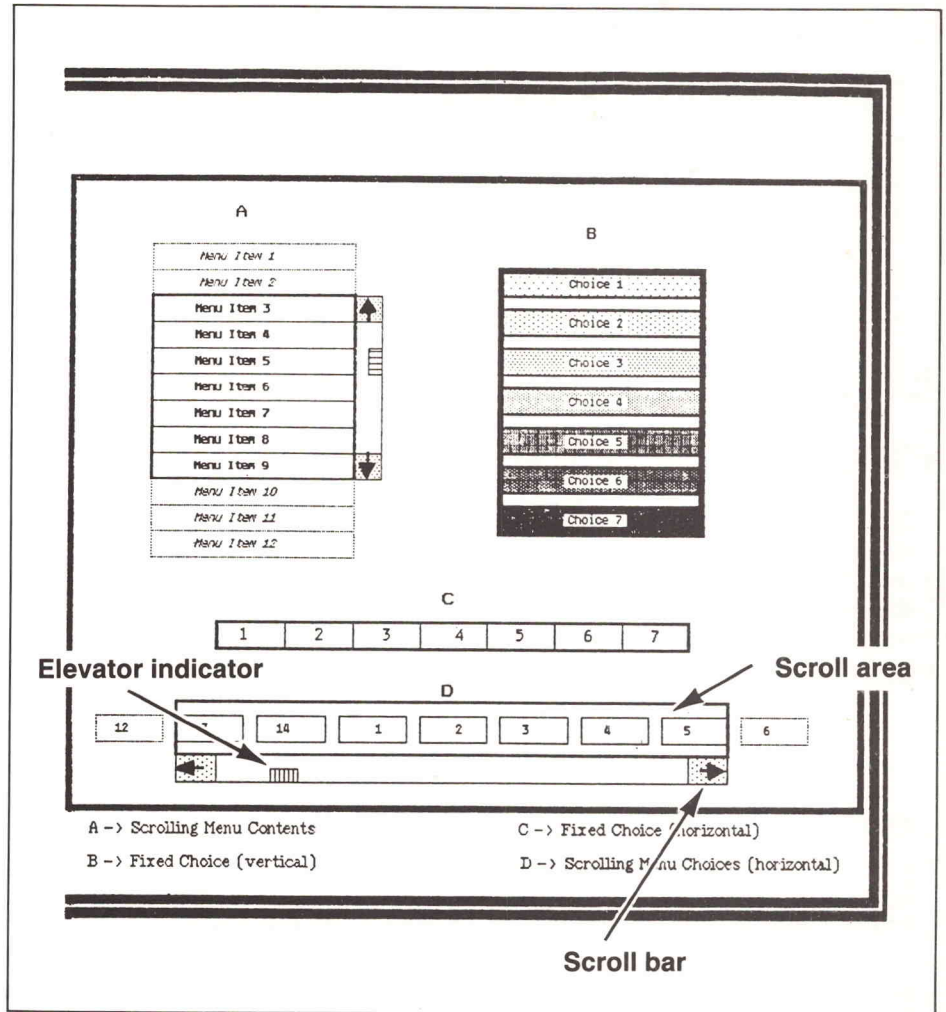


Figure 1. UIMS menus may present all menu items to the user at once or may selectively present items within a scroll area.

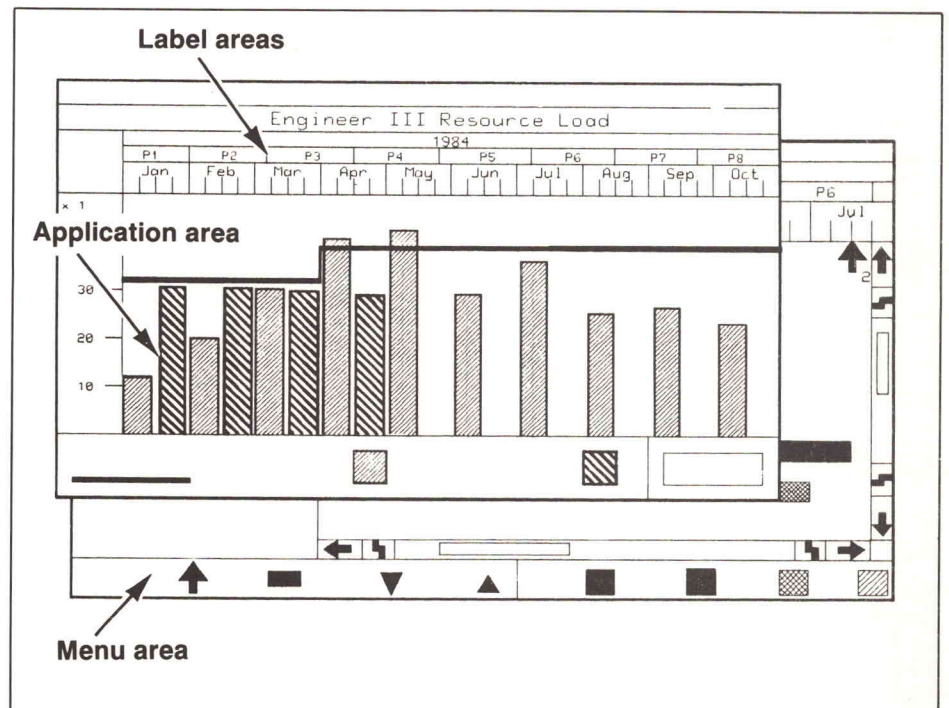


Figure 2. A screen defined with the UIMS Screen Formatter for a project management application establishes consistent and comprehensive labeling procedures.

and out zooming are all supported by the UIMS. The scroll bar features an "elevator" indicator that graphically represents the total size of the file being scrolled and the position within that file held by the information currently appearing on the screen (Figure 1).

- **Prompt-Command Areas** are available for standardizing the placement and style of textual input. Prompts detail the type of input sought and how to supply it. If the user's response is through the keyboard, it is echoed in the command section of the area. A Prompt-Command area can be created to contain both a prompt and command section or either type exclusively.
- **Message Areas** are created to contain information concerning the progress of the application and to display warnings and other general notes. Message areas may contain icons and/or text. Standardization of the border and interior colors of these areas greatly aids recognition by the user.
- **Error Message Areas** are used to display error and exception messages exclusively. Standard border and interior colors can also be assigned to these areas.
- **Label Areas** are used to display the name or title of an area or group of areas. Distinctive border and interior colors can also be specified for each label area.
- **Application Areas** are areas devoted to application-dependent data and are controlled by the application program. For example, when formatting a screen for a project management application program, an application area would be designated for displaying cost analysis information; when formatting a screen for a word processing application program, an application area would be created to contain text. The appearance and functionality of application areas are determined by the application developer, rather than predefined by the interaction designer.

- **Group Areas** link a number of previously defined areas into a set that can be manipulated as a unit. Areas defined as a group can be displayed, positioned and cleared from the screen together. Any one of the defined area types can be contained in a group area.

Dialog Definition

The primary function of the second Tool editor, the Dialog Definer, is to smoothly integrate the user interface with the application program. The process of dialog definition involves defining valid user input for each screen area specified with the Screen Formatter.

Mouse key interactions keyboard entries and mouse movement into and out of an area are among the types of input recognized by the UIMS. Desired feedback, echoing, highlighting and other interface components for each type of physical user input are defined by the interaction designer. Once these specifications are provided, an application routine is specified to carry out the application-dependent tasks called for by the various types of input. In this way, the UIMS is "taught" how to acknowledge user input and what information to present next.

When an application is running, the UIMS gathers all user input. Next it determines which area the input occurred in, and executes the appropriate interactions for that input in that area. Finally, it calls the application-dependent routine that handles the application-dependent tasks associated with the received input.

The interaction designer can also specify how the UIMS should respond to unrequested user input. With proper instruction, the UIMS can relate the currently executing routine to the latest user input to determine the correct course of action. It may send an error message or, if the intended input is predictable, it may call the appropriate

application-dependent routine to handle the input.


UIMS Run-Time System

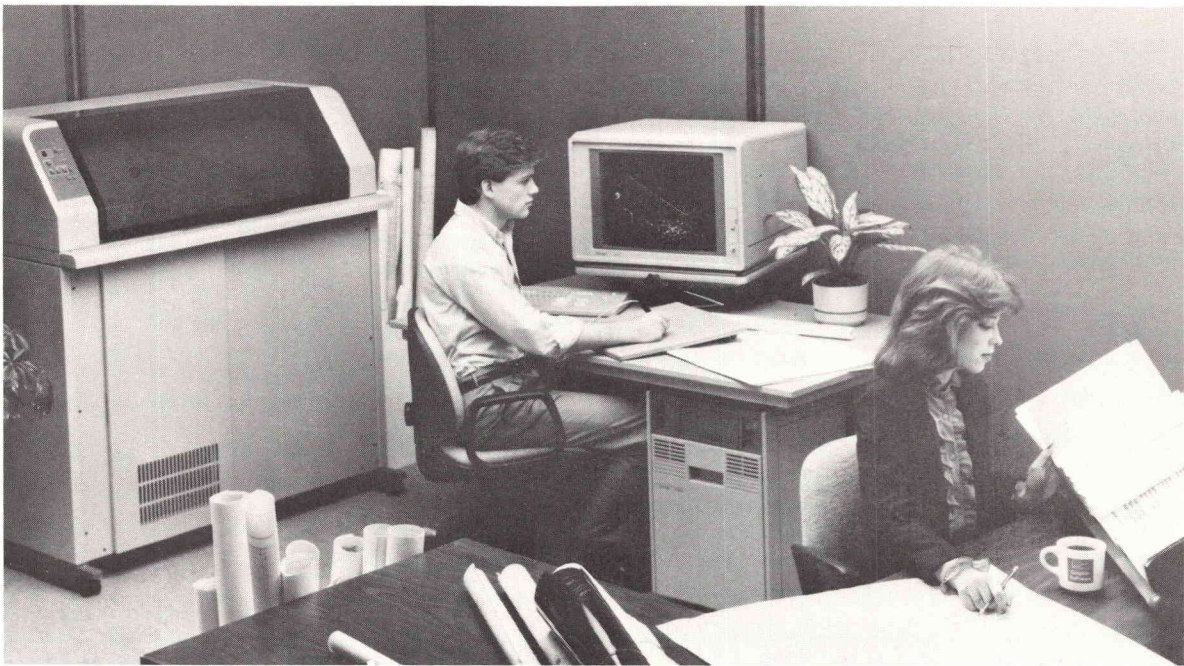
The UIMS Tool provides the means for specifying the user interface. The second component of the UIMS, the Run-Time System, executes the application and its specified user interface.

The Run-Time System provides a library of routines that can be used to dynamically alter the user interface. These routines provide the mechanism for manipulating screen areas and user interface routine dialog. Library routines can be invoked to cause an area to appear or disappear and to set area attributes, such as border color and visibility. Inquiry routines offer system status information at any time during application program execution.

Included in the library are routines which facilitate application output and the scrolling of application areas; menu area routines to alter the appearance and behavior of menus; and message area routines to output specific message text and, in the case of Prompt/Command areas, the input of specific commands.

Conclusion

The UIMS is an ideal component in a computer-aided workstation environment. It opens several doors for a range of users. Those developing their own application programs are freed from the burden of creating a user interface from scratch. The interface can now be created by an interaction specialist, knowledgeable in areas of human factor engineering. And, those looking to applications programs for productivity gains are provided with tailor-made user interfaces consistent across applications. 



The 4991S1 Graphics Input Workstation automatically transforms hardcopy drawings into CAD databases for storage and future CAD system processing.

Graphics Input Workstation Systemizes Hardcopy-To-Database Conversion

Even in today's most sophisticated CAD environments, Tektronix market studies show that only 20 to 30 percent of the drawings created are computer-generated. Typically this is due to a backlog of design information that must be entered into a database, such as drawing masters archived on paper or film and hardcopy part drawings provided by component and sub-assembly suppliers. By addressing this data capture challenge, the recently introduced 4991S1 Graphics Input Workstation increases CAD system productivity in two respects:

- Vast amounts of graphics information can be input in a minimum amount of time and with a high degree of accuracy.
- Existing computer-aided design/engineering systems can be put to greater use when more complete data base files are available.

The graphics input process is automated – from the scanning of hardcopy documents to the transmission of structured graphics data to host mainframes for storage in CAD databases.

4991S1 Components

The 4991S1 Graphics Input Workstation is composed of a Tektronix 4991 Autovector-

izer, Graphics Structuring Software, Host Interfacing Software, an M4115B Computer Display Terminal and a 4957 Graphics Tablet. The 4991 Autovectorizer automatically scans and vectorizes hardcopy drawings. Graphic Structuring Software converts the vectorized data into high-level graphics primitives. And Host Interfacing Software directly transmits the structured drawing data to the host for CAD database storage and future CAD system processing.

All graphic structuring and drawing enhancement steps are performed locally on an M4115B Computer Graphics Terminal. Complete with a 4957 Graphics Tablet that eases user interaction, the 4991S1 Graphics Input Workstation is a total system solution to graphics input.

Autovectorizing

The 4991 Autovectorizer automates the basic inputting task, providing the user with a preliminary vectorized drawing, ready for review and further manipulation. Based on a proprietary algorithm developed at the Tektronix Applied Research Laboratory, the 4991 Autovectorizer performs raster-to-vector conversion in realtime. Vector lines are encoded as the drawing is scanned. Memory costs are significantly lower than

vectorizing methods which require that raster data be stored until scanning is complete.

The Autovectorizer accepts drawings submitted on paper, Mylar* or other media in sizes of up to 35-by-47 inches (E size). Image parameter defaults are programmed to provide optimal image quality, reducing the set-up procedure to a simple matter of identifying drawing quality, size, input media and desired output protocol. When necessary, threshold adjustments can be made for difficult-to-scan documents and media with different contrast ratios. Custom parameter sets can be stored for later use.

The 4991S1 Graphics Input Workstation offers productivity gains estimated by Tektronix market research to be up to ten times over manual input methods such as tracing drawings with a digitizing tablet or on-screen redesign. Manual methods also risk introducing new errors in existing designs.

Graphics Structuring

Once vectorized, a drawing file is available for graphics "structuring" – a process that translates vector line coordinate data into higher-level graphics primitives, such as lines, arcs, text and symbols. Graphics primitives require less storage space and can be

transmitted to and from the host more efficiently than vectors. During the graphics structuring phase, the user selects graphic manipulation and structuring functions from comprehensive on-screen menus. Structuring functions include line straightening; n-point registration; and semi-automatic arc, line, text and symbol replacement.

Undesirable drawing features caused by smudges, creases and stains on the hardcopy are easily eliminated at this time. Also, the user can distinguish specific portions of the drawing using color tagging (a programmed function that assigns a distinctive color attribute to an indicated item) or by adding alphanumeric annotation.


Graphics structuring may also include dividing drawings into "layers" for separate

storage, display and manipulation. For example, a mechanical drawing might be stored with separate layers for text annotation, the part outline and dimensional information.

Graphics Data Transmission

In addition to improving the clarity and content of the drawing file, the local graphics structuring step allows the 4991S1 Graphics Input Workstation to take maximum advantage of existing data links. Structured drawing files are sent directly to RS232-compatible host computers in a data format recognized by three leading CAD software systems - IBM's CADAM*, Computervision's CADD5 4X* and Tektronix PLOT 10 Computer-Aided Drafting software (TekniCAD). All host communications take place in background mode, freeing the

M4115 for further processing. In addition, because all scanning, vectorizing and structuring functions are performed locally, host processing power is reserved for higher-level CAD database manipulations.

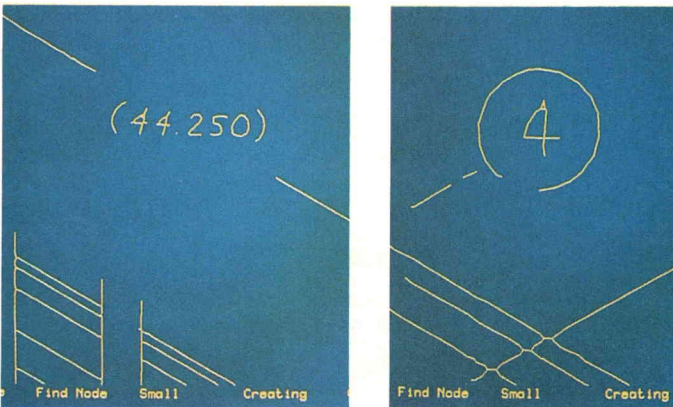
The system will be available in the United States in January 1985, and internationally in June, 1985. 

*Mylar is a registered trademark of E.I. duPont de Nemours and Co., Inc.

CADAM is a registered trademark of CADAM, Inc.

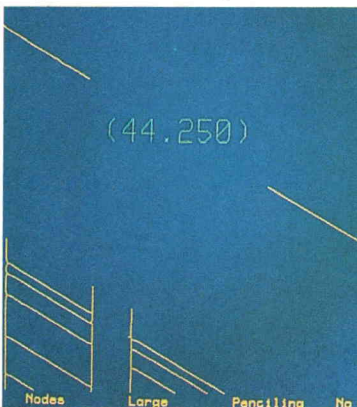
CADD5 4X is a registered trademark of Computervision.

Before Graphic Structuring

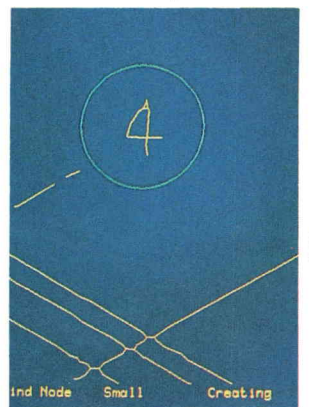


After Graphic Structuring

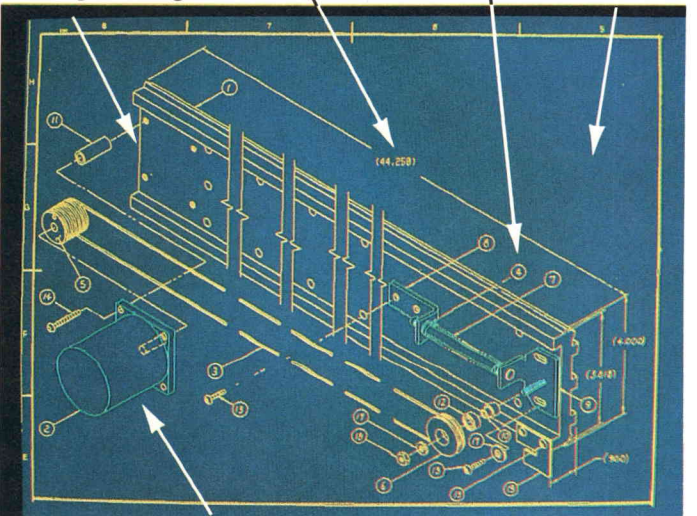
Alphanumeric replacement



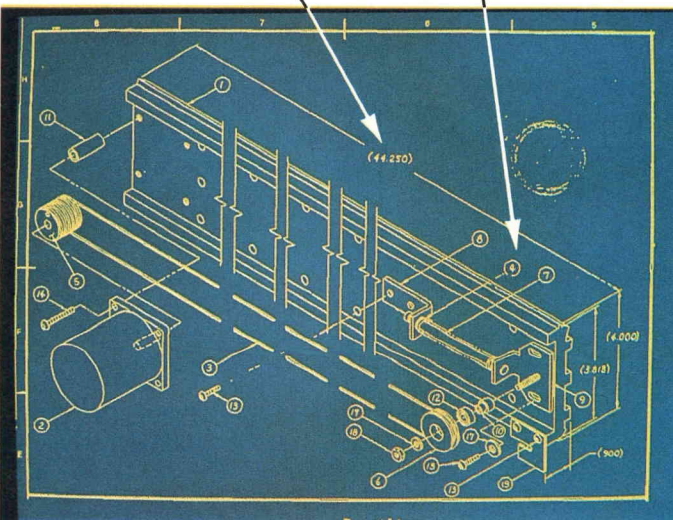
Arc/circle replacement



Line Straightening

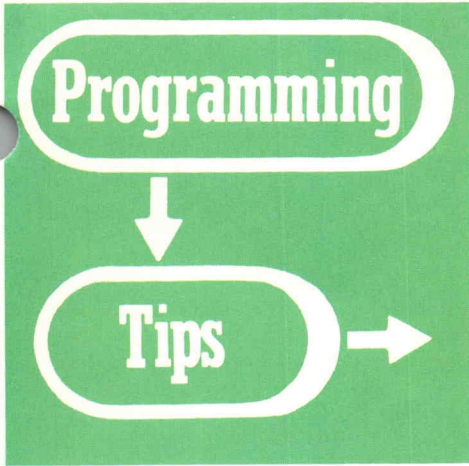


Cleanup



Object Color Tagging

Once a drawing is scanned and vectorized by the 4991 Autovectorizer, a line segment representation is available for viewing on the M4115B Computer Display Terminal. Using Graphics Structuring Software, the user can convert this drawing file into a database of objects and attributes. Vectors, arcs, lines, text and symbols can be replaced with higher-level graphics primitives to improve drawing clarity, as well as data transmission and storage efficiency. Also, drawing features can be registered to a specific N-point; objects can be "tagged" with a distinctive color; and undesirable drawing features can be eliminated.



4110B LoPro Date/Time Access

by Stan Jensen
Tektronix, Inc.
Santa Clara, CA

Date and time values are maintained by the CP/M-86 BDOS in the "data page" area. These values are initialized with the TOD command (or the ITS utility on the Program Exchange disk). To access this clock function from your 4110B LoPro program you can use a routine which is part of the Universal System Development Interface (UDI), rather than writing a program in assembly language. The routine is called DQGETTIME and is in LARGE.LIB on the FORTRAN Volume 2 disk. The sample FORTRAN program below shows how to call the routine and receive the data.

Notes:

- The DQGETTIME routine requires an INTEGER*1 array. The sample subroutine uses the FORTRAN INT2 func-

tion to convert the data to the default INTEGER*2 storage type.

- The data is returned as ACSII decimal equivalents (ADEs). You may need to convert them to numbers (by subtracting 48) to perform time calculations.
- The accuracy of the clock function can't reasonably be specified, especially in the multitasking environment of a 4110B terminal.
- This technique will not work with Release 1.1 of the 4170 operating system because of the absence of the TOD command.

For more information on the date/time fields, consult the "CP/M-86 System Guide," Application Note 01 (located just inside the back cover of the manual). For more information on the UDI, read Appendix I in the "FORTRAN-86 User's Guide."

```

INTEGER JARRAY(16)*2
CALL TOD(JARRAY)
WRITE(*,100) JARRAY
100 FORMAT('DATE/TIME = ',16A1)
END
SUBROUTINE TOD(JARRAY)
INTEGER I, IARRAY(16)*1, JARRAY(16)*2, KERROR
CALL DQGETTIME(IARRAY, KERROR)
DO 100 I=1,16
100 JARRAY(I)=INT2(IARRAY(I))
RETURN
END

```

Joypop to the Top/Bottom of the 410X Dialog Area Buffer

by Barry Lewine
Tektronix, Inc.
Wilsonville, OR

A number of users have requested that joydisk-controlled scrolling in the 410X dialog area be enhanced to allow popping to the top or bottom of the buffer the same way a quick flip of the vertical thumbwheel works on a 4110 product.

That feature is already accessible (thanks to the X3.64 command set) and requires only that two short macros be defined.

From Tektronix mode (CODE TEK):

```
Define - 142 /( P)(ESC)(%)(!)(1)(ESC)(I)
(5)(0)(0)(T)(ESC)(%)(!)(0)( P)/
```

```
Define - 140 /( P)(ESC)(L)(V)(1)( P)/
```

Or from ANSI mode (CODE ANSI):

```
Define - 142 /( P)(ESC)(I)(5)(0)(0)(T)( P)/
```


```
Define - 140 /( P)(ESC)(%)(!)(ESC)(L)
(V)(1)(ESC)(%)(!)(1)( P)/
```

Explanation: The process uses one native Tektronix and one ANSI command. Note that the primary difference in the macro pairs above is that the functional portion of the definitions are delimited with the appropriate SELECT CODE command [(ESC)(%)(!) parameter] to assure that the proper parser is always in place. Note also that these commands will not work in SETUP mode and that the first and last characters of each macro must match the current key execute character ((P) in this example).

These macros define the shifted joydisk-down key (- 142) to issue the ANSI <<scroll-lines-down>> command. (The example uses 500 for number of lines.) This will move the

top of the buffer into view.

Similarly, there is an ANSI <<scroll-lines-up>> command, but using it would move the specified location of the declared buffer into view, rather than moving the bottom portion of the used buffer into view. (E.g. DAB = 500 w/200 lines used . . . A move to the bottom of the buffer would result in a blank screen with many blank lines above it!)

No problem, though. Simply issuing the TEK command <<dialog-visibility-on>> has the desired effect. The above definitions associate that command with the shifted joydisk-up key (- 140). 

These commands are documented in all 410X Host Programmer's Reference manuals.

IDG Program Exchange

In November, 1984, the IDG Program Exchange formally ceased operation.

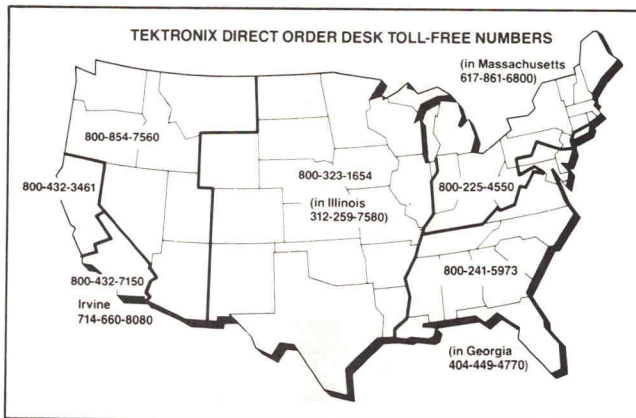
The exchange was initially established in 1976 as the 4051 Applications Library. It provided a means for the new users of Tektronix first Desktop Graphics Computer, to share program information with other users by accessing library programs through either program contribution or a nominal charge. The applications library expanded significantly with input from the 4052 and 4054 Graphics Desktop Computer user base as the 4050 series grew. At its peak, the program exchange catalog included more than 700 programs.

In the past several years the activity level through the exchange slowed significantly. This year, usage of the Program Exchange was so light that support of the exchange has been discontinued. No new programs will be accepted.

A limited supply of the current program packages are available in inventory. While in stock, these may be ordered through the processes described below:

U.S. Orders

Order IDG Program Exchange packages through the toll-free number of Tektronix Central Parts Ordering. The following map delineates the geographical regions and the toll-free number serving each region.



Call the number serving your area and give the customer service representative the nine-digit part number and name of the Program Exchange package you wish. If you have any questions, call your local Tektronix Field Office. Effective July 1, 1984, each Program Exchange package will cost \$95 (U.S. dollars). Manuals, ordered separately, will cost \$15 (U.S. dollars) each. (Note: The minimum acceptable Tektronix order is \$25.00.)

Orders Outside U.S.

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

4000 Series Graphics Terminals

The 4010, 4020 and 4100 Series programs are packaged and nomenclated with the appropriate prefix and disk number.

Each package includes media with the source code together with the supporting documentation; listings are not included. Documentation may be purchased separately.

Package Title	Documentation Part #	Package Part #
4110 Local Programmability		
4110 LP Program Exchange Disk 1.....		062-6955-01
4110B LP Basic Slidemaker...	062-7740-00	062-7740-01
4110B LP TekniCAD Demo...	062-7679-00	062-7679-01

Software Support Category C

The program material contained herein is supplied without warranty of any kind, and without any representation regarding quality, performance or suitability. TEKTRONIX specifically disclaims any implied warranties

of merchantability of fitness for a particular purpose. Software support is TEKTRONIX Category C: Software is provided on an "as is" basis.

Package Title	Documentation Part #	Package Part #
4115		
VAX/VMS Driver for 4115 DMA (Option 3A).....	062-7305-00	062-7305-01
VAX/UNIX (Berkeley) Driver for 4115 DMA (Option 3A).....	062-7327-00	062-7327-01
4170 Local Programmability		
4170 MEMDRIVE.....	062-7741-00	062-7741-01
4170 Basic Slidemaker.....	062-7739-00	062-7739-01
4170 CGRAPH.....	062-7742-00	062-7742-01
4170 Color Drawing Board...	062-7766-00	062-7766-01

4050 Series Desktop Computers

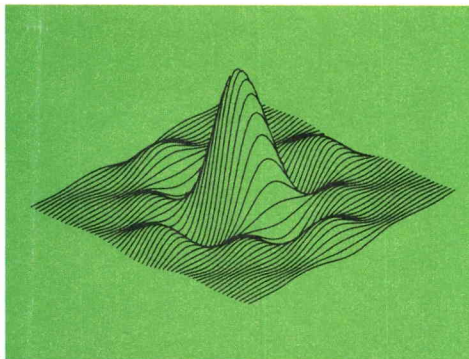
The 4050 Series programs included in the IDG Program Exchange prior to September 1981 are packaged and nomenclated by function. Those programs accepted after September 1981 are packaged and nomenclated with the Volume and Number of the corresponding issue of *TEKniques* in which the package was announced.

Each package includes the source code on tape or disk (T=tape; D=disk) together with the supporting documentation; listings are not included. Documentation may be purchased separately.

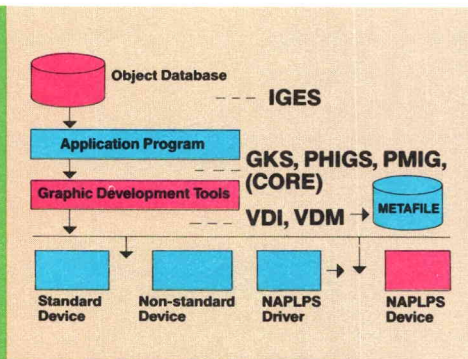
The IDG Program Exchange catalog contains the abstracts describing the programs in each package along with representative output in most cases. The catalog part number is 062-6343-00.

Package Title	Documentation Part #	Package Part #
Business Aids T1.....	062-5987-00	062-5987-01
Business Aids T2.....	062-5988-00	062-5988-01
CAD T1.....	062-5976-00	062-5976-01
CAD D1.....	062-5977-00	062-5977-01
Character Generator T1.....	062-5951-00	062-5951-01
Education/Research T1.....	062-5982-00	062-5982-01
Education/Research T2.....	062-5983-00	062-5983-01
Electrical Engineering T1.....	062-5978-00	062-5978-01
Graphing T1.....	062-5964-00	062-5964-01
Graphing T2.....	062-5965-00	062-5965-01
Graphing T3.....	062-5966-00	062-5966-01
Graphing D1.....	062-5967-00	062-5967-01
Graphing D2.....	062-5968-00	062-5968-01
Interfacing T1.....	062-5984-00	062-5984-01
Mapping T1.....	062-5980-00	062-5980-01
Mechanical Engineering T1.....	062-5979-00	062-5979-01
Programming Aids T1.....	062-5971-00	062-5971-01
Programming Aids T2.....	062-5972-00	062-5972-01
Project Aids T1.....	062-5985-00	062-5985-01
Project Aids D1.....	062-5986-00	062-5986-01
Recreational Plots T1.....	062-5989-00	062-5989-01
Slidemaker T1.....	062-5962-00	062-5962-01
Slidemaker D1.....	062-5963-00	062-5963-01
Text Processing T1.....	062-5969-00	062-5969-01
Text Processing D1.....	062-5970-00	062-5970-01
Utilities T1.....	062-5974-00	062-5974-01
Utilities D1.....	062-5975-00	062-5975-01
Tekniques Vol. 5 No. 4 T1.....	062-5981-00	062-5981-01
Tekniques Vol. 6 No. 1 T1.....	062-6443-00	062-6443-01
Tekniques Vol. 6 No. 1 D1.....	062-6442-00	062-6442-01
Tekniques Vol. 6 No. 2 D1.....	062-6515-00	062-6515-01
Tekniques Vol. 6 No. 3 D1.....	062-6516-00	062-6516-01
Tekniques Vol. 6 No. 3 T1.....	062-6514-00	062-6514-01
Tekniques Vol. 6 No. 4 T1.....	062-6704-00	062-6704-01
Tekniques Vol. 7 No. 1 D1.....	062-6715-00	062-6715-01
Tekniques Vol. 7 No. 3 D1.....	062-7276-00	062-7276-01
Tekniques Vol. 7 No. 3 T1.....	062-7275-00	062-7275-01
Tekniques Vol. 7 No. 4 T1.....	062-7455-00	062-7455-01
Tekniques Vol. 7 No. 4 T2.....	062-7456-00	062-7456-01
Tekniques Vol. 8 No. 1 T1.....	062-7611-00	062-7611-01
Tekniques Vol. 8 No. 1 D1.....	062-7612-00	062-7612-01

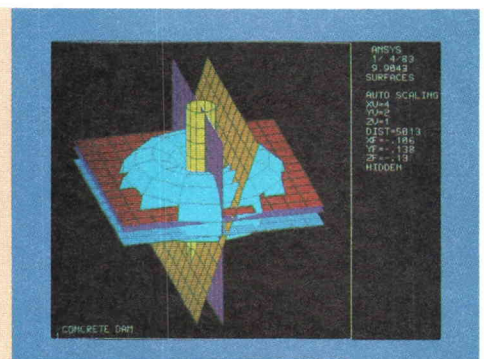
Tekniques



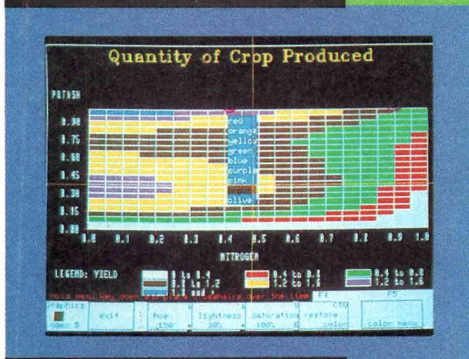
4050 Series



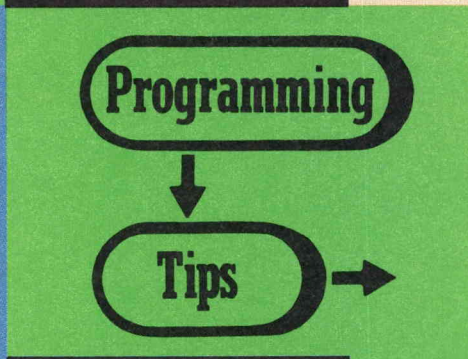
Special



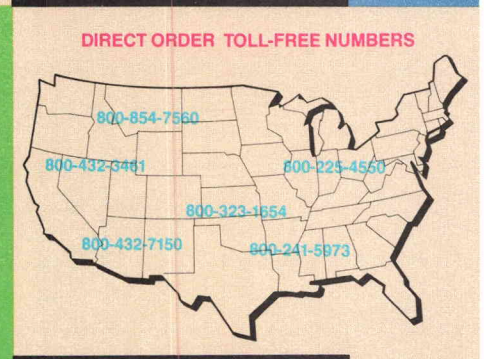
4110 Series



4100 Series



Programming Tips



Program Exchange

Tekniques keeps you up to date . . . and it won't cost you a penny.

Name _____ Date _____
 Title _____ Company _____
 Department _____ Telephone _____
 Street Address _____
 City/State/Zip _____

What Tektronix products do you have (check all that apply)

- 4050 Series 4100 Series 4020 Series 6100 Series
 4010 Series 4110 Series 4404 Series 6200 Series

My application of graphics is in:

- architectural engineering management/presentations scientific/technical data analysis
 electrical design/engineering mapping/cartography
 drafting mechanical design engineering

I'm using graphics in the following area:

- aerospace computers energy/petroleum
 automotive education government
 chemical electronics military
 communications

Now you can subscribe to the Tektronix IDG quarterly publication, *Tekniques*—and reap all the benefits of information designed to help Tektronix users stay ahead. Just complete and return this postage-paid subscription form and receive . . . absolutely free . . . valuable and timely abstracts, programming tips, new product announcements and applications features. Don't wait—send in the coupon today.

Tektronix
 COMMITTED TO EXCELLENCE



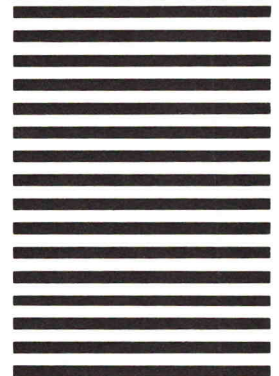
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 1 BEAVERTON, OR

POSTAGE WILL BE PAID BY ADDRESSEE

Tektronix, Inc.
P.O. Box 1700
Beaverton, Oregon 97075



IDG Graphics Displays and Computing System Publications

The following is a summary of all current manuals related to IDG products.

The manuals may be ordered through the Tektronix Central Parts Ordering office serving your area. See page 20 for the map and numbers.

Key to Titles

Operator's = operation instructions for hardware product
 Service = maintenance information
 Instruction = operation and maintenance combined
 User's = operation instructions for software product
 Reference = programmable features

4041 Systems Controller

4041 Opt. 3 Prog. Ref. Manual 070-5202-00
 4041 Opt. 3 Operators 070-5203-00
 4041 Opt. 3 Service 070-5204-00
 4041 Opt. 3 Programmers Reference Guide 070-5205 -00

4100 Series Computer Display Terminals

Standard
 4106/4107/4109/CX CDT Reference Guide 070-4892-01
 4106/4107/4109/CX CDT Prog. Ref. 070-4893-01
 CX 4100 Computer Display Terminal Operators 070-5256-00

Service
 4106/4107 CX Computer Display Terminal Service 070-4889-01
 4109/CX Computer Display Terminal Service 070-4890-01

Options
 4100F19 Interface for 4690 Series Color Graphics Copies 070-5199-01
 Tek E-Z Test Software for the 4105 Terminal 070-5225-00
 CX4100 Series Computer Display Terminal Host Support 070-5257-00

4110 Series Computer Graphics Terminals

Options
 4115B Series Command Reference 070-3892-01

PLOT 10 Software

400P70 Graphical Kernel Systems Reference 070-5296-00
 User's 070-4993-01
 User's 070-4994-01

4100P30 Computer Aided Drafting

User's 070-5248-00

4170 Graphics Processing Unit

Service
 4170 Memdrive 062-7741-00
 4170 Graphics Processor Service 070-7615-00
 Options
 4170 Color Drawing Board 062-7766-00

4510 Color Graphics Rasterizer

Service
 4510 070-5042-00

4970 Cluster Controller

4970 System Reference 070-4810-01

4970P01/02 3270 Controller Software

User's 070-4771-01

634 Video Display Monitor

Option 20 Test Fixture Power Supply 061-3021-00

GMA 303 Color Raster Monitor

Instruction Sheet 070-5260-00

61KP01 Hard Copy Interface

Service 070-5335-00

Segment Subroutine-Window Mgmt. Support

Reference 070-5138-00

IDG Program Exchange

Africa, Europe, Middle East

Contact local sales office

Australia

IDG Program Exchange
 Tektronix Australia Pty. Limited
 Sydney
 80 Waterloo Road
 North Ryde, N.S.W. 2113

Canada

IDG Program Exchange
 Tektronix Canada, Inc.
 P.O. Box 6500
 Barrie, Ontario
 Canada L4M 4V3

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

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

Japan

IDG Program Exchange
 Sony/Tektronix Corporation
 9-31 Kitashinagawa-5
 Tokyo 141 Japan

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-F0

MICHAEL R OLIVERI

TEKNIQUES

Address Correction Request — Forwarding and Return Postage Guaranteed.

RECEIVED
DEC 17 1984
RECEIVED
L. I. FIELD OFFICE