TEKTRONIX®

committed to technical excellence

TEKSCOPE

Volume 8 Number 4 1976

Two graphics technologies merged

Charles Ceranowski

The 4081 Interactive Graphics Display inaugurates a new era of sophisticated graphics terminals designed to work effectively as a terminal in a host computer environment, and with the aid of internal programming tools, makes a giant advance forward in operation as a highly flexible stand-alone graphics display.

Graphic displays in the past have used storage tube technology to offer high-resolution graphics of complex images and refresh-only tube technology to provide motion. Now, years ahead of its time, the two technologies are merged into one impressive system with the advantages of both refresh and storage in one tube. For the first time hardware and software have also merged to provide an integrated package—a powerful, interactive graphics display.

A low-cost storage terminal produces an excellent image presentation, even of complex pictures, but these images have proven to be static; using a refresh terminal for its dynamics provides excellent picture capabilities, but its extensive electronics have proven to be expensive. The 4081 gives a user the best of both—refresh and storage together, at a price much lower than a refresh-only display. The 4081 does this in four ways:

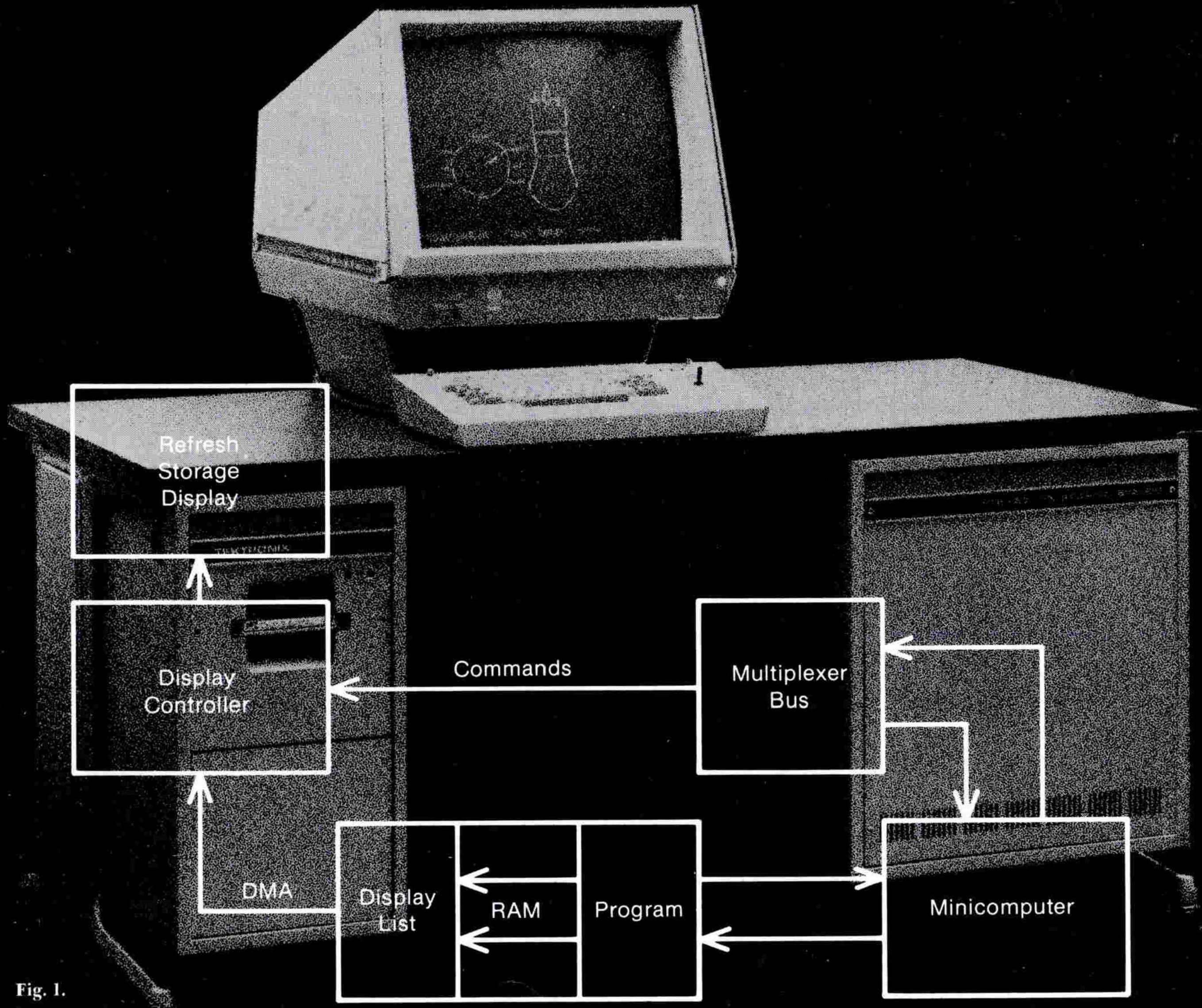

Fig. 1.

1. The standard tube technology combines storage and refresh capabilities.

2. In the 4081 this capability has been enhanced by the addition of local intelligence and a unique display controller that can *write* stored images and *maintain* refreshed images.

3. The 4081's picture processor and local storage capability also combine to lower communications overhead, timesharing costs, and host loading costs.

4. The same software that supports simultaneous refresh and storage operations also efficiently channels graphics data and commands to the proper system components.

In total, the 4081 adds up to better design capabilities and better business management.

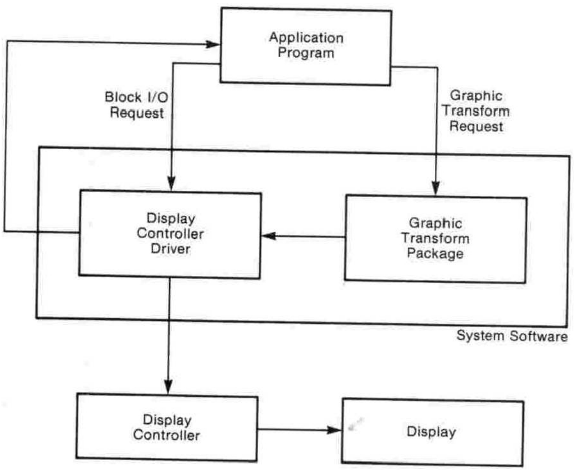

Fig. 2. Software overview. The software consists of two major areas—Application and System. The Application program makes requests to the System software for graphics output.

Hardware Operational Overview

The hardware for mixing refreshed and storage images is shown in Figure 1. It consists of a picture processor, a random access memory (RAM), a 19-inch refresh/storage display monitor, and a microcoded display controller. On command from the processor, the display controller accesses the display list in memory via the high-speed direct memory access (DMA) channel. This display list contains beam positioning and status information. The display controller directs the generation of vectors from which a picture is constructed.

The 4081 may function in three distinct modes: command, host, and program.

Command Mode: This is the mode of the 4081 when the Graphic Operating System (GOS), the interface between a program on the 4081 and the 4081 system hardware, is initially IPL'ed (loaded into memory). The 4081 is under operator control using the GOS command processor; GOS commands set various system parameters and invoke data files and programs to be executed.

Host Mode: Host mode instructs the 4081 to act like a full-ASCII alphanumeric terminal, similar to a teletype.

Program Mode: This mode is controlled by programs such as the 4014 Emulator, IGT, TECO, Assembler, and utilities. For programs that use the asynchronous communications interface, the communications parameters are usually set by the system COM command, and the communications interface must be connected to a modem or RS-232 computer communications line.

Software Functional Overview

The software can be divided into two major areas: Application and System (Figure 2). The Application program makes requests to the Graphic Operating System software for graphic output. Before actual display,

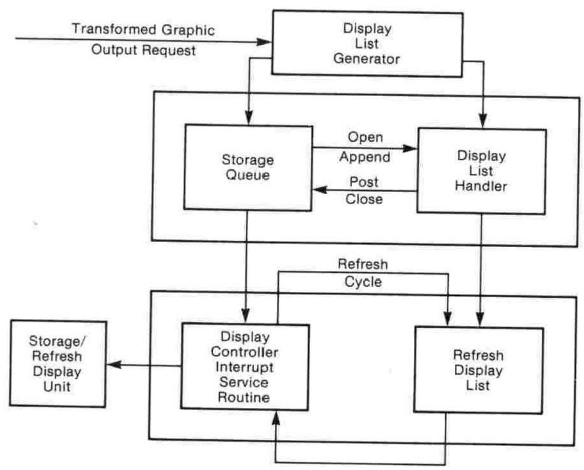

Fig. 3. Display controller driver. The display controller driver allows distinct graphic entities to be manipulated independently of one another.

the graphic output is manipulated by the Graphics Transform Package (GTP).

Graphics Transform Package. The GTP is responsible for Rotation, Scaling, Clipping, and Window/Viewport Mapping. The GTP also provides virtual graphics input and control, and support refreshed picture control. As an added feature it allows the operator to effect graphics input and output from, and to, a wide variety of graphic input devices without regard to the unique transformations required for each device. This feature is known as "device independence." Once transformed, the graphics data is passed to the Display Controller Driver (DCDVR), which builds display lists for the display controller.

Display Control Driver (DCD). The DCD (Figure 3) consists of three major modules: the Display List Generator which is responsible for taking transformed graphic output requests and generating display lists displayed as stored output or passed to the second module; the Display List Handler which manages the insertion, deletion, and modification of the refreshed display list; and the Display Controller Interrupt Service routine which is responsible for starting up a display list and switching between storage and refresh output. Together, these modules manage the output of storage graphics while maintaining refresh display lists. The Display Controller Driver supports a transformed display file maintaining separate display code sequences for each individually identified refreshed picture. This allows distinct graphic entities or components to be manipulated independent of one another. Any graphic output request which is not identified with a refreshed object is directed between refresh cycles to the storage display.

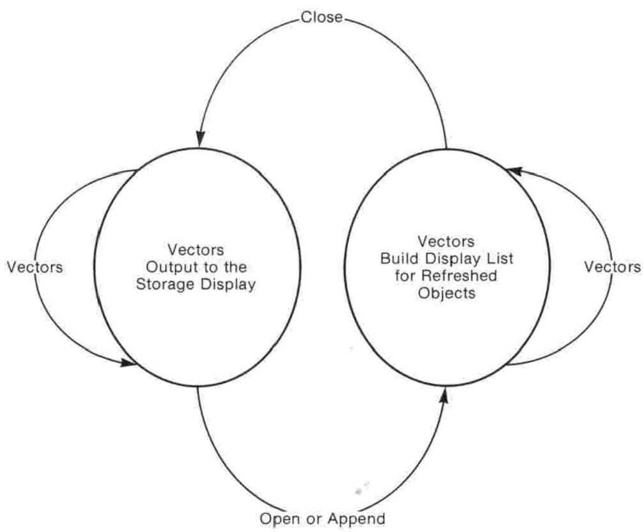

Fig. 4. State diagram depicting construction of an object.

Graphic output requests to the Graphic Operating System are made with a parameter block as outlined below:

The OP-CODE specifies the type of graphics (i.e., MOVE, DRAW, DASH, or POINT and ABSOLUTE or RELATIVE), while X and Y fields specify an X-Y coordinate pair to which the beam is to be positioned. For example, to draw a box in storage with sides of 100 units at (0,0) the following sequence of vector requests would be made:

OP-CODE	X	Y
MOVE ABSOLUTE	0	0
DRAW RELATIVE	100	0
DRAW RELATIVE	0	100
DRAW RELATIVE	−100	0
DRAW RELATIVE	0	−100

The display controller driver takes these requests and builds display code in fixed length blocks of memory. The interrupt service routine outputs the storage graphics between refresh cycles. When displayed, these blocks of memory are re-used for subsequent storage output requests.

A sequence of vector output requests, like the one given normally generates a stored image. The display code generated for the picture is traversed once and discarded. However, the user may conditionally save the display code generated by such a sequence as a refreshed object. This is done by preceding vector output requests with an OPEN command. This builds the display code generated by graphic output requests into dynamic memory areas. A CLOSE command stops the display code capturing process, and any further graphic output reverts to the storage display. The just-built 'refresh' object is not visible on the screen but is available for individual manipulation.

The process of object construction may be summarized by the state diagram in Figure 4.

An existing object may have vectors added to it by an APPEND command. A KILL command deletes the object from the system and returns the memory used to hold the display to the dynamic memory pool.

Plot 80: Graphics Operating System

The Plot 80: Graphics Operating System is the interface between a program on the 4081 and the 4081 system hardware.

If the user is a 4081 internal programmer, he may use GOS to simplify application program development. Many facilities, which must be coded by the users of other graphics systems, are provided as part of the GOS even in its very minimum configuration.

One of the more significant elements of the Graphics Operating System for use by the 4081 internal programmer is the Graphics Transform Package (GTP) that, as cited previously, does clipping, rotation, and scaling and allows the selection of window-to-viewport mapping. It also provides virtual graphics input and output and supports refreshed picture segment control. Using GTP the operator is able to perform graphics input and output operations from, and to a wide variety of graphic input/output devices without regard to the unique transformations required for each device (device independence). For example, data may be read from disc or tape; graphic input may be obtained from the joyswitch or the tablet; listings may be directed to the display, line printer, or to a disc or tape file. Device independence also allows the internal programmer to direct his effort toward his specific application, rather than being required to be cognizant of all the hardware details of a given graphics device.

Several window/viewport pairs may exist on a display and each may be referenced as separate logic units. This capability enhances the graphic device independence of the system for a 4081 internal programmer.

GOS graphics input services are also device independent with transformations in three forms: screen coordinates, virtual coordinates relative to some window/viewport pair, and virtual coordinates relative to some input device window.

The Graphics Operating System also supports two different software character types: 1. Characters as elements of pictures. 2. Characters as elements of computer messages from, or to the human user within the monitor viewport. Character type 1 allows the continuous scaling, rotation, translation, and clipping required in a picture. Character type 2 circumvents the portions of Graphics Transform Package when displaying the character. A carriage return and line feed are automatically inserted at the right margin of the monitor viewport, so as to avoid loss of important display information.

Plot 80: Intelligent Graphics Terminal (IGT) Package.
Plot 80: IGT resides in the 4081 and responds to commands from a host computer by initiating the execution of the specified capability within the 4081, such as the refreshed display of a picture. When used with a suitable host resident support package, the IGT package allows the 4081 to behave as a sophisticated "intelligent graphics terminal." Access to the IGT from the host computer is through a set of FORTRAN-callable routines which are resident in the host computer. Each routine performs a conceptually simple task, but when taken together provide the programmer with access to the full power of the 4081.

Uses of the 4081

As of the date of this article the 4081 has been delivered to many sites throughout the continental United States and Europe. *Uses of the 4081 in performing interactive graphics have been varied* and include among others:

Design of automobile bodies
Electronic equipment design
Assembly plant layout
University course scheduling
Schematic drafting applications
Complex mathematical analyses
Design of oil drilling platforms
Military mapping applications
Sophisticated weapons planning
Grey scaling applications

For the majority of tasks that require interactive graphics, the 4081 has been designed to satisfy needs of the customer and to serve as a solution to his graphics applications. Future advances in the design of the 4081 will open many new thresholds for Tektronix to explore within interactive graphics. The highly effective graphics capabilities of the 4081 Interactive Graphics Display, its wealth of graphics software, its ease of programming, and its low cost all will contribute greatly to both the success of the 4081 for Tektronix as well as successes for the entire graphics industry in the United States and abroad.

Acknowledgments

Equipment of the complexity of the 4081 can only be a success with the contribution of a wide variety of personnel. Contributing to the hardware design of the 4081 were David Heinen, Roger Handy, and Donlan Jones. Software design and data were handled by Jon Meads, Gary Neher, and Ned Thanhouser. Data for inclusion in this article were mainly obtained from these individuals. In addition invaluable assistance and technical support for information included in this article was obtained from Richard Drew, Curt Coleman and Jeanne Judah in Marketing and George Rhine in Engineering.