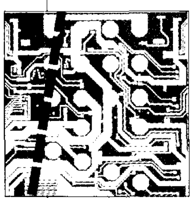
Anatomy of an X terminal

Based on a client/server model, an open, vendor-independent standard offers much flexibility on a local-area network



Not too long ago, a software engineer working on a complex project (such as a device driver or a high-level application) had to switch among different editing sessions to access all the files needed. This information resided on different hosts connected in some type of network and was generally displayed on a character-cell-based ASCII terminal.

That setup is giving way to an environment in which several computer resources can be called up concurrently over an open, distributed network that embraces an entire enterprise. The X Window System is an open, vendor-neutral standard based on the client/server model. This model permits physically independent application processing and display processing, communicating over a local-area network by means of a well-defined set of graphics commands that comprise the X protocol.

X terminals' Graphical User Interface (GUI) [Fig. 1] improves significantly on the ASCII terminal interface. While ASCII terminals usually support only one character font and only one or two windows per display, X terminals can support any number of windows with any type of font or window size. Also, the X terminals' bit-mapped screen allows the application to display all sorts of data formats (text, images, drawings, and so on) simultaneously, enhancing the user's productivity. Finally, since X terminals are diskless, the user does not have to serve as a system administrator; yet the human interface is such that the user feels as if he or she is working at a very powerful workstation.

In fact, in hardware terms, an X terminal is like a workstation that is diskless: it therefore lacks such features as high-speed disk controllers, large caches, and memory management units that are needed in workstations with disk storage. The focus is instead on network communications, graphics performance, and graphical user interface enhancements, and the end result might be viewed as an "application-specific workstation," in the sense that its hardware and software are optimized for running the X protocol.

Because of this dedication, X terminals are generally 40–50 percent more cost-effective than workstations for running X. For example, the IPC color workstation from Sun Microsystems Inc., Mountain View, Calif., clearly a cost-effective Unix workstation, retails for US $8995, while NCR Corp.'s XL-17c X terminal retails for $4470. Similarly, Sun's XLC monochrome workstation retails for $4995, while the 15b unit from Network Computing Devices Inc., Mountain View, Calif., is sold for under $1600.

HUMAN FACTORS. Many factors affecting the human interface are considered in the design of an X terminal. The display monitor should limit eyestrain. The monitor's footprint determines how the unit can be positioned in the work environment. The keyboard and mouse should feel comfortable to the touch. The goal is to make an X terminal easy to use for extended periods of time.

The design of the display monitor should above all minimize visual fatigue. The average human eye detects noticeable flicker at a frequency of 60 hertz, so a vertical refresh rate of greater than 70 Hz is recommended. In the past, a 60-Hz vertical refresh rate was used to achieve a frequency which would minimize aliasing. However, 60 Hz still registers as flicker on most people's peripheral vision (look away from the center of a monitor at about a 30-degree angle, and the screen will seem to flicker to a degree that depends on the persistence of the phosphor it uses and the age of the viewer).

Pincushion distortion and lack of focus quality can strain the eyes also. Although pincushioning is most noticeable as a bulging at the sides or top and bottom of a screen (making it look like a pincushion), it distorts everything displayed. Focus quality should be judged across the entire display; a poor-quality monitor may be in focus in the center, but not on the edges or corners. The eyes' attempt to compensate for distortion and poor focus tends to strain them.

Other ergonomic factors that enhance a user interface are overall display resolution and display size. As resolution increases, so does the amount of information an operator can view simultaneously. If the resolution is held constant, increasing the display size makes text appear larger by decreasing the dot-per-inch density, which may be desirable to some users.

Generally, X terminals come in two cabinet styles, with the logic unit either integrated with the monitor or resident in a separate base, which in most cases can support the monitor. There are advantages and disadvantages with each cabinet style. With a modular cabinet, an X terminal vendor may let the user choose the monitor he or she prefers, or even use a monitor that he or she already has. A modular cabinet may also be placed in an out-of-the-way location, such as under a desk. On the other hand, an integrated cabinet generally provides a lower-cost solution (compared to a modular cabinet unit and monitor of the same size and resolution) because it can use dc power from the monitor rather than requiring its own power supply.

COMMUNICATIONS. Since the X Window System protocol is implemented for a client/server architecture, a great deal of data often travels between the client (typically running on the host) and the server (typically running on the X terminal). Because of the interactive nature of the X terminal application and the volume of traffic that must move between the two machines, traditional RS-232C style communication is not always good enough for the X protocol.

RS-232C serves today's ASCII terminal, which some see being replaced by the X terminal. After all, like an ASCII terminal, an X terminal can be designed to receive VT100, VT220, and/or 3270 style commands from a host using a non-X Teletype-network mode and port. However, in X mode, the

Defining terms

Content-addressable memory (CAM): also known as associative memory, a memory with comparators on each cell to allow a parallel comparison of the input data with all the memory contents.

Point-to-point protocol (PPP): a variation of the serial line internet protocol (which see) that minimizes the amount of data sent by removing redundant protocol information from sequential data packets.

Serial line internet protocol (SLIP): a variation of Ethernet's TCP/IP (short for transmission control protocol/internet protocol) used in X terminals to let them transmit data packets and access several hosts through an RS-232C serial communication line.

Simple network management protocol (SNMP): an application-level protocol that allows logically remote users to inspect and alter network-management variables.

Angel E. Socarras, Robert S. Cooper,
William F. Stonecypher
NCR Corp.

0018-9235/91/003-0052$1.00©1991 IEEE

terminal must in addition be able to pass bit-mapped images, window management commands, and pointer information between the server and the client using the X protocol.

X terminal vendors approach the RS-232C scenario differently. Some design the terminal as if it had to rely solely on RS-232C as the communication link and therefore reduce traffic between the host and the terminal. These vendors accordingly split the server software—the X server—to run on the host and on the terminal [Fig. 2], a redistribution of terminal resources that also lowers terminal cost. Then, by using data compression and a proprietary protocol, they reduce traffic over the serial link enough to get acceptable performance for some applications using a 19.2-kilobaud, or faster, link. The major drawback of this architecture is that vendor-specific software must be ported to each X terminal's host, since a nonstandard protocol, rather than the standard X protocol, is traveling the link.

Since X Windows demands support for multiple simultaneous windows, vendors of RS-232C-based X terminals implement internet style protocols such as TCP/IP over SLIP [see Defining Terms]. Two or more hosts may then communicate with a single terminal, but at 19.2 kBd the performance is quite poor.

A higher-performing protocol that is becoming popular among RS-232C users is PPP. Point-to-point protocol, to spell it out, reduces the amount of information transmitted by "compressing" the TCP and IP header information, making a 19.2-kBd system quite usable. Few hosts currently support this protocol, though.

Other vendors provide RS-232C in addition to some preferred high-speed network link, but the RS-232C port is not used for the X protocol. Thus, an RS-232C printer can also be connected to the terminal for local screen dumps or to provide printing services to other local-area network (LAN) users. Also, since an ASCII terminal is often required for workstation consoles, the RS-232C port can be used to provide a "console window" on the X terminal display.

Most of the first-generation X terminals used 8- or 16-bit controllers as part of their Ethernet interface. The interface chip sets usually consisted of three components—a controller, a Manchester encoder/decoder, and a transceiver. Second-generation X terminals are employing 32-bit Ethernet controllers that not only have a wider data path, but are available with the Manchester encoder and, possibly, the transceiver and other discrete components on a single substrate. This reduces the component count, board space, and power consumption. The designers of 32-bit Ethernet controllers have also reduced the amount of intervention necessary for the system processor to handle Ethernet traffic.

The network management capability of the parts has also been enhanced. Much of the information that is vital to network management protocols, such as the simple network management protocol (SNMP), is now directly available from the Ethernet hardware. In the past, most of this information had to be tracked in software. Content-addressable memory is also beginning to be offered for applications where high-speed broadcasting to select network nodes (multicasting) is important, as on DECnet.

GROUNDING IN GRAPHICS. Graphics depends heavily on memory architecture. For an idea of its importance, consider how X terminals display information—in a bit-mapped fashion on a cathode-ray tube.

A typical display consists of 1024 horizontal picture elements (or pixels) by 800 vertical elements, or approximately 800 000 pixels. All the pixels need to be re-displayed 70 times per second.

For every pixel location on the screen, there is a bit or set of bits in memory: 1 bit for a monochrome display, anywhere from 4 to 8 bits for color. Grayscale displays require 2 to 4 bits to support 4 or 16 levels, respectively, of gray tones ranging from black to white. The display information is stored in an area of memory that is called the frame buffer because the data for the next screen to be displayed is stored in it while the current frame is displayed.

To implement a frame buffer in hardware, the most cost-effective approach (especially for monochrome systems) is to use a section of the terminal's main memory as a buffer and periodically copy its contents into a 256-bit serializer. The serializer outputs the pixel stream serially at the pixel scanning frequency.

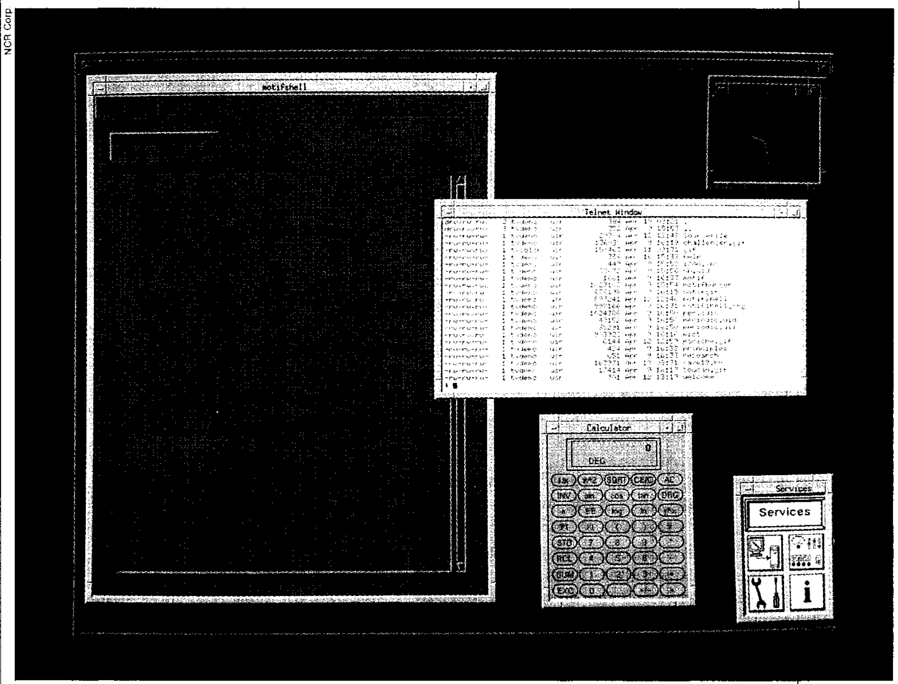
There is one major drawback, however. Since standard dynamic RAMs (DRAMs) have only one access port, a certain percentage of bus time must be used for loading the serializer, preventing the system processor from updating the frame buffer. This might not hamper single-bit-per-pixel monochrome displays or systems with big write-through caches, but it becomes prohibitive in multibit-per-pixel systems [Fig. 3].

Today's most popular frame buffer implementation uses a fairly recent innovation in memory chip design called video memory, or VRAM. In addition to a DRAM, a VRAM contains an n-bit-long shift register that can be loaded in parallel from the DRAM's n sense amplifiers and output data through its own serial output port—hence the other name for this type of memory, dual-ported RAM. This data can be shifted out of the register without using the random access port or taking up bus bandwidth. The shift register in the VRAM is used as a serializer to refresh the display. When it empties, the processor performs a transfer cycle and reloads the shift register with new data—the only overhead for VRAMs.

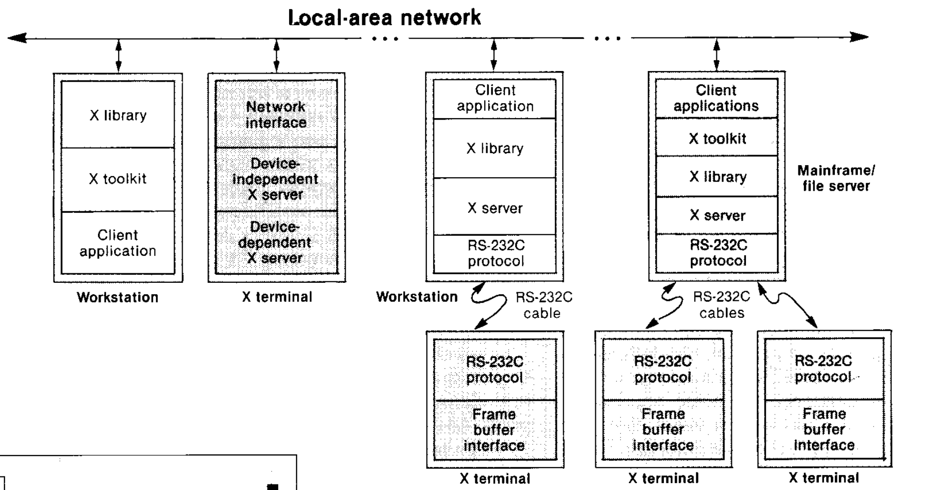
Since the VRAM allows the processor to utilize the random access port more efficiently, it provides faster graphics performance. However, this added performance is not free. For the same memory density, prices of VRAMs are about twice as high as for DRAMs. Still, for color and gray-scale applications, VRAMs make the most sense.

Another important use of memory X terminals is for look-up tables. For displays, a look-up table can be thought of as a way

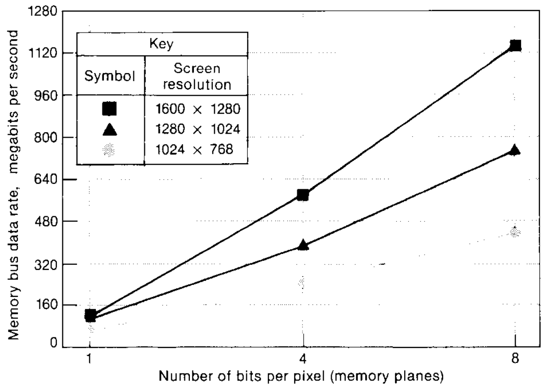
NCR Corp



[1] X terminals' ability to handle multiple windows and mouse commands calls for more elaborate hardware than was needed by their predecessor, the ASCII computer terminal.



[2] X terminals fit into today's networked environments in several different ways: connected directly to the local-area network, or linked to a workstation or other host with an RS-232C serial interface. The way they are used affects not only their interface hardware but their software requirements as well.



[3] With a dynamic RAM frame buffer, display size and the bits per pixel determine memory bus performance requirements. Monochrome (1 bit per pixel) does not have much impact, but high-resolution color (8 bits per pixel) demands a very fast bus.

Some color X terminals

Product	Company	Resolution, picture elements (W × H)	Screen diagonal, inches	Price, U.S. $
View Station 21c	Human Designed Systems Inc.*	1280 × 1024	21	$6999
Xstation 120/6091	IBM Corp.		19	$9686
XP29	Tektronix Inc.		19	$5995
17c	Network Computing Devices Inc.		17	$5000
XL3412 17C	NCR Corp.	1024 × 768	17	$4800
700/X	Hewlett-Packard Co.		16	$4995

Source: Dataquest Inc. * King of Prussia, Pa.

of mapping a set of predefined values to different domains having either a different number of elements or more information content. Its most common application is to increase the number of displayable colors without increasing the number of bits per pixel.

A color display with n bits per pixel that does not use a look-up table can display only 2^n colors at a given time. The n bits select a value for the red, green, and blue components of each pixel in the color display, maybe $n/3$ bits for each of the three electron-beam colors (red, blue, and green). For instance, a 6-bit-per-pixel display could assign 2 bits to each of the three color elements, which would result in a range of four shades per color element. In combination, the four shades per element result in 64 possible colors.

The use of a look-up table, however, yields a larger range of combinations. The table employs the number represented by all n bits as an index to a table of values m bits wide. The total number of colors is now a function of m and not n. That is, any 2^n subset of 2^m colors can be displayed. For example, a 16-bit-wide look-up table can contain what is commonly referred to as a color palette of over 64 000 colors. At 6 bits per pixel, 64 colors can be selected from the palette and displayed on screen at any instant.

CURSORY NOTES. The X terminal GUI specifies a cursor block directly controlled by the mouse. Since multiple windows can reside in the same display, the cursor is used as a pointing device to the window being worked on. The simplest way of implementing the cursor block is for it to be painted on the screen by the display processor when the electron beam reaches the right scan line, and then removed after the beam has passed. One way of implementing this would be to have a timer interrupt the processor before and after the electron beam passes through the cursor area. For example, Texas Instruments Inc., Dallas, builds this interrupt feature into its TMS34010 and TMS34020 graphics processors.

When the interrupt is sensed, the display processor performs a block transfer of the cursor information into and out of the display area—to the detriment of the terminal's overall graphics performance, however.

An alternative is to implement the cursor completely or partially in hardware, unburdening the processor and resulting in better overall performance. Thus some color X terminals drive their displays with a video controller and color palette combination that have, or make it possible to add, cursor logic. For example, the Bt459 from Brooktree Corp. in San Diego, Calif., provides a 64 × 64 cursor block. Some other of the RAM-plus-digital-to-analog-converter devices like the Bt458 provide several overlay inputs to help implement a cursor block externally. All of these devices have built-in look-up tables that modify the display information when the cursor block is being displayed.

OPTIMIZED DRAWING. To achieve the best performance possible in X terminals, vendors will spend time optimizing drawing operations as well as the upper layers of the X server. Hardware may be used to support the lowest level of graphics primitives, with lengthy operations (such as primitives that move windows) being prime candidates for optimization. Hardware interfaces that work in conjunction with the microprocessor during the drawing operations are used by some vendors like Network Computing Devices and Datacube Inc. in Peabody, Mass. Other vendors, like IBM Corp., Hewlett-Packard Co., and NCR Corp., use dual processors to split the workload of rendering, X server, and communications overhead. Both methods have advantages.

Some X terminals use graphics assist methods to boost the performance of drawing operations such as a bit block transfer, also known as bitblt. A bitblt can be defined as a logical operation performed on bit-wise boundaries between source and destination regions; an example is the movement of a window from one position to another on the screen.

The graphics assist logic is an interface to the frame buffer and system memory. This interface—which handles barrel shifting for bit alignment, concatenation of data, and preprogrammed logical operations—"assists" the processor by letting it act merely as a two-dimensional direct-memory-access controller for the graphics assist interface. This can double the inner loop performance of the bitblt operation.

Some X terminals in the market today have two processors—a system CPU and a graphics processor—with which a certain level of parallelism can be achieved. For instance, in one approach, the system processor supports the communications and handles system interrupts, while the graphics processor runs the X server and handles the drawing operations. Some parallelism occurs here because the CPU can process TCP/IP packets for future operations while the X server running on the graphics processor finishes past operations.

Other vendors make the system processor responsible for communications and the top layers of the X server (where event handling and region clipping is performed), passing only drawing commands to the graphics processor. Parallelism can occur here because the X server can process several commands while the graphics processor is drawing previous operations.

As an example of how parallelism can be used in multiple-processor X terminals, suppose an engineer is running a computer-aided design (CAD) application and a series of lines and text representing a schematic is displayed. The application displays a button labeled "redraw." When the button is "pressed" (by using the mouse), the following events may occur:

- The button press is processed and passed to the application.
- The application recognizes the event by highlighting the button.
- It clears the CAD drawing area.
- It recreates a list of graphics primitives to redraw.
- It sends the list to the X terminal.
- It unhighlights the redraw button.

Here, the first event is initiated by the X server, while the others are initiated by the CAD application on the host.

The fifth event gives the best example of how parallelism can be used. As the list of graphics primitives arrives at the X terminal (in one or more packets), the portion of the X server running on the system processor can clip the primitives and place them one by one into a drawing queue. While the system processor builds this queue, the graphics processor can asynchronously draw the screen using the commands taken from the queue. Obviously, parallelism can significantly improve X terminal performance.

RATING PERFORMANCE. Crucial here is the terminal's response time; a user does not want to sit in front of a terminal for 5 seconds waiting for a page to be painted or a Unix session to start. Total response time is affected by network bandwidth and utilization, host performance and loading, X server implementation, and graphics rendering performance.

There have been a few attempts at rating terminal performance. The most popular is a test suite developed by Claus Gittinger from Siemens GmbH, headquartered in Munich. This benchmark, commonly known as Xbench, has the host send a set of graphics commands to the terminal, and measures the time the terminal takes to complete the commands. The tests cover the six most common operations: line drawing, rectangle drawing, bitblts, arcs, text rendering, and complex operations. The output of this benchmark has three forms: absolute performance data from the individual tests, statistics and relative performance data for each test subgroup, and a weighted average of relative performance named "xstones." All the relative performance ratings are based on a direct comparison with the performance of a SUN 3/50 workstation running the X11.3 version of X under OS 3.4.

> Present tests of X terminals don't measure true performance

Another performance bench mark is X11 perf. Made up of 166 tests, it is more comprehensive than X bench but does not give a single rating.

These tests provide a good measure of the X terminal's graphics performance. However, they fail to measure the overall response time of the terminal; that is, the delay from the time a user causes an asynchronous event (by depressing a mouse button, say, or moving a window across the screen) to the time the proper response occurs at the terminal. There are no widely accepted benchmarks for this purpose and, until they arrive, X terminal performance evaluations are limited to the graphics component, perhaps the largest piece of the pie.

MARKET POTENTIAL. X terminals have been bought primarily for engineering design environment applications, such as CAD and computer aided engineering, where they have served as display devices with a lower cost per seat than traditional Unix workstations. Another market that has begun to migrate to X terminals has been manufacturing, with applications such as process control, which requires high-resolution displays.

X terminals have also been used to enhance material requirement planning applications by enabling the user to view multiple text modules from different hosts concurrently and to "copy and paste" among these modules. A number of markets requiring customer servicing applications—such as utilities, airlines, and freight forwarding—are implementing X terminal environments for concurrent viewing of different customer database records.

As X terminals achieve new price performance plateaus, the market is expected to expand to more business applications requiring the network connectivity and multiple viewing of applications that X offers [Table].

TO PROBE FURTHER. A good source on most issues on computer graphics hardware design is *Computer Graphics Hardware: Image Generation and Display* by Hassam K. Reghbati and Anson Lee (IEEE Computer Society Press, Washington, D.C., 1988). *Fundamentals of Interactive Computer Graphics* by J. D. Foley and A. Van Dam (Addison-Wesley, 1982) gives a more general overview of technologies involved in graphics rendering. Another important reference work, now in its second edition (1990) from Digital Press, Bedford, Mass., is *X Window System* by Robert Scheifler and Jim Gettys (who originally proposed the X Window System).

For a detailed understanding of the X Window System, O'Reilly & Associates Inc. offers a nine-volume series called *The Definitive Guides to the X Window System*, now in its second edition (Sebastopol, Calif., September 1990).

ABOUT THE AUTHORS. Angel E. Socarras (M) and Robert S. Cooper are senior principal design engineers at NCR Corp., Orlando, Fla., and William F. Stonecypher is a principal design engineer there. Socarras received two BS degrees in electrical and computer engineering from Florida Institute of Technology in 1985, and an MS in electrical engineering from Georgia Institute of Technology in Atlanta in 1986.

Cooper received his BS and MS in electrical engineering from the University of Florida at Gainesville in 1985 and 1987.

Stonecypher received a BS and MS in electrical engineering from Georgia Institute of Technology in 1986 and 1987. ◆