DOCUMENT RESUME

ED 124 148 IR 003 536

AUTHOR	Sherwood, Bruce Arne; Stifle, Jack
TITLE	The PLATO IV Communications System.
INSTITUTION	Illinois Univ., Urbana. Computer-Based Education Lab.
SPONS AGENCY	Advanced Research Projects Agency (DOD), Washington, D.C.; National Science Foundation, Washington, D.C.
REPORT NO	CERL-R-X-44
PUB DATE	Jun 75
CONTRACT	USNSF-C-723
GRANT	NSF-GJ-29981
NOTE	33p.
AVAILABLE FROM	PLATO Publications, Computer-based Education Research Lab, 252 Engineering Research Lab, University of Illinois, Urbana, Illinois 61801 ($1.00, prepayment required)
EDRS PRICE	MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS	Computer Assisted Instruction; Computer Graphics; Computer Oriented Programs; Computer Programs; *Computers; *Computer Science; Computer Storage Devices; Data Processing; *Electronic Equipment; Higher Education; Information Systems; *Input Output Devices; Instructional Media
IDENTIFIERS	Computer Hardware; *PLATO IV; Programmed Logic for Automatic Teaching Operations

ABSTRACT

The PLATO IV computer-based educational system contains its own communications hardware and software for operating plasma-panel graphics terminals. Key echoing is performed by the central processing unit: every key pressed at a terminal passes through the entire system before anything appears on the terminal's screen. Each terminal is guaranteed 1260 bits/sec of output at all time. There is no "store-and-forward" machinery, and output error and correction are integral parts of the system. (CH)

CERL Report X-44

THE PLATO IV COMMUNICATIONS SYSTEM

BRUCE SHERWOOD and JACK STIFLE

Computer-based Education Research Laboratory

University of Illinois Urbana Illinois

THE PLATO IV COMMUNICATIONS SYSTEM

Bruce Sherwood and Jack Stifle

Computer-based Education Research Laboratory*

University of Illinois, Urbana 61801

*We are reporting on work involving many members of this laboratory. Staff members involved in system software have included Paul Tenczar, David Andersen, Richard Blomme, John Carstedt (CDC), Ruth Chabay, Chris Fugitt, Don Lee, James Parry, Robert Rader, Masako Secrest, Bruce Sherwood, Donald Shirer, and Michael Walker. Staff members responsible for the development of PLATO communications equipment have included Donald Bitzer, Jack Stifle, Paul Tucker, and Michael Johnson.

This work has been supported by the State of Illinois, by the National Science Foundation (U.S. NSF C-723 and NSF GJ 29981), and by the Advanced Research Projects Agency (U.S. Army/DAHC 15-73-C-0077).

The PLATO IV computer-based education system contains unique communications hardware and software for operating hundreds of graphics terminals. Key echoing is performed by the central processing unit. Each terminal is guaranteed 1260 bits/sec of output at all times. There is no "store-and-forward" machinery. Output error detection and correction are integral parts of the system.

I. Introduction

The PLATO IV[1] computer-based education system hardware consists of a Control Data Corporation Cyber computer with electronic swapping memory, hundreds of plasma-panel graphics terminals, and a communications network. At the time of writing (July 1975), there were 950 terminals installed on the University of Illinois PLATO system. As many as 480 terminals have been active simultaneously. Additional PLATO systems are centered in Minneapolis (Control Data Corporation) and Tallahassee (Florida State University). System software includes a fast-response operating system and a compiler and executor for lesson materials written in the TUTOR language.[2] PLATO IV supports elaborate computer graphics including lines, characters from dynamically alterable character sets, and animated displays. All graphics modes include selective erase capabilities. This article will describe the communications component of the PLATO IV system.

PLATO IV is almost unique among multi-terminal interactive systems in eliminating the swapping of programs between the computer and the mass memory of mechanical disks or drums. Instead, electronic swapping memory is used (CDC Extended Core Storage). It has a transfer rate one hundred times greater and an access time one thousand times shorter than those of disks or drums. These enormous quantitative advantages make it possible to provide fractional-second response times to hundreds of graphics terminals.

A second distinctive feature of PLATO IV is that every key pressed at a terminal passes through the entire system, including the central computer, before anything appears on the terminal's screen. There is no local echoing of keys at the terminal. This means that the keyboard is redefinable: the "n" key is not restricted to causing an "n" to appear on the screen but may, in context, display a line drawing or an appropriate Cyrillic character.

The average response time is one-eighth second (125 milliseconds). Half of this delay is due to hardware: serial bit transmission on input and output, hardware buffering at various levels of the interface equipment, etc. The other half is due to software, mainly input and output polling and buffering; the actual processing is typically a few milliseconds and only a fraction of a millisecond for simple key echoing.

The processing of each keypress by the central computer has often been criticized as needlessly expensive. The appendix to this paper discusses some of the basic reasons why single keypress interactions are appropriate in the PLATO environment.

Sections II and III discuss in detail the communications hardware and software. Much of this discussion assumes an understanding of the basic communications configuration described briefly below. A simplified diagram of the equipment is shown in Figure 1.

Most of the PLATO communications equipment is designed with "classroom" clusters in mind. Up to 32 terminals in such a cluster are connected to a "Site Controller."[3] This Site Controller transmits information from a high-speed channel to the individual terminals. The Site Controller also concentrates keyset inputs from the terminals for transmission to the computer. The purpose of the Site Controller is to permit the use of highly multiplexed communications lines for long links. Input to the computer is carried from each Site Controller to the computer complex on a single voice-grade telephone line (which therefore serves 32 users). At the computer center a Network Interface Unit (NIU)[4] receives the inputs from 32 Site Controllers and delivers them to the computer for processing.

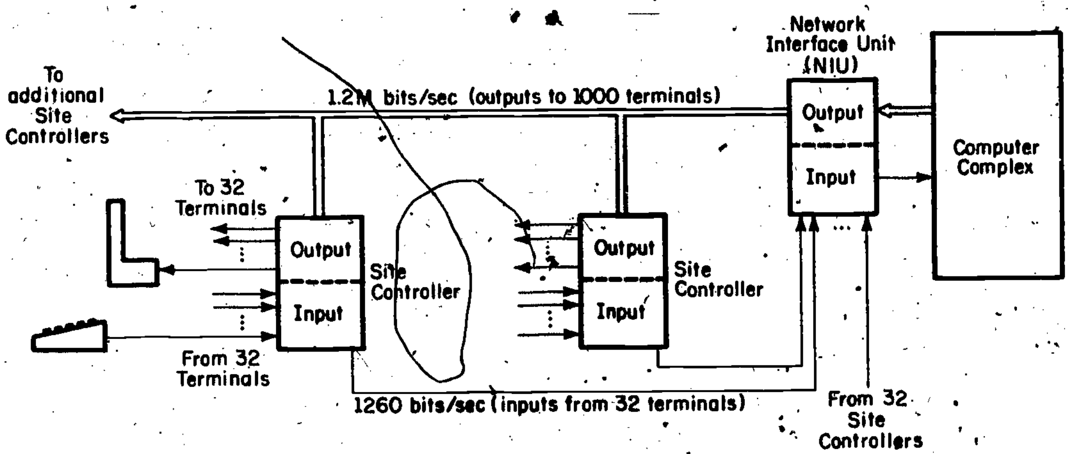

Figure 1. Communications hardware configuration.

In processing a key the computer normally generates output for the terminal. The computer delivers this to the NIU, which multiplexes the output data for 1008 terminals onto a television channel. The television-formatted data is transmitted via standard television techniques (cable or microwave) to the 32 Site Controllers sharing the TV channel. Each Site Controller recovers from the television channel all of the data destined for the 32 terminals serviced by that Site Controller and transmits the data to each terminal. The data rate between the terminals and the Site Controller is 1260 bits/second in each direction, which permits the data to be transmitted over ordinary telephone circuits. The terminals may, therefore, be remotely located from the Site Controllers.

Terminals within a few miles of the Site Controller normally have a four wire (full duplex) connection with one pair for input and the other for output. The more distant terminals use a modem (modulator-demodulator) which frequency-divides a single telephone line (one pair of wires, or half duplex connection) to carry both input and output simultaneously. Distant terminals may also be clustered in groups of four terminals using a more expensive modem to share a single telephone line. In any case, the ultimate input and output connections are through a Site Controller and the NIU, and all terminals are supplied with the full 1260 bits/sec at all times.

Probably the most stringent requirement of a computer-based education system is an average response time of less than 0.25 seconds. This requirement more than any other determined the nature of the PLATO IV communications system. It eliminated at the outset any system employing the use of store and forward techniques where messages may be temporarily stored and thus delayed while network conflicts are resolved. It eliminated systems employing polling techniques where terminals must wait until interrogated before sending data to the central computer. The response time requirement also implied the need for a full duplex mode of data transmission, one in which information can flow simultaneously in two directions. In half duplex systems where information can only flow in one direction at a time, the time required to reverse the direction of information flow is generally 0.4 seconds or more.

The data formats for PLATO were determined primarily by the graphical display nature of the data and by the need to transmit the maximum amount of data via the lowest possible data rate over readily available facilities. These requirements eliminated any serious consideration of the use of ASCII because of the high overhead associated with this code. For example, to send the 18 bits necessary to specify a point on the PLATO IV display via ASCII (8 to 11 bits per byte) would require the transmission of 24 to 33 bits representing an overhead (i.e., waste) of 25% to 45%. The PLATO IV data format, on the other hand, contains only 14% overhead in the case of graphic data. The use of such an efficient data format permitted the use of the relatively low data rate of 1260 bits per second which in turn permitted the use of low cost modems and the lowest grade telephone circuit.

The choice of a television channel for the distribution of the PLATO signal was in anticipation of the growth and availability of cable television

(CATV) systems which would provide a low cost ($6 - $10) data link to homes and schools in most metropolitan areas. In such systems PLATO would require the use of only one TV channel for every 1000 users. Although the anticipated growth of CATV has not occurred, thus preventing its use by PLATO, the TV approach is still economically attractive. For example, microwave is presently being used in lieu of CATV to distribute the TV signal in Champaign-Urbana, with the monthly operating cost of a PLATO IV terminal ranging from $1.45 to $2.35, as compared with $4.85 to $14.60 using telephone lines.

To further expand the distribution of PLATO materials, an effort is now under way to connect the Urbana PLATO system with the Minneapolis system. This link will be particularly useful in transferring programs, data bases, and "electronic mail." It represents the first step toward the creation of a network of PLATO systems.

II. Communications Hardware

The equipment used in the PLATO communications system can be understood by following the information flow from the terminal to the computer and back to the terminal. The eight bits of input data generated by a PLATO terminal[5] are identified by a two-bit code to specify one of four sources: 1) keys which may be pressed at the keyboard or generated electronically on computer command, 2) touch panel,[6] 3) external input devices, or 4) the terminal's own processing unit. The total data package transmitted to the Site Controller consists of 12 bits--a message start bit, the two bits identifying the source, 8 bits of data, and a parity bit. The Site Controller attaches a five-bit terminal identification code, adjusts the parity bit, and multiplexes the package, now 17 bits, onto the telephone line to the NIU. Upon receipt of the message the NIU attaches a five-bit site identification code. The NIU also examines the parity bit and converts this bit into an error indicator. If the message is error-free, this bit is set to "zero"; if an error is detected the bit is set to "one." The complete message package, now 21 bits in length (consisting of 5 bits specifying site, 5 bits specifying terminal, 2 bits specifying origin point within the terminal, 8 bits of data, and an error indicating bit), is presented to the computer for processing.

The Site Controller transmits 17-bit data packages to the NIU at 1260 bits/sec. Therefore, it can transmit (1260 bits/sec)/(17 bits) = 74 inputs per second from the 32 terminals at that site. This rate represents an average of (74 inputs/sec)/(32 terminals) = 2.3 inputs per second per terminal, which is almost five times the measured average of 0.5 inputs per second for PLATO users. Because the Site Controller multiplexes the data to the NIU on an asychronous (demand) basis, the burst rate per terminal can normally be equal to the output rate of the Site Controller (up to 74 inputs per second). To prevent loss of data during infrequent periods of high input activity occurring simultaneously at all terminals, a four-input buffer for each terminal is provided in the Site Controller. If all 32 terminals simultaneously made inputs, an input would be delayed in the Site Controller (32 terminals)(17 bits)/(1260 bits/sec) = 430 milliseconds (ms). The normal delay for a key passing through the Site Controller is (17 bits)/(1260 bits/sec) = 14 ms. The transmission time from the terminal to the Site Controller is (12 bits)/(1260 bits/sec) = 9.5 ms, and there is a control delay within the terminal of approximately 6.3 ms. Adding a delay time through the NIU of 0.2 ms gives an average total transmission time from key press to computer of 30 ms. (Add 0.005 ms per mile for remote terminals.)

A 21-bit output package is sent from the computer complex to each PLATO terminal every sixtieth of a second (1260 bits/sec). These packages can consist of commands (erase the screen, set to line, character, or dot mode, etc.) or data (character codes, line coordinates, etc.). Each package contains a message start bit, one bit to distinguish commands from data, one parity bit, and 18 bits containing the data or command. All bits zero indicate a "do-nothing" command, which is what the terminal receives whenever there is no new information being sent to that terminal. In line mode the 18 data bits contain 9 bits of X and 9 bits of Y to specify a line endpoint on the 512 × 512 dot plasma panel. This makes it possible to draw 60 connected lines per second. There are commands to change the terminal's X-register and Y-register if screen repositioning is required. In character mode the 18 data bits contain three 6-bit character codes. The terminal contains a character generator (8 × 16 grid) whose memory is organized as four 63-character groups, two of which contain fixed character patterns (standard alphanumerics), while the other two groups contain character patterns which can be loaded from the computer. Thus, central software can design and distribute alphabets (e.g., Cyrillic or Arabic) or simply pieces of pictures that can be displayed at high speeds by writing "text" on the plasma panel. Normal text consists of long strings of numbers, standard punctuation, and lower-case letters, all of which are in the first memory group, so that text can be sent to the terminal at a rate of nearly 180 characters per second (sixty 21-bit packages per second, each containing three 6-bit character codes). An "uncover" code (the 64th character in every character group) followed by a character memory group code is used to switch from one group to another. Note that standard alphanumeric terminals normally use a 10-bit package for transmitting a single character, which yields only 120 characters/sec on a 1200 bit/sec line.

In addition to line and character graphics, there are commands to select a microfiche image or audio message[7] or to send data to any kind of digital device attached to the terminal.

Output data packages destined for the terminals are passed from the computer to the NIU. Contained within the NIU are two memories, each containing 1008 twenty-bit words. Each memory can hold one output package for each terminal connected to this NIU. In an interval of a sixtieth of a second, the NIU transmits the entire contents of one memory onto a television channel. Concurrently the computer is loading the other memory with new data. At the end of each time interval a switching action occurs and the contents of one memory are delivered to the television channel while the other memory accepts new data from the computer. This alternating action results in a data package being transmitted to each of the 1008 terminals every sixtieth of a second. (A data package is actually 21 bits; but because the message start bit is always a "one," it is not stored in or transmitted by the NIU but instead is attached to a data package at the Site Controller.) During these time intervals the NIU automatically generates and transmits an all "zeros" code to those terminals which have no computer output. Although this all "zeros" code is interpreted as a "do-nothing" operation at the terminal, it does automatically perform modem resynchronization. (Since the plasma panel has inherent memory, only graphics changes need be transmitted to the terminal.)

In the transmission of data from the NIU to the television channel, the first bit for terminal 0 is transmitted followed by the first bit for terminal 1, then the first bit for terminal 2, etc., until the first bit for terminal 1007 has been transmitted. The second bit for all terminals is then transmitted, followed by the third bit for all terminals and so on until all 20 bits have been transmitted. The 20 bits for each terminal are thus spaced evenly through the sixtieth-second time interval with adjacent data bits for a particular terminal being separated by 1007 bits for the other terminals in the system.

With this data format, the Site Controller need only select and pass data through the terminal; it does not have to store data arriving at a high rate and forward the data at a low rate to the terminal. This simplifies the design and operation of the Site Controller. Moreover, a short burst of noise on the megabit channel will alter one bit for several terminals, not several bits for one terminal. Thus, a single-bit parity check by the terminal can detect nearly all transmission errors. More will be said later about the detection and correction of errors.

In addition to handling output data, the NIU generates and combines standard television horizontal and vertical synchronization and blanking pulses with the digital data, so that the resulting composite signal is completely compatible with standard cable or broadcast television transmission equipment. Figure 2 is a brief time exposure of this signal as displayed on a television monitor. An active terminal's 20-bit package appears as a vertical column of dots spaced every 12 horizontal scan lines over 240 scan lines, transmitted every sixtieth of a second. (Because television interlaces two sixtieth-second frames to make 30 complete pictures every second, alternate frames are displaced vertically by one-half scan line.) The 1.2 million bits/sec data rate in this television channel is a conservative use of the bandwidth.

The horizontal and vertical television synchronization pulses transmitted by the NIU synchronize the Site Controllers. The Site Controller extracts its terminal data by timing signals derived from these synchronization pulses. The Site Controller attaches a start bit to the data package so that the terminal actually receives a 21-bit package sixty times per second (1260 bits/sec).

Output data packages incur a sixtieth of a second delay passing through the NIU and another sixtieth of a second delay in transmission to and accumulation in the terminal. The Site Controller adds an additional delay of 0.8 millisecond (ms). The terminal requires approximately 2.5 ms to plot the character on the panel. The total hardware delay time from computer output

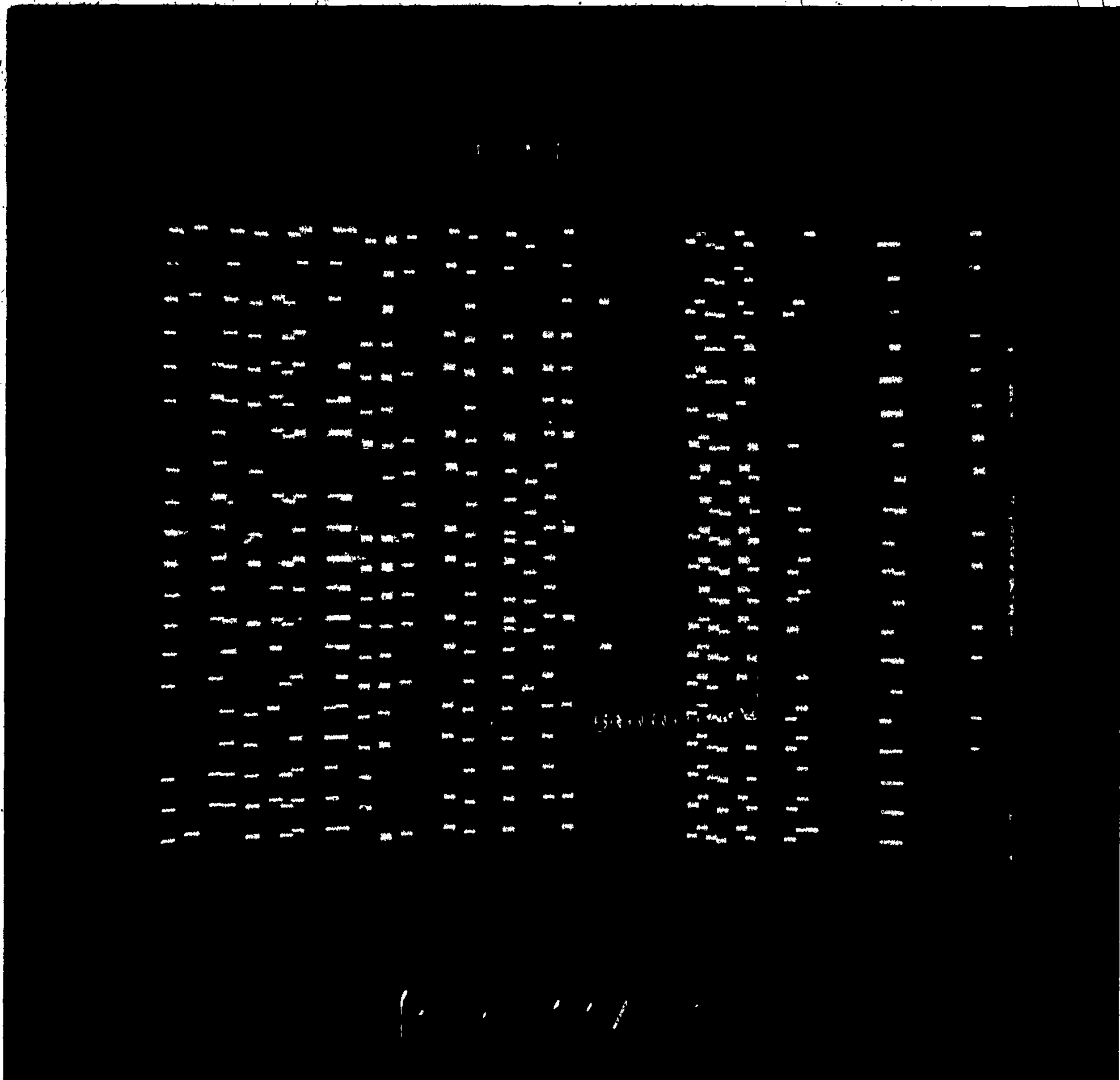

Figure 2. Photograph of NIU television output.

to appearance on the plasma panel is, therefore, 37 ms. Adding to this figure the average input transmission time, the total delay between a key-press and the appearance of a character on the screen due to the PLATO communications hardware is 37 ms + 30 ms = 67 ms. (Distantly located terminals may incur an additional 12.5 to 90 ms delay due to use of long distance modems, plus 0.01 ms/mile round-trip transmission time.)

It is important to emphasize that the PLATO output scheme guarantees that each and every user can receive output at the maximum rate at all times. There is an open 1260-bit/sec "pipe" running directly from the computer to each terminal. There are no shared "store-and-forward" buffers interposed between computer and terminal, and there are no hand-shaking protocols between nodes of the communications network. Not only is the terminal user guaranteed full output, free of jerkiness, but the computer is guaranteed that it can simply pour data out to the terminal without worrying over the possibility of overflowing some buffer somewhere in the communications network.

Measurements show that on the average, twenty percent of the active terminals receive data in each sixtieth-second interval, but statistical fluctuations in this load do not cause output delays, since each terminal always receives 1260 bits/sec. Tighter packing of the data to handle more terminals would require terminal identification bits to be transmitted and would require additional error check bits due to the increased sensitivity to noise bursts. These additional bits would cancel much of the hoped-for gain, and the Site Controllers would have to be considerably more complex as well.

However, our operational experience shows that even though most of the attached terminals are used every day, no more than 50% of them are ever active simultaneously. The NIU, therefore, is seldom required to service during any sixtieth-second interval much more than 100 terminals (20% × 50% × 1000). Although there are correlations from one interval to the next, the large number of independent terminals makes the distribution approximately Poisson around the mean of 100: the standard deviation of the fluctuations is, therefore, $\sqrt{100} = 10$. On the basis of these usage patterns, it is clear that it would be exceedingly rare to observe as many as 200 terminals receiving data in a sixtieth-second interval. In a redesign of the distribution system one might, therefore, try to handle several thousand installed terminals with a single television channel.

III. Communications Software

Before describing the communications software, it is helpful to describe briefly the computer system architecture, since it is quite unconventional. In Figure 3 it can be seen that the Control Data Corporation "Extended Core Storage" (ECS) is the heart of the system: ECS has ties to central memory (CM), and thus to the central processing units (CPUs), and to the peripheral processing units (PPUs). ECS has an access time of less than 5 microseconds, which is typical of inexpensive slow core memories, but the ECS transfer rate is ten million 60-bit words per second! This enormous transfer rate is achieved through parallel and closely-phased memory operations. Basically, programs and data in ECS are swapped in and out of central memory for processing by the CPUs. This is far more efficient than swapping interactive jobs between central memory and electromechanical disk or drum memory because of the two orders of magnitude improvement in transfer rate and three to four orders of magnitude improvement in access time. Disk drives furnish permanent storage of programs and data but play no role in the swapping process once this material has been moved to ECS.

The ten peripheral processing units (PPUs) shown in Figure 3 are mini-computers used to handle input and output. In particular, one PPU handles input from the NIU and passes output to the NIU. Like the CPUs, the PPUs have direct access to ECS. Figure 4 illustrates the relationship of the input and output PPU to the NIU. The input section of the PPU uses the 10-bit terminal identification number to store the 10-bit input data in a terminal-specific location in ECS. The PPU also sets a bit in an "input polling list" in central memory to notify the CPU that this user requires processing. (If the input error indication bit is set, the PPU merely tells the CPU to increment a count of input errors for this site.)

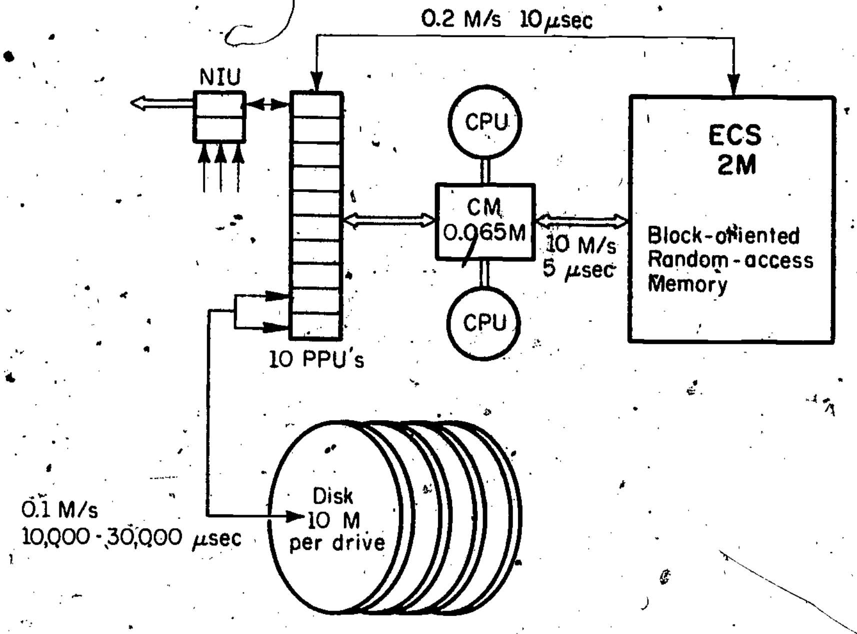

Figure 3. PLATO IV computer architecture, showing memory sizes in 60-bit words, transfer rates in 60-bit words/sec, and access times in microseconds. M = million, CPU = central processing unit, PPU = peripheral processing unit, NIU = network interface unit, CM = central memory, ECS = Extended Core Storage. Programs and data are swapped between CM and ECS. Conventional disk drives provide permanent storage for programs and data. The basic computer is a Control Data Corporation Cyber 73-24.

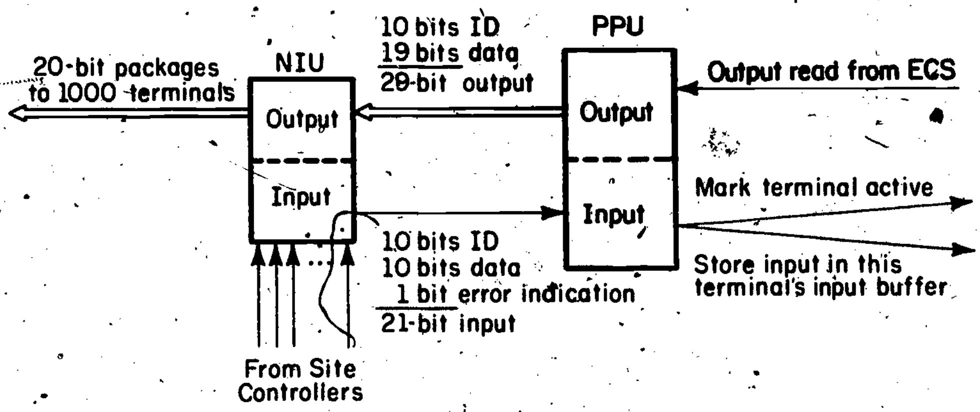

Figure 4. Role of the input and output PPU interfacing to the NIU. The 10-bit terminal identification (ID) passed to the output section of the NIU determines the time position of the associated data in the television format.

Every sixtieth of a second the output section of the PPU reads from ECS a list of 29-bit packages, each package consisting of ten bits of terminal address plus the nineteen bits of data destined for that terminal. The PPU passes each package to the NIU, which generates a parity bit to go with the 19 bits of data and stores the resulting 20-bit quantity in the specified slot in its initially empty memory. The list of data plus the zeroes contained in the other NIU slots make up one complete "frame" of television output.

Corresponding to the 1008 terminals serviced by an NIU there are 1008 "station banks" in ECS. Each station bank contains a buffer to receive the input data from the NIU. The station bank also contains information concerning the current state of the terminal, including pointers to relevant data bases (see Figure 5). The station banks are in fixed locations in ECS, but the other blocks shown in Figure 5 are referenced through the station bank pointers. The PLATO software keeps track in the station bank of the complete current state of the terminal: mode (including whether in character or line mode), current settings of the terminal's X and Y screen position registers, current character memory group setting, etc. More precisely, this information describes the state the terminal will be in upon completion of transmission to the terminal of all pending output (which may be ten seconds in the future). This information is frequently utilized by PLATO users: for example, to specify a new screen position relative to the present screen position.

Approximately twenty times a second a CPU looks through the central memory input polling list for terminals marked active by the PPU. (Actually, the PPU-generated polling list is merged with a CPU-generated list of terminals requiring processing due to an earlier interrupt, disk I/O delay, timing return, etc.) The process of looking through this list

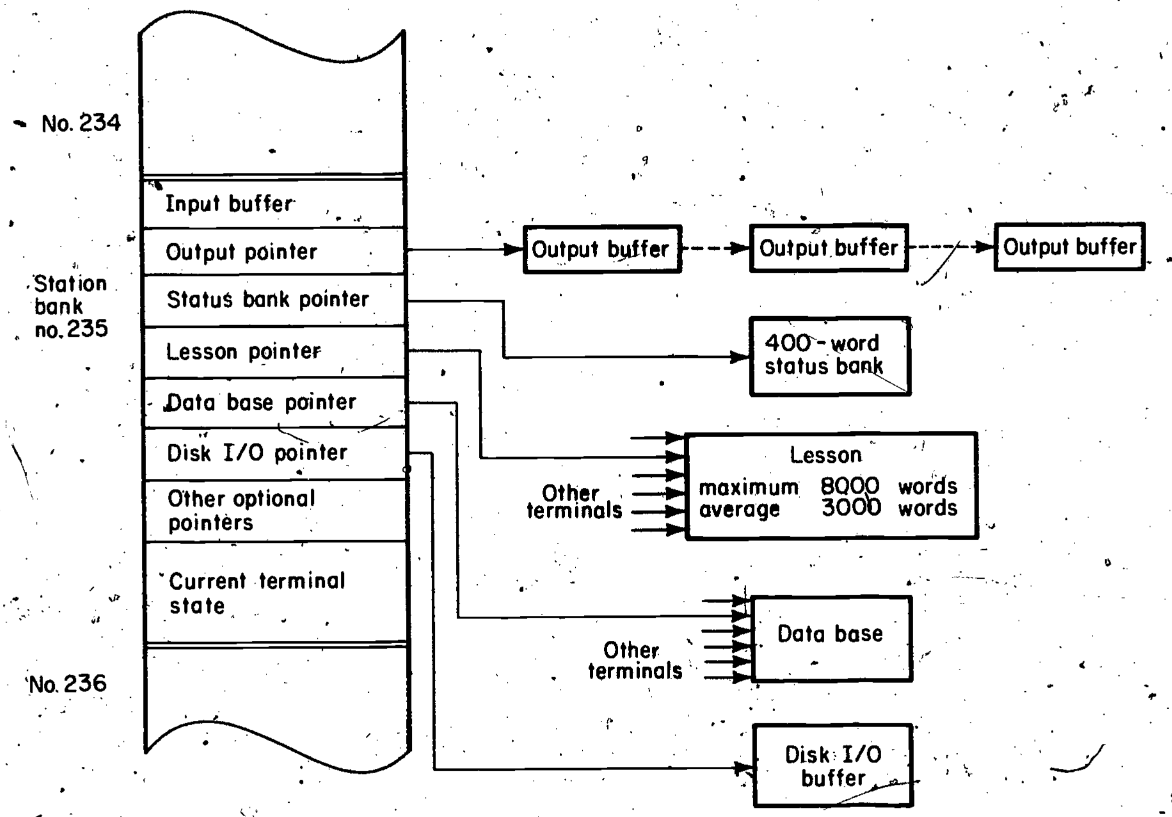

Figure 5. Relationship of fixed-location station bank to associated swapping-memory space allocations. Dotted lines lead to optional block allocations. Many terminals can share a single copy of a lesson or a data base, but other blocks are not shared. Station bank information on current terminal state includes screen X and Y, terminal mode, character memory group, etc.

is very rapid: active terminals in a site are represented by bits set in a floating-point word, and a "normalize" instruction is used to skip over inactive terminals and generate an address of an active terminal. The site number is encoded in the exponent of the floating-point word. The CPU processes the inputs for all terminals marked in the list, then either starts through the list again or, if less than a twentieth of a second was consumed, may perform lower-priority "background" tasks in the time left over.

The CPU begins processing a terminal by using the generated address to read the station bank from ECS into central memory. The PPU had stored the 10-bit input data in the station bank buffer. This input plus the complete program status contained in a 400-word "status bank" determines whether simple key echoing is required or whether the algorithms of the user's particular "lesson" (program) are needed to process the input. (About one-half the keys do require the lesson for processing.) Note that the separation of the individual status banks from lessons makes it possible for a single copy of a lesson to serve many terminals. As is indicated in Figure 5, both lessons and data bases may be shared by many terminals.

In either the key-echoing or algorithm-executing case, the CPU normally generates terminal output and alters information in the station and status banks. Output is generated in a standardized form which is independent of the state of the terminal. This output is stored in ECS where it can be read by a separate CPU program, the "formatter." The formatter grinds through the standardized output from the job to produce the final output for the target terminal. Maximum use is made of the current terminal state specified in the station bank in order to optimize the utilization of bandwidth to the terminals. A simple example is the avoidance of specifying the line mode if the terminal is already in line mode. A command to set the Y-coordinate is skipped if the terminal's Y-register will already contain that value. If the terminal is currently in character mode and using the character memory group which contains capital letters, no character group change code need be sent in order to transmit a capital letter to the plasma display panel. At the end of the formatting process, the station bank contains the terminal state that will exist when all pending output has reached the terminal.

Output can be a single character or can be an entire page of text, a graph, or an animated cartoon, some of which can take five or ten seconds to transmit to the terminal at 1260 bits/sec. These data are stored in ECS and dealt out to the NIU every sixtieth of a second. As the formatter generates output, it adds the output to frame buffers in ECS. There are 128 frames held in ECS, constituting a circular buffer of just over two seconds duration. Every sixtieth of a second the PPU merely transfers a new frame from ECS to the NIU. If a terminal has more output than will fit into the 128 frame buffers, the formatter stores output in a chain-linked string of buffers associated with that terminal's station bank. (Actually, all output, not just the overflow, is stored this way in order to facilitate transmission error correction, as will be discussed later.) The station bank points to a 32-word output buffer for formatted output data; 31 words are used to hold 31/60 of a second of output (each 60-bit word holds one 19-bit package) and the 32nd word contains status including a pointer to additional 32-word output buffers drawn from a system-wide pool as needed. (See Figure 5.)

If there is overflow, due to having more than 128 frames of output pending for the terminal, the formatter makes a note in a first-in-first-out list to process this terminal again after 1.2 seconds. Since 1.2 seconds is considerably less than 128 frames (= 2.13 seconds), the formatter can insure that the frame buffers will be refilled in time to prevent interruptions in the output. The formatter is called up on every poll, which occurs roughly every twentieth of a second. It need look only at the first, oldest entry in the list to determine whether any terminals now must have additional output moved to the frame buffers. Once generated, output will appear smoothly on the terminal screen, without breaks or hesitations. This is particularly important for producing moving, animated displays. (Breaks in the output will, of course, occur if the CPU does not generate output fast enough; but once the output has been generated, there will be no interruptions due to frame manipulations.)

Frame transmission is initiated when the PPU receives an indication that the NIU has transmitted one of its 1008-word memories and has switched to the other memory, so that the basic timing of output activities is actually provided by the NIU.

Note that there are four autonomous phases during the processing of a job: a PPU stores the input and marks the input polling list; a CPU processes the input and produces standardized output; the CPU formatter generates formatted output; and the PPU passes the next frame of output to the NIU every sixtieth of a second. The two CPU processes run as separate jobs, each with its own separate central memory area, but they share ECS which serves as a communications link between these autonomous processes. One PPU handles both input and output by intermixing output activities with input activities.

IV. Error Correction

Detection and correction of output transmission errors involves both hardware and software. The PLATO terminal contains a 7-bit "word count register" to count the 21-bit output packages as they arrive (ignoring "do-nothing" commands). The word count register is reset at appropriate times so the system can associate a word count with a particular 21-bit package. When a terminal detects a parity error, it sends a special input to the CPU containing an error flag and the present contents of the word count register: this enables the system to retransmit the erroneous data. Moreover, the terminal refuses to process the erroneous data package or any following data until it receives a special restart command. This action eliminates the situation where, for example, the erroneous data would have changed the terminal from character to line mode and the terminal would have interpreted the following (character) data as line endpoints!

Upon receiving an error message, the CPU adds "do-nothing" outputs for this terminal at the ends of the appropriate frames, which cancels any outputs by overwriting the NIU memory. The CPU then sends a restart command and an echo command instructing the terminal to send a special input acknowledging receipt of the restart command. If no response is received from the terminal within a second, the CPU repeats the sequence. After several retries without a response, the computer gives up and relies on the terminal user to clear the error state manually.

Once the terminal has been restarted, the CPU error correction software moves data from the output buffers associated with the station bank into the frame buffers, so that the terminal will again receive the data it previously ignored. A detected and corrected error is generally observed at the terminal as a fractional-second hesitation in the normally smooth output appearing on the screen.

While the average response time is 125 milliseconds, there could in rare cases be long delays due to the holding of input in a Site Controller when all terminals make simultaneous inputs. For this reason, the CPU does not release one of the 32-word output buffers until the buffer is one second old. This insures that the computer retains at all times the last second of output, making retransmission possible in case of a detected error.

Note that all this elaborate "hand-shaking" between computer and terminal is necessary only in the case of transmission error. Normally the software can assume that the terminal is always ready to receive data and need not query the terminal to determinal its state.

Appendix

A common criticism of the PLATO IV communications scheme is that processing and transmission costs could be reduced by having the terminal or a nearby minicomputer handle the routine echoing of keys, thus sending to the central computer only those inputs which require more elaborate processing. There are several reasons why this criticism is not valid in the PLATO educational environment.

First, by actual measurement fully half of all keys pressed do not involve mere echoing but require the use of the CPU, processing the particular user's lesson, to generate the resulting output display. This is partly due to the highly interactive nature of PLATO, whose input rates, output rates, and response speeds far exceed standard time-sharing systems. It is also related to the environment in which student responses usually consist of a few keys and where a single key may be used to request additional help or to proceed from one informational display to another. This is in contrast to an environment in which terminals are used mainly for entering long computer program statements and CPU activity is required only on every "end-of-line" key, which might be every tenth key on the average.

Second, normal key echoing on a PLATO terminal is in itself an elaborate process requiring a large scientific computer to do a satisfactory job of fully implementing the complex typography needed for education, including the use of superscripts, subscripts, backspaces, underlines, accent marks, a modifiable character set, etc. Selectively erasing the appropriate last character or word on the screen is a major task given these possibilities, and requires that the CPU know the present state of the terminal and its display.

The two-dimensional nature of the plasma panel and its selective erase capabilities lead to different demands from those of an alphanumeric terminal, which is essentially a linear device.

Third, in a situation where a relatively small fraction of the keys are to be processed locally by a minicomputer, but the rest are to be sent on to the CPU, there would be a high cost in processing and transmission associated with the necessarily complicated handshaking between the two processors. Not only must the CPU adequately inform the minicomputer as to the user's current status, but the minicomputer must make a complex decision from that status and the next input to determine whether to transmit or echo. This is precisely the kind of decision only the CPU can make because only it has access to the complete user status coupled with adequate processing power to act on that status. All keys must eventually be sent to the CPU, so local echoing does not decrease the input bandwidth requirements, and frequent short transmissions lead to fewer queuing difficulties.

Fourth, actual measurement has shown that less than five percent of the total work load of the central processor is devoted to simple key echoing. Thus, even totally neglecting the overhead that would be incurred, the implementation of a scheme for local echoing of keys could not possibly produce any significant time savings.

For these reasons, in the context of the PLATO system the use of the CPU for echoing is simpler, probably cheaper, and certainly more satisfactory than local echoing.

References

1) For a general discussion of the PLATO IV system, see D. L. Bitzer, B. A. Sherwood, and P. Tenczar, "Computer-based Science Education," CERL Report X-37 (1972), reprinted in "New Trends in the Utilization of Educational Technology for Science Education," UNESCO, Paris (1974), in "Computers in the Instructional Process; Report of an International School," edited by K. L. Zinn, M. Refice, and A. Romano, EXTEND Publications, Box 407, Ann Arbor, Michigan (1974), for sale for $8, and in "Federal Information Systems Plans--Federal Use and Development of Advanced Information Technology (Part 1)," U. S. Government Printing Office, Washington, D. C., 20402 (1973), for sale as stock number 5270-01934, $1.85.

2) B. A. Sherwood, "The TUTOR Language," CERL (1974). P. Tenczar, "TUTOR Graphic Capabilities," Society for Information Display 1974 International Symposium, p. 70.

3) J. Stifle, "PLATO IV Architecture," CERL Report X-20 (1973). Site Controllers are manufactured by the Magnavox Corporation, Fort Wayne, Indiana.

4) J. Stifle, D. L. Bitzer, and M. Johnson, "Digital Data Transmission via CATV," CERL Report X-26 (1972), and P. Tucker, "A Large-Scale Computer Terminal Output Controller," CERL Report X-27 (1971). Network Interface Units are also manufactured by Magnavox.

5) J. Stifle, "The PLATO IV Student Terminal," Proceedings of the Society for Information Display 13, 35 (1972). PLATO terminals are manufactured by Magnavox and incorporate the Digivue plasma panel manufactured by Owens-Illinois, Toledo, Ohio. For a description of the plasma panel, see R. L. Johnson, D. L. Bitzer, and H. G. Slottow, "The Device Characteristics of the Plasma Display Element," IEEE Transactions on Electron Devices 18, 642 (1971).

6) F. A. Ebeling, R. S. Goldhor, and R. L. Johnson, "A Scanned Infrared Light Beam Touch Entry System," Society for Information Display 1972 International Symposium, p. 134.

7) D. L. Bitzer, R. L. Johnson, and D. Skaperdas, "A Digitally Addressable Random-Access Image Selector and Random-Access Audio System," CERL Report X-13 (1970).