The CRT display subsystem of the IBM 1500 instructional system

by R. H. TERLET
International Business Machines Corporation
Los Gatos, California

INTRODUCTION

The IBM 1500 Instructional System is an experimental system for computer-assisted instruction, designed to administer individual programmed lessons to 32 students at once. Working through one or more teaching devices at his own instructional station, a student may follow a course quite different from, and independent of, lessons presented at other stations. Instructional programs stored in central files control lesson content, sequence, timing, and audio-visual medium, varying all of these according to the student's responses.

Briefly, the system works like this: The processor retrieves instructional material from the files and presents it on a station input/output device. The student responds as directed. The processor then compares his response with the answers anticipated in the instructional program and continues with the next lesson material or branches to remedial instruction. The system can keep records of student answers, response times, and accuracy.

The IBM 1500 Instructional System is shown in Fig. 1a. The central processor, an IBM 1131, has access to core storage of 32,768 sixteen-bit words and a cycle time of 3.6 or 2.2 microseconds. Under direction of the 1500 Operating System, the processor controls the time sharing of the student stations (Fig. 1b) and the execution of the instructional programs. The IBM 1502 Station Control multiplexes the input and output of the station devices, which may include an input/output typewriter, an image projector, and an audio play/record unit, as well as the CRT display to be discussed. Each CRT display unit includes a keyboard for student responses, and may include a light pen as well. The 32 CRT display units, the station control and the processor core storage together make up the display subsystem.

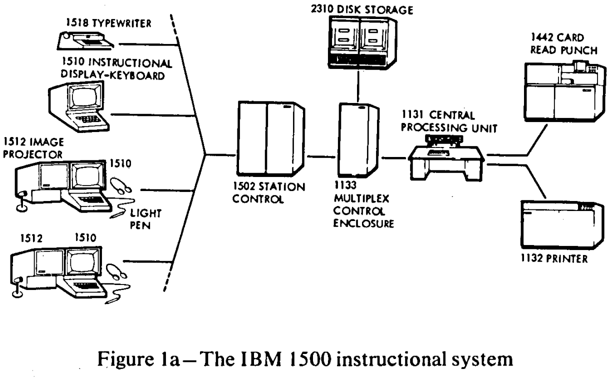


Figure 1a—The IBM 1500 instructional system

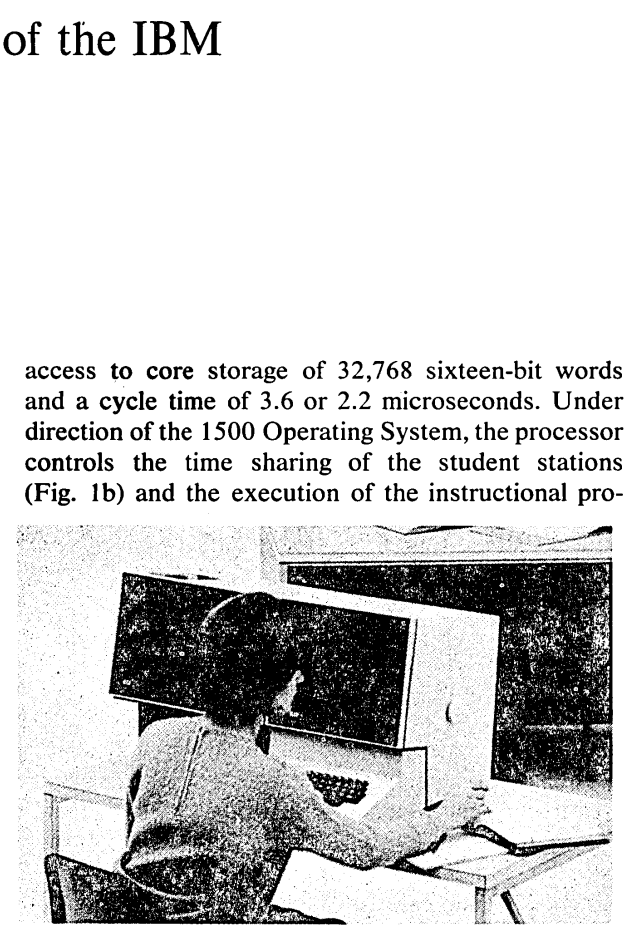


Figure 1b—The IBM 1510 Student station

CRT speed and flexibility make this type of display unit attractive as a basic instructional device. In some ways, however, the display requirements for instruction are more demanding than those for conventional data processing applications.

A character set for instruction must often include a far larger number of characters and symbols than is needed for other applications. For example, teaching a foreign language may require that two different alphabets be displayed at once. In mathematics, a display must often include exponents, subscripts, and fraction lines, as well as alphanumeric characters and mathematical symbols.

An instructional display must also be unusually flexible. A student should be able to complete displayed sentences and to insert words within a sentence. It should be possible to display combinations of simple images and printed text in teaching certain concepts. To meet the needs of different student stations and achieve a stimulating variety in lesson presentation, character fonts should be easily changed.

Finally, to give course authors freedom in varying the mode of lesson presentation, the CRT displays should be in a form compatible with the alternative typewriter printout.

These system objectives were achieved in a versatile display system capable of handling 32 student displays. The CRT display units are of conventional design,[1] with a magnetic disk buffer to store and refresh the images. Characters and images are positioned under program control. The system is able to handle any number of large character sets by means of program-changeable fonts ("dictionaries") placed directly in core storage, where they are accessible to the character generation logic and to the system program. Allocation of space for the fonts reduced the available core storage, already a good deal smaller than is common in multiterminal time-sharing systems. The problem of satisfactorily sharing the relatively small core area remaining was solved by the application of data chaining techniques in core storage. Here the high interrupt servicing overhead usually needed to chain blocks of data was avoided by making such chaining a hardware operation. These solutions depended on a flexible manipulation of core storage made possible by the storage access channel feature of the processor.

Hardware description

The major components of the Display Subsystem, shown in Figure 2, are (1) the CRT display, keyboard, and light pen at the student station; (2) the display control and light pen adapter of the station control; and (3) the core storage of the processor.

Lesson material called from the disk storage is interpreted by the Operating System and, under the direction of the display control logic, is translated from a stream of computer-coded characters and symbols into a sequence of the appropriate displayable dot patterns. These patterns are obtained from the dictionaries which occupy a portion of the core storage. As they are translated, the dot patterns are temporarily stored, a line at a time, in the line buffer, before transfer to the video buffer. The video buffer is a magnetic disk assembly in which a separate recording track, with an associated fixed read/write head, is assigned to each display unit. This track can store one complete frame of text and image material, already coded in the video dot pattern, which can be read again and again to maintain a continuous display.

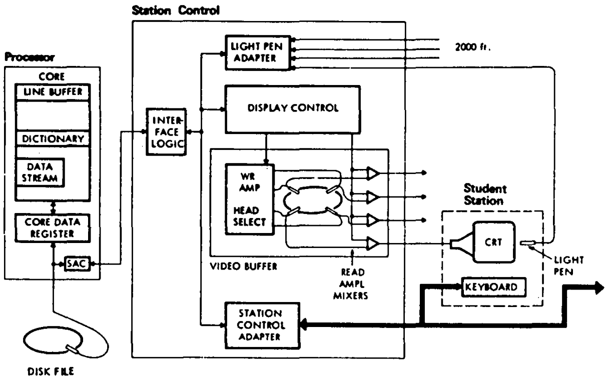


Figure 2 – The IBM 1500 display subsystem

The display control logic also generates timing signals that control the vertical and horizontal synchronization of the CRT displays and, during student light pen responses, serve to identify the pen position for the light-pen adapter.

The display unit consists of the CRT display, the keyboard, and the light pen. The CRT display presents instructional text and images to the students. The keyboard is the major input device for student entries, which are in most cases immediately displayed on the CRT under program control. The light pen is an optional input device with which the student can point to selected response areas on the face of the CRT.

The general characteristics of the CRT display are summarized in the table. The display area on the screen is 4.8 inches high and 8 inches wide. The control logic divides this area into 40 columns and 32 horizontal half-lines.

Each character is based on an 8 by 12 dot matrix that is one column wide and two half-lines high. The system logic does not allow for space between columns or half-lines; thus dot patterns can be joined to form continuous lines, while needed space can be written into the display code.

The display electronics are of conventional television design, with some special attention to linearity of the vertical and horizontal sweep circuits.

The video buffer consists of a specially assembled pack of six IBM 2316 disks, with 32 magnetic heads,

Horizontal Visible Dots	320
Vertical Visible Lines	192
Horizontal Frequency	6.5 kHz
Video Digital Frequency	2.5×10^6 pulses per second
Vertical Frame Rate	30 Hz
Interlace Ratio	2/1
Visible Dots	61,440
Character Matrix	8 x 12 dots
Display Size	8 x 4.8 inches
Geometric Distortion (Initial Settings)	Vertical Line ± 0.06 inch Horizontal Line ± 0.01 inch
Linearity (Deviation from Nominal)	Center Third: ± 15% Outside: ± 25%

General characteristics of the IBM 1500 instructional display unit

in four assemblies of eight each, mounted around the disks as shown in Figure 3. The heads are fixed in position, and each one reads and writes a specific track assigned to a given student display. Thirty-two tracks on the outer edges of the 14-inch disks are chosen to give longer and approximately equal track lengths.

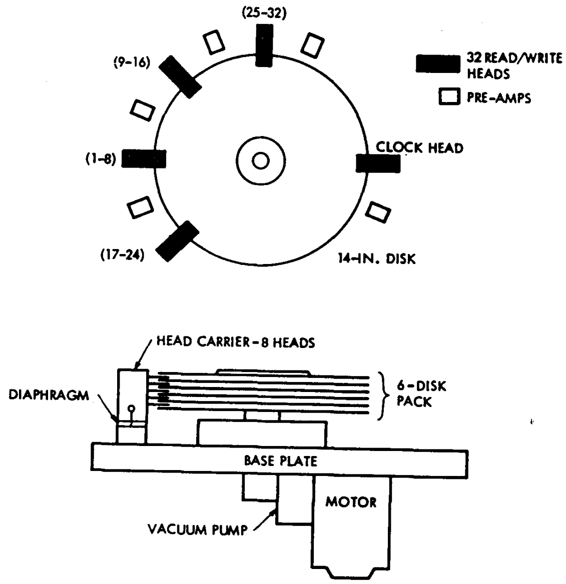


Figure 3a—The video buffer, top and side view

The video buffer stores and regenerates the images displayed on the CRT. The information on the video buffer tracks is recorded in the non-return-to-zero (NRZ) mode. The digital recording is a one-to-one image of the dot patterns being displayed on each instructional unit.

The disks rotate at 1800 rpm, and one revolution of the disk corresponds to one frame on the display. The CRT scanning is synchronized with the rotation of the video buffer disk pack, so that the format of the data on the track corresponds to the scanning of the face of the CRT, as shown in Figure 4. Since each frame of the display consists of two interlaced fields, half of the data track contains the odd scan field data, while the other half contains the even scan field data.

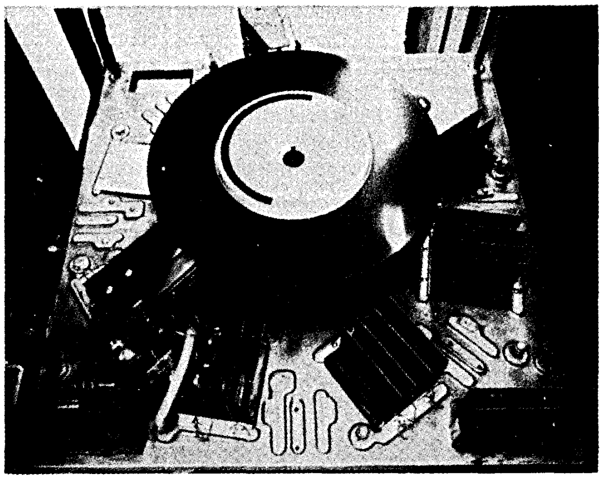


Figure 3b—The video buffer (photograph)

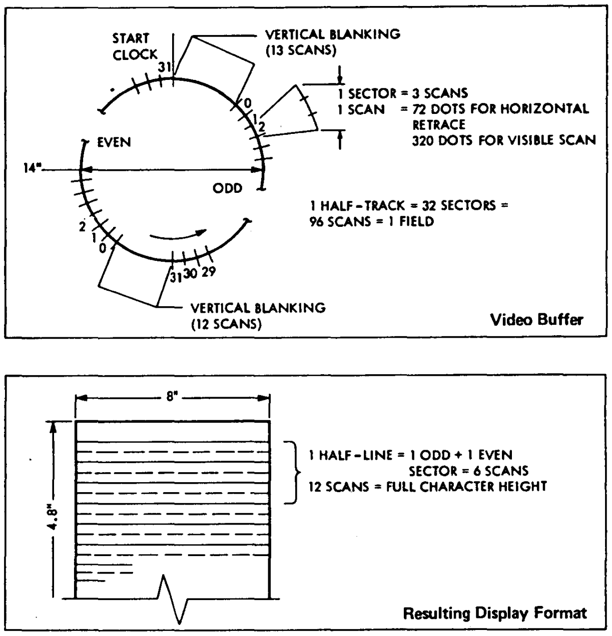


Figure 4—Data format on video buffer and resulting display format

The 192 visible scan lines in a frame are addressable in 32 half-line segments of six scans each (automatically divided into three even and three odd scans, which are stored on different halves of the data track). A complete line of characters then represents 12 scans on the CRT, six in the odd field, and six in the even field. Each half-track, or field, consists of 96 visible scans, plus 12 scans for the vertical retrace in the odd field, and 13 scans for the vertical retrace in the even field—the extra scan here allows the field interlace.

Each scan line in the video buffer consists of 392 dots (or bits). The first 72 are always blank to allow for horizontal flyback time; the remaining 320 dots correspond to the 40 columns of displayable data in a horizontal sweep of the CRT.

In addition to the data tracks for each display unit, the video buffer contains a separate timing track read by its own fixed clock head. The timing track stores a steady sequence of clock pulses with a blank home gap that serves as a reference for display sync signals and disk read/write functions. The clock pulses increment digital counters, whose decoded output supplies signals that control display blanking and vertical and horizontal sync. Since the display data in the video buffer is written on the disk under control of the same data clock, the buffer output and CRT display are thus properly synchronized.

The video buffer contains the necessary electronics for head selection, writing, and sync mixing. Each head is in a read (or display) mode except when writing, and a single write driver serves all the heads.

The display control logic circuits are time shared for video buffer recording and character generation.

A single command to the video buffer control logic can change any track (1) by erasing (i.e., rewriting with blanks) the complete track, or (2) by rewriting or erasing:

(a) a full line (12 scans),
(b) a half-line (six scans),
(c) a word or phrase (full or half-height) within a line.

Insertion of single words within a line is simplified by our choice of the NRZ mode of digital recording, with a binary magnetic state directly corresponding to black and white dots on the display. The write circuits and control logic were designed to insert short dot patterns on successive segments of a track without disturbing previously recorded data.

The display unit uses a standard IBM keyboard. The station control is an I/O device multiplexer operated under program control of the processor. At regular intervals the adapter polls the I/O devices and reads in any keyboard input. Under I/O servicing program control, the student's input message is assembled and the character he keyed is displayed on his CRT. In this way the computer program positions the displayed student response and selects the character font in which it is displayed. This means that a keyboard character set can be changed by the program. The student keyboard itself could be changed by an overlay.

The subsystem response time to keyboard inputs is a function of the length of the queue to the video buffer. The length of the queue in turn is related to the input keying rate of all student stations. The response time then can vary from 20 msec to a worst case of 1.07 sec, 200 msec being typical.

Character generation

The Display Control logic basically adds two special instructions, "translate" and "transfer," to the repertoire of the processor. A sequence of translate and transfer instructions records a new display frame on the video buffer, ready for immediate continuous display.

We can trace this process by looking again at Figure 2. The character generation logic operates on a string of characters (the data stream) in core storage. It translates the computer-coded characters into video-coded dot matrices by a table lookup technique—finding the proper dot image in the designated dictionary, always at hand in core storage. The character generation hardware then stores the translated video code in a reserved area in core: the line buffer. The line buffer holds one full line (12 scans) of video information.

When the CPU program gives a transfer command, the line buffer contents are recorded on the selected track and sector of the video buffer. The execution of the transfer command automatically clears the line buffer on readout.

The data stream may be of any length, but is always organized in blocks of 32 sixteen-bit words. These blocks are chained by familiar list processing techniques; that is, the last word in the block is the address of the next block. In this case, though, programming trouble and processing time were saved by "wiring in" the repetitive subroutine necessary to link the blocks. No programming intervention is needed to determine the new address; the logic automatically reads in the address and continues the processing of the data stream from the new core location.

Each dictionary occupies 768 consecutive core locations. The number of dictionaries in the system is a user option. Location of the dictionaries directly in core, readily accessible to the character generation logic, was a choice dictated by the need for flexibility in changing fonts in an instruction system. Different dictionaries can be used in one line of text. The dictionaries can be loaded, altered, or switched directly by program.

We could afford to use a fairly large proportion of the relatively small available core storage for this important function because list processing of the data saved space. And we could afford to use list processing in the data stream of a multidevice time-sharing system because automatic data chaining by the hardware saved time.

The line buffer is a fixed area of core, capable of storing 240 words, It too can be directly loaded by program.

The character shapes are specified by the user when he constructs a dictionary. Each character or image pattern takes six words of storage, enough to describe an 8 by 12 dot matrix. Characters are usually constructed in 7 by 10 matrices, leaving one blank space between characters and two blank scans between lines on the display. The 7 by 10 matrices can adequately describe many kinds of characters sets, including upper and lower case alphabets. For graphics or very large characters, basic patterns are usually specified in 8 by 12 matrices which can be combined to form larger patterns.

Characters are generated in a "cycle steal" operation. The display control logic "steals" a cycle from the core storage through the storage access channel. This feature gives the display control random access to all core areas needed for character generation. It was decided to let this operation take all the processor memory cycles it needs until the whole data stream is completely translated. This had the effect of reducing appreciably the programming interrupt overhead associated with the displays. Storage utilization is the same as if the processing was overlapped, but no interrupt servicing is required because none of the processor registers are changed at the end of the execution. File operations were given a higher "cycle steal" priority so that system performance did not suffer.

Each code of the data stream is scanned in succession. A character code is translated and the appropriate bit pattern stored in the line buffer; a function code will usually modify a control counter. Translation relies on one dictionary until a dictionary change code appears in the data stream.

Translation stops at this point, and the program can intervene to specify the next dictionary address. This program intervention allows flexible assignment of dictionary areas in core storage and relieves the course author of the problem of addressing dictionaries in a multi-user environment.

A character is translated as shown in Figure 5. After initialization of the display control hardware, one data stream code is read in (1). The code is combined with the dictionary address to access the dictionary (2,3). The character generation hardware reads in the 16-bit dictionary word it finds (4), writes the eight high-order bits in the line buffer (5), and increments the line buffer address counter one scan position (6). Next it stores the eight low-order bits in the line buffer, and again advances the line buffer address counter.

For each character, the dictionary is accessed six times and 12 partial scan lines are stored in the line buffer. At the end of this translation the line buffer address counter points to the next character column.

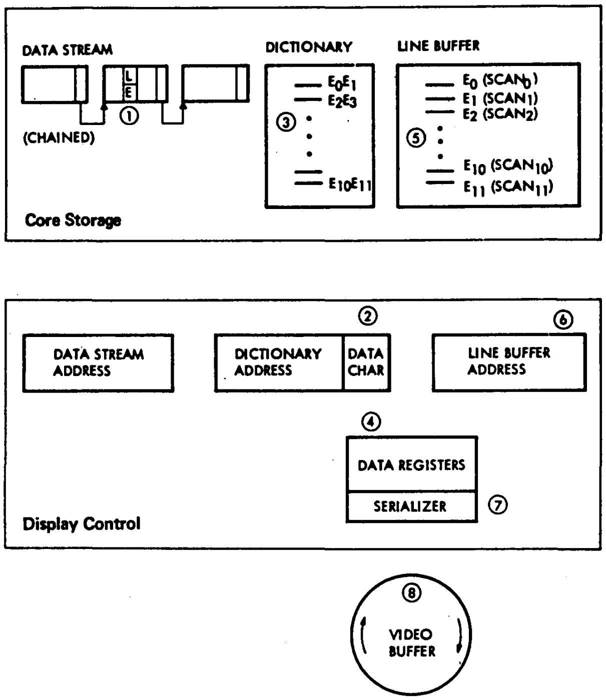


Figure 5 – Character translation

A transfer command causes the information in the line buffer (5) to be read by the display control logic (4), serialized (7), and written in the designated track and sector of the video buffer (8).

A transfer command may result in transfer of a full line (Figure 6a), a half line (Figure 6b), or a few columns within a line (Figure 6c), as in word insertion.

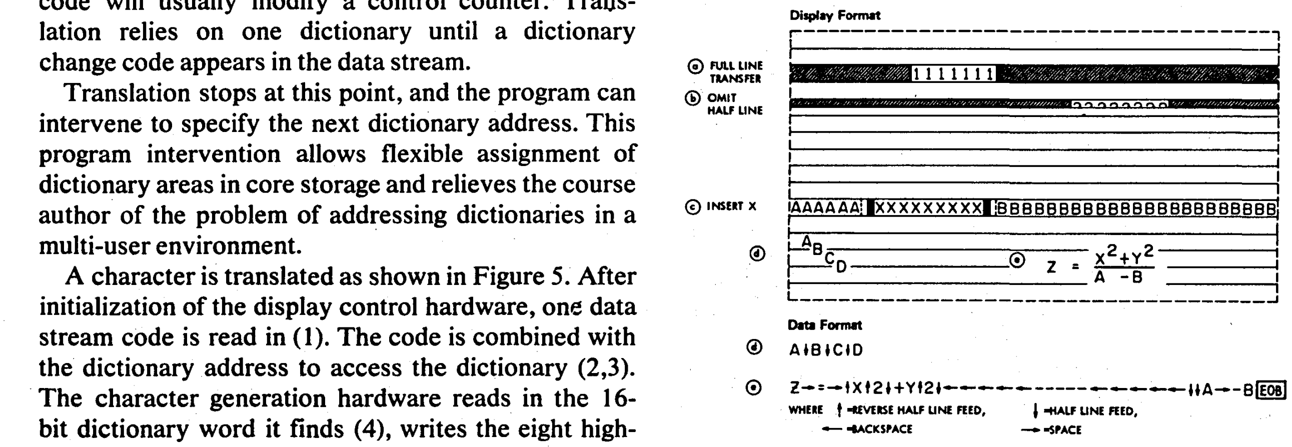


Figure 6 – Variable spacing with function codes

The video output to the selected display is blanked only for the line that is being recorded. The blanking extends over a line and a half to blank out the transients induced by the write circuits in the read amplifier. The blanking lasts 2.5 milliseconds and is not distracting to the person viewing the display.

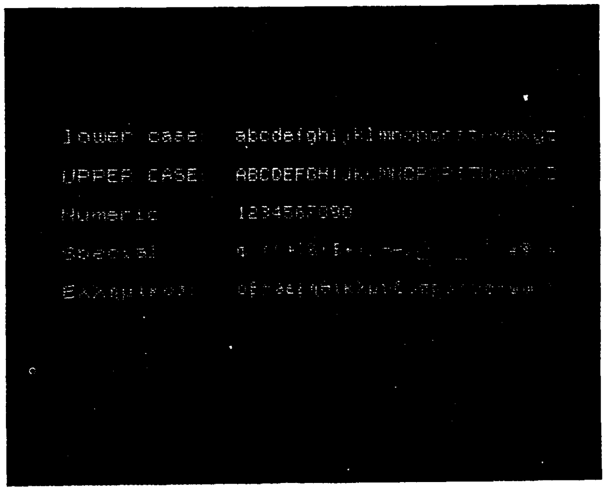
Since storage in the line buffer is basically an OR operation, characters can be superimposed on others (as in underlining or placing accent marks over letters) by combining the new character with a previously stored character.

The character generation logic also can handle subscripts and superscripts, as well as lines of text, formulas, and graphic images extending over more than one line of the display. Four function codes give the flexibility needed for this kind of variable spacing (see Figure 6d and 6e): half-line feed and reverse half-line feed to move a character down or up half a line (e.g., for subscripts and superscripts); space and backspace to move a character one column to the left or right. With these function codes, the data stream in the instructional system can have the same basic format for CRT displays and typewriters—the two basic output modes; and input formatting is simplified.

The vertical displacement codes alter the scan reference count in the line buffer address counter, while the horizontal displacement codes alter the column count. The original address stored in this counter is specified by each translate command.

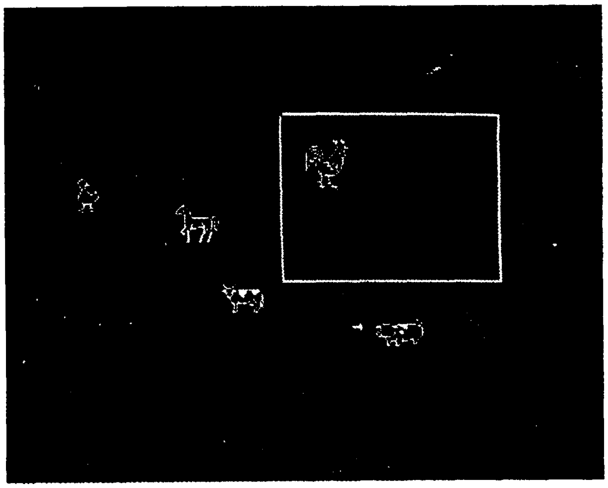
Since the line buffer stores just one line (12 scans) at a time, it is necessary, in writing an expression that extends over many scan lines, to give one translate and one transfer command for each line.

Figure 7 shows how characters and graphic images constructed in this way actually appear on the face of the CRT.



The light pen

An optional feature on the CRT display is the light pen, a pen-shaped device that lets the student respond directly to the displayed instructional material.



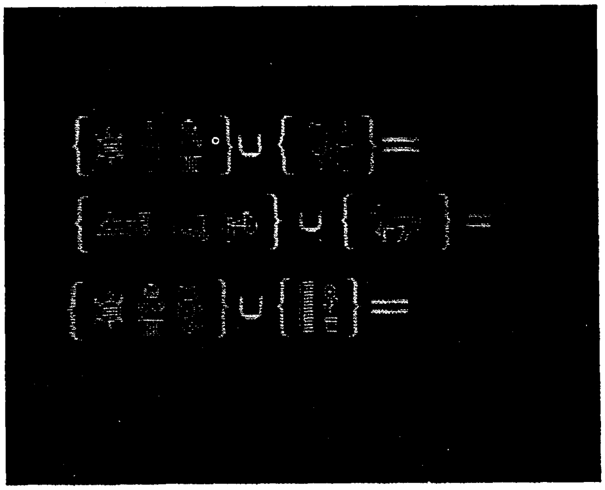


Figure 7—Sample CRT displays

The pen is a light-sensing device that transmits signals to the light pen adapter.

To select a response from a number of alternatives displayed on his screen, the student presses the tip of the pen against a designated target area. The required depression of the pen tells the system that the student is actively responding (and not just moving the pen over the face of the display) and gates the detection circuits that will transfer position coordinates of the selected target to the program.

Light pen operation may be understood from the diagram of Figure 8. Depressing the tip closes a switch within the light pen. The photodiode can now detect the phosphor light output as the electron beam sweeps across the selected target on the face of the CRT, and the station is in an enter mode. In this mode, horizontal and vertical sync signals from the CRT deflection circuits are gated back to the light pen control logic, and the amplified photodiode pulses are gated to the same coax line.

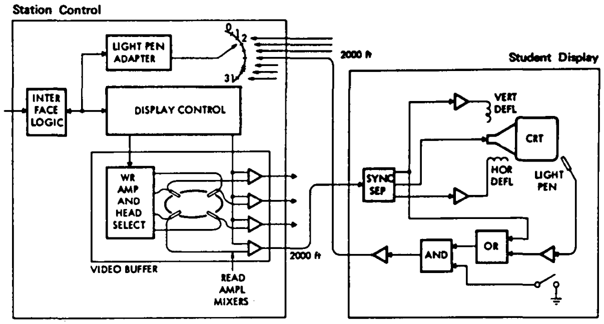


Figure 8 – CRT light pen operation

The horizontal and vertical sync signals from the display control unit are used by the control logic to determine that the light pen is in the enter mode. The horizontal sync pulses also reset and start a counter that counts timing pulses (from the video buffer clock track) until a photodiode detect pulse is received. The number of timing pulses between the horizontal sync pulse and the photodiode pulse (see Figure 9) gives the horizontal position of the pen on the face of the CRT. The vertical position is obtained by copying the display control half-line counter at the time the photodiode pulse is received.

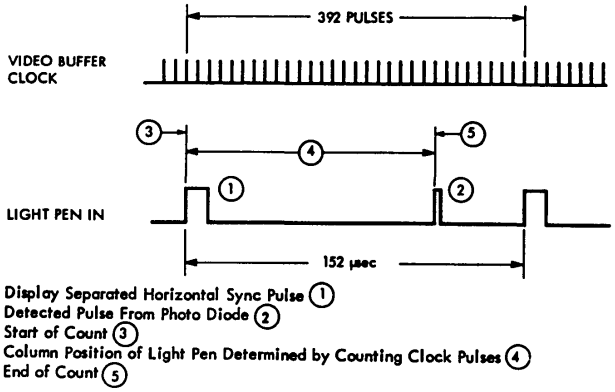


Figure 9 – CRT light pen timing

One control unit handles input from the light pens at all 32 student stations through time multiplexing. The control logic scans all the pen input lines in succession until it detects the sync signals that identify a station in the enter mode. The logic then waits for the detect pulse, determines the light pen position, and interrupts the processor to transfer the position coordinates to the program. Once the interrupt is serviced, the logic automatically resumes the scanning operation. Provision is made to avoid tying up the control circuits if a light pen should accidentally be depressed against a nontarget area of the CRT or some other surface. If no detect pulse is received during a full revolution of the disk buffer, scanning automatically resumes. The resolution of the light pen is an area one column wide and half a line high.

The display subsystem is capable of servicing one light pen every 33 milliseconds, on the average. Since one light pen is serviced at a time, the servicing rate varies with the number of active light pens.

CONCLUSION

The objectives were to design a very versatile display subsystem at a minimum system cost with available technology. To achieve these objectives, hardware and software design were closely coordinated.

Video buffering on magnetic disks proved to be an economical method of storing and refreshing 32 independent displays.

The design capitalized on the availability of a processor needed for the general system operation. User-definable, program-changeable fonts located in core storage met the need for many large character sets, capable of writing simple graphic images as well as alphanumeric text. The hardware design was both general (independent of the font) and versatile, giving a high degree of software control over the construction of expressions (e.g., accents, superscripts) and their positioning on the display (e.g., word insertion).

The following features were implemented in hardware to minimize subroutines size and associated processing overhead.

1. The number of interrupts was reduced by making the character generation a special instruction instead of a pure channel operation.
2. The limited availability of core space was eased by a hardware chaining of small core areas.
3. Finally, the hardware was designed to handle character streams in a manner compatible with conventional typewriter character streams. This had the double advantage of CRT-typewriter character stream compatibility and simplified formatting of the student inputs for the CRT display.

ACKNOWLEDGMENT

The CRT display system described, as an integral part of the IBM 1500 Instructional System, is a result of a cooperative effort of the engineers and programmers in the IBM Instructional Systems Development Department, to which many others elsewhere in IBM have also contributed. Special thanks must be given to Kenneth Senne and P. A. Smith, who worked with the author on the logical design and implementation of the CRT display subsystem, and to M.G. Hurley, W. D Musson, and Dr. G. H. Royer, who were very helpful in the preparation of this paper.

REFERENCE

1 R AZIZ
An instructional display terminal
Proceedings of the Eighth National Symposium of the Society for Information Display May 24 1967 San Francisco California p. 83