A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME SHARING CONSOLE

Robert H Stotz, et al

Massachusetts Institute of Technology
Cambridge, Massachusetts

July 1967

ESL-TM-316

A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE

by

Robert H. Stotz and Thomas B. Cheek

July 1967

Contract No. Nonr-4102(01)
DSR 79474

Electronic Systems [illegible]
MASSACHUSETTS [illegible]
Department of E[illegible]

TECHNICAL MEMORANDUM Copy No. 39

ESL-TM-316

A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE

by

Robert H. Stotz and Thomas B. Cheek

July 1967

Contract No. Nonr-4102(01)

DSR 79474

Work reported herein was supported by Project MAC, an M.I.T. research program sponsored by the Advanced Research Projects Agency, Department of Defense, under Office of Naval Research Contract Number Nonr-4102(01). The M.I.T. Project Number for the part of the work in the Electronic Systems Laboratory is DSR 79474.

Approved by John E. Ward
John E. Ward
Deputy Director

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

ABSTRACT

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays--a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of $5,000 or less. The unit uses a direct-view storage tube (DVST) for a display screen and contains a vector generator and a symbol generator for the full ASCII symbol set. It can connect to a central computer via a 1200-2400 baud dataphone line. A manually-controlled electronic cursor for graphical input to the computer can also be added.

ABSTRACT

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays--a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of $5,000 or less. The unit uses a direct-view storage tube (DVST) for a display screen and contains a vector generator and a symbol generator for the full ASCII symbol set. It can connect to a central computer via a 1200-2400 baud dataphone line. A manually-controlled electronic cursor for graphical input to the computer can also be added.

CONTENTS

FOREWORD

This basic paper was presented at the 8th National Symposium on Information Display in San Francisco, California in May, 1967. Since that time several changes have been made to the format of ARDS-II. These changes were:

1. Changed Set Point and Line codes from DC2 and DC3 to GS and RS, respectively.
2. Added Short Vector command using code US.
3. Added Mode Control.
4. Extended Lines from 9 bits magnitude to 10 bits magnitude.

The new format is discussed in complete detail in Appendix B.

Appendix A contains additional photographs to better illustrate the techniques used to build ARDS-II.

LIST OF FIGURES

A. INTRODUCTION

Experimental computer time-sharing systems have been in operation for several years, and from all indications, will soon become widely available. With a few notable exceptions, users of these systems communicate through mechanical teletypewriters of some form (e.g., Teletype, IBM 1050) which operate over standard switched telephone lines and can thus be located at virtually any remote site. Although apparently satisfactory as keyboard input devices where they match man's manual dexterity, teletypewriters are woefully inadequate for computer output due to their slow speed and rigid format. Alphanumeric output at 10 to 15 characters per second is well below human scanning speed, and input and output of graphic data is cumbersome, if not impossible.

At Project MAC, M.I.T., there has been an effort to define the elements of an improved time-sharing terminal and to build working hardware for experimentation and demonstration. This paper discusses the results to date of this work. The opinions expressed are based primarily on our own experiences and are colored by Project MAC's avowed goal of developing a prototype of a computer public utility.[1*]

B. DESIRED CHARACTERISTICS

It is evident that there is an urgent need for a better console for time-sharing systems. We feel it should be a display device capable of handling characters, points, and lines in a free format. It should

---

* Superscripts refer to numbered items in the Bibliography

operate as a stand-alone unit from a standard telephone line as teletypewriters do now, but it should make use of the full available data rate phone lines offer (today's teletypewriters operate at 130 baud over lines capable of operating as fast as 2000 baud). The restriction to telephone-line operation has been imposed because the telephone system represents the only communication network widely available to the public, and it appears that its services will be used in any computer public utility. Clustering of consoles about a central control unit is a time-honored way to bring the cost per terminal down. However, this can be done only in certain environments and is not suitable for a general-purpose time-sharing terminal.

Graphic output is essential to the full development of the potentials of the time shared computer. When the computer can communicate in pictures, a whole new dimension is added to man-machine "conversations." Lists of numbers become graphs, bridge structures can be illustrated, electrical circuits drawn, flow pictorialized, etc. In the vast areas of computer application where the real world is modeled (e.g., computer-aided design, simulation, process control), use of graphics is of particular importance. A highly desirable adjunct to graphic output is graphic input. This capability is required to truly "converse" in graphical language.

It is important that hard copy be available to the user, but with suitable software aids we feel for most applications it is not required at the local console. At Project MAC, over 90% of the teletype output paper goes directly into the trash basket. The hard copy that is taken away is generally for record purposes--a need which can be fulfilled by centralized hard-copy generators. The output paper is occasionally used while at the console to refer to previous data or conversation. If appropriate system programs can retrieve this information quickly and easily, this "need" for hard copy vanishes. Those users who absolutely require immediate local hard copy must either use teletypewriters, take polaroid photographs or pay for one of the many emerging hard copy devices.

The size and resolution of the display screen, the quantity and type of data that may be displayed, and the cost of the terminal are obviously subject to engineering compromise. We shall here state some "specifications" which we feel are reasonable as goals, although we realize that they are not necessarily within the state of the art today.

The display area should be roughly 100 square inches, which is about the area of standard 8 1/2 x 11 paper. It would be desirable to have resolution comparable to the printed page but we can certainly live with a great deal less. The resolution of high-quality CRT displays today (50 black-white line pairs per inch) is adequate. With this resolution and size, the device should be able to display 4000 characters. Many users would be content with less, but anything less than 1000 characters is inadequate for a general-purpose time-sharing console.

The picture should erase rapidly (less than 1 second) and display new data as fast as it comes over the line from the computer. In order to enhance the speed of receiving new information, the data should be as highly coded as is economically feasible.

Since a user may work for hours at a time and he will often want to refer to other papers, the console must be easily legible in a moderately lighted room without flicker, blink or eyestrain.

To allow proper operation and control in a computer time-sharing environment, the console should also have such features as computer controlled keyboard lock, an interrupt capability, and a unique identification code which the computer can read. These are requisite properties for any time-sharing terminal and are discussed in an earlier paper on the MULTICS system.[2]

In order for such a console to fulfill its role as a computer time-sharing terminal, it must be inexpensive. The production quantity cost we would like to see is from $3,000 to $5,000. Although somewhat arbitrary, this figure looks reasonable from several viewpoints. It provides an order-of-magnitude increase in capability over present teletypewriters for a modest increase in cost. From a systems point of view, terminal costs (assuming that there are several hundred of these units per computer) will be roughly that of other major system components (CPU, memory etc.). Thirdly, with the cost of components coming down, with proper design, and with the projection of large-volume production, a $3,000 to $5,000 sales price seems attainable.

It is understood that for some time to come, time-sharing display terminals will fall short of the above goals, particularly in meeting the cost objective.[8] Our purpose has been to use these goals as standards to evaluate various design techniques in order to choose one that can eventually evolve into hardware that will fulfill the requirements.

In considering possible approaches, TV displays were ruled out because of the difficult transformation of generalized graphics to video format. If this is done at the remote unit it is expensive. If it is done at the central computer, the mass of data (10^6 bits per picture) requires much too long a time to transmit over a telephone line. Repetitively regenerated random-access digital displays, while technically feasible for telephone-line operation, require high-cost deflection amplifiers and high-speed memory to keep large amounts of data refreshed without noticeable flicker.

The approach we have taken is to abandon the refreshed-type display, and store the image directly on the viewing surface. At the moment, we are using a direct-view meshless storage CRT (DVST) made by Tektronix. Although it has only 3" x 4" active area, larger size tubes are technically quite feasible. Beside the DVST, there are a number of other promising image-storage techniques under development (e.g., photochromics, EL-PC panels, photoplastics) which may someday permit large, high-resolution image-storage displays at very low cost. It is an important restriction of a computer time-sharing terminal that data be immediately visible as it is written. Image storage techniques that require developing and fixing of a latent image (e.g., photography) will not perform well in a man-computer conversation.

C. INTRODUCTION TO ARDS-II

To confirm our conviction that a low-cost display using image storage techniques was feasible, we have designed and constructed a prototype unit, which has been designated ARDS-II (Advanced Remote Display Station - II), a block diagram of which is shown in Fig. 1. A previous breadboard output-only unit (ARDS-I) was constructed in 1965 (partly as a thesis project) to demonstrate the feasibility of the basic display generation techniques.

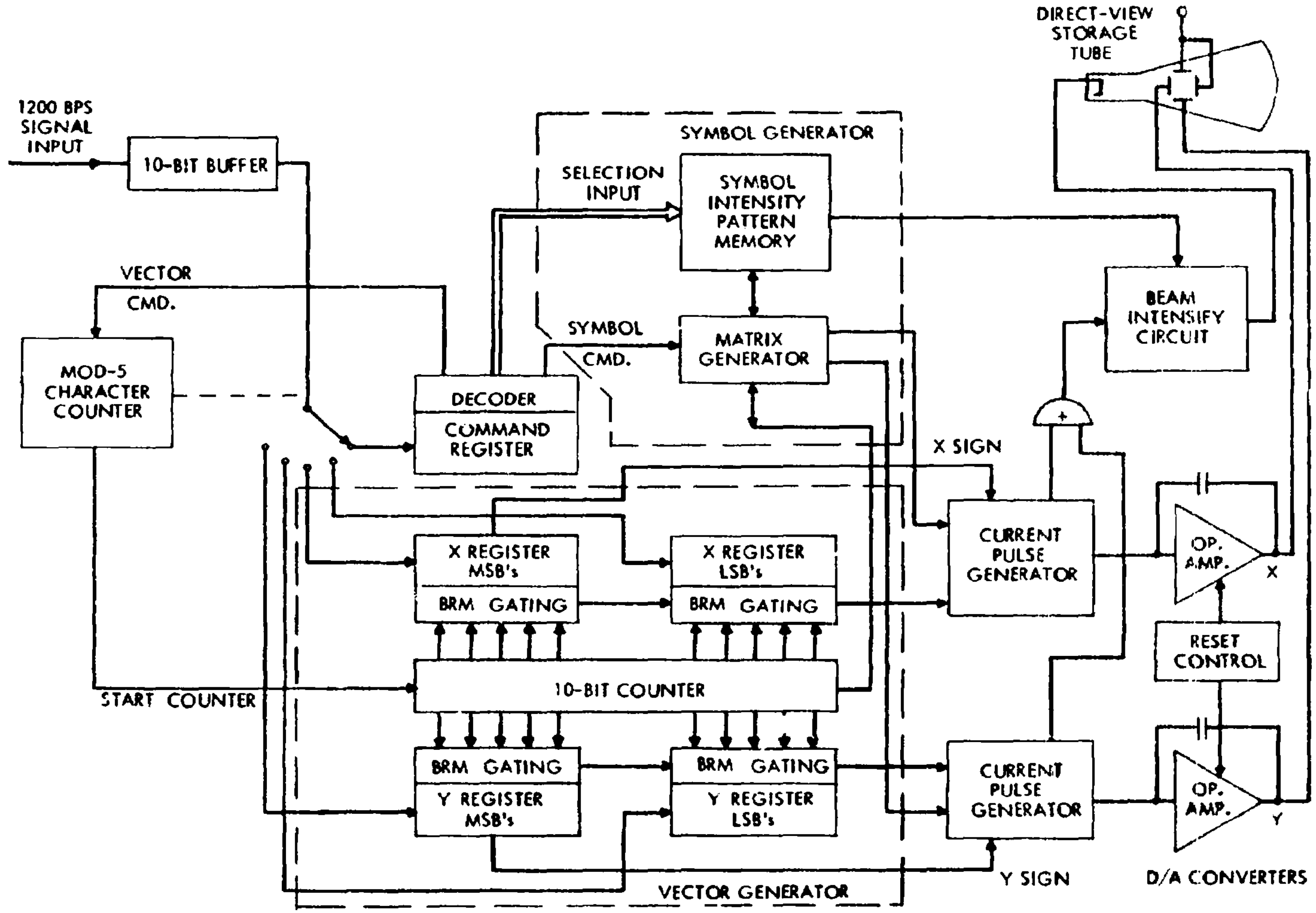


Fig. 1 Block Diagram of ARDS-II

The ARDS-II consists of (1) a direct-view storage-tube (DVST) oscilloscope as a display device, (2) a 94-symbol (ASCII) keyboard, (3) a simple graphical input device, and (4) an electronic control unit that interfaces with a telephone line, provides analog voltages to drive the display screen, and properly formats signals from the keyboard and graphical input to send to the computer. It is connected as a remote console to the Project MAC Time-Sharing System, in which as many as 30 consoles may be in operation at once. The station will be connected to the computer over a dialed-up telephone line. The output rate from the present computer is 1200 baud, but ARDS-II is capable of operation at 2000 baud. Typical displays produced by ARDS-II are shown in Fig. 2.

Our efforts in building the ARDS-II have been mainly concentrated in designing a suitable electronics package to control the display, and in developing a low-cost graphical input system. The keyboard is an off-the-shelf commercial product, and we are confident that suitable high-resolution DVST display monitors will be available commercially in the near future. Presently, we are using a special high-resolution DVST in a standard Tektronix 564 5-inch storage-tube oscilloscope.

D. DESIGN CONSIDERATIONS

The electronics design problem was, of course, highly influenced by the restraints imposed by the overall system. To review briefly, the chief restraints were:

- Bandwidth-limited data input -- telephone line speed.
- Need for fast text display and random vector drawing ability.
- Need for a low-cost unit in a stand-alone configuration.

The use of a direct-view storage CRT was a key design decision. With such a device, there is no need to provide either an electronic memory or high-speed electronics, such as are required for rapid picture regeneration in displays using conventional CRT's. New data is written only once and may be entered randomly on the display screen at rates compatible with the relatively low-input bandwidth; thus speed requirements on the electronics are quite modest.

The low-input bandwidth implies that efficient data coding be used to generate complex displays in a reasonable time--say 10 to 20 seconds for a full screen. Unfortunately, such coding increases the amount of data processing equipment at each display station. In this case, it was concluded that a vector and a symbol generator must be included in each display terminal.

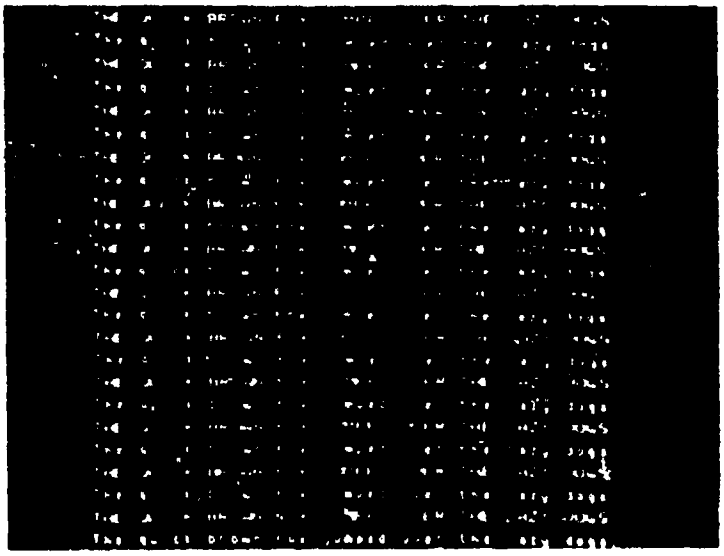
At this point, we can see the general requirements of the electronics package. It must be able to accept serial binary signal trains, consisting of commands and data, and use these signals to program internal vector and symbol generators which provide vertical and horizontal deflection voltages for the CRT. Since the "stand-alone" requirement means that identical sets of electronics will be needed at each station, the design problem was to perform these functions with low-cost equipment, exploiting if possible the low requirements on speed of display generation, as compared to regenerative displays.

E. BASIC BEAM POSITIONING

Our first task was to choose an appropriate beam-positioning technique that would be capable of random point plotting, vector generation, and symbol generation. Our group had previously constructed a complex display station known as the ESL Display Console,[3] which has been operating at Project MAC since 1963. This machine is an incremental digital display generator which uses two binary-rate-multipliers (BRM's) to draw lines. The BRM's, in conjunction with up-down counters and digital-to-analog converters, move the CRT beam in small, discrete voltage steps. By intensifying the beam after each step, a constant intensity line, made up of a series of closely-spaced dots, is drawn on the screen.

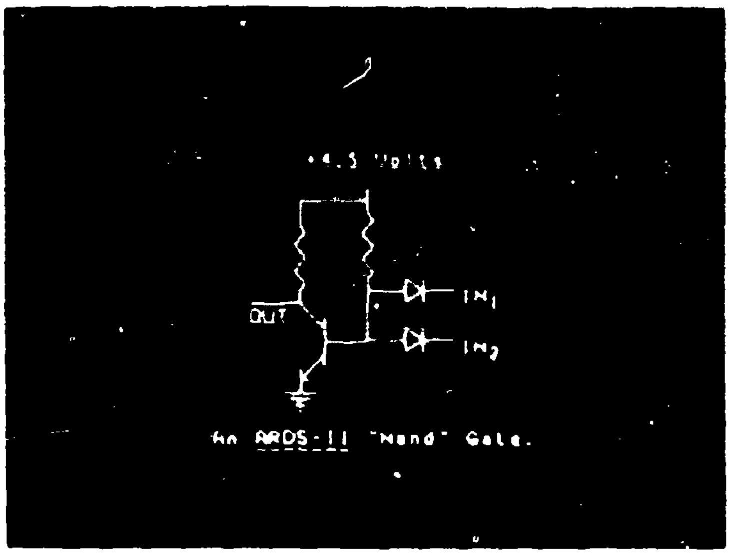
Because of the success of this incremental scheme in producing very clear and stable displays, and its basic simplicity, we decided to use a similar approach in the ARDS-II. BRM's can easily be made from readily-available digital building blocks. For the pulse integrator and digital-to-analog conversion tasks, however, we decided to abandon conventional techniques. Pulses produced by each BRM are shaped to have constant amplitude and duration and then are fed to an operational amplifier connected as an integrator. Polarity of

a) 94 Symbol ASCII Character Set

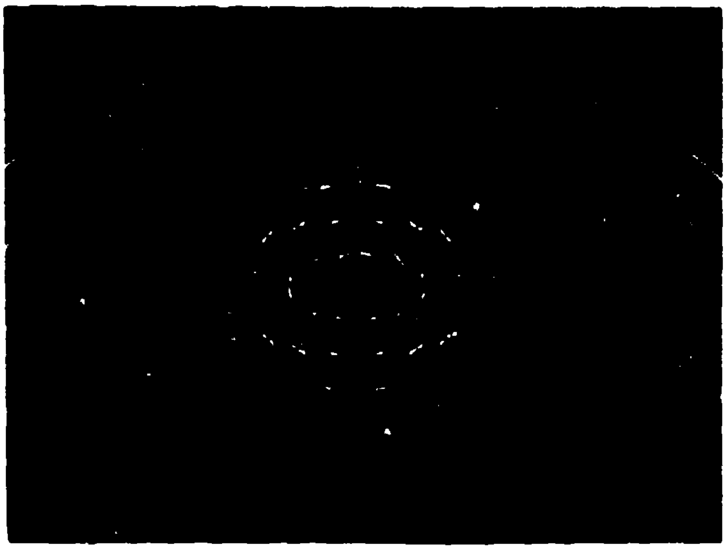


b) 1104 Symbols (including spaces)
Writing time: 10.2 seconds

Fig. 2 Four Examples of Displays Produced on ARDS-II



c) Mixed Graphics and Symbols
Writing time: 4.7 seconds



d) Curves Produced by Straight Line Segments

Fig. 2 (Continued)

the pulses is controlled in accordance with the sign of the vector component. The resulting amplifier output produces the same discrete-step characteristic as the more expensive up-down counter and D/A converter combination. Thus in the ARDS-II, all beam motions are controlled by pulse inputs to the integrators.

The major circuit problem with this approach is that the two integrating capacitors must "hold" the voltages impressed upon them, without significant drifting during the 10 to 20 seconds required to produce a typical "picture." Because of various leakage paths, the capacitors will tend to discharge and cause the beam position to drift. However, with careful design--use of very stable amplifiers, very low-leakage FET input gates and high-quality capacitors--drift has been held to one screen position per two minutes. This is quite adequate. Despite the requirement for high-quality components in the integrators, costs are substantially less than the equivalent up-down counter and D/A converter, primarily because of the present availability of low-cost solid-state operational amplifiers.

F. VECTOR AND POINT PLOTTING

Pictures are drawn on the ARDS-II by using incremental vectors--that is, picture elements are constructed by connecting straight-line segments end-to-end. In a Vector command, the incremental vector lengths are defined as 11-bit sign-magnitude numbers, which are loaded into horizontal and vertical registers. The values stored in these registers program the binary-rate-multiplier to produce synchronized ΔX and ΔY pulse trains of 0 to 1023 pulses each. These pulse trains are fed as positive or negative pulses, depending on the ΔX and ΔY sign bits, into the two operational amplifiers where they are integrated and cause the CRT beam to trace out the desired line. Vectors can be visible or invisible, as specified by an intensity bit in the command.

In the Vector command described above, the beam motion is incremental, i.e., each vector starts at the beam location resulting from the previous command. In order to produce starting points for vector and symbol sequences, and to perform random point plotting we also needed to be able to position the beam at absolute screen locations. Since the absolute screen location X, Y is the same as the end position of a vector with incremental values ΔX, ΔY plotted from starting location 0, 0, it was possible to use the vector generator to also act as a "set-point" generator.

A Set Point command operates in the same manner as a Vector command except that reed relays short the integrating capacitors and return the beam to the screen center (0, 0) before the vector generator is started. No penalty in display speed is incurred because the slow data input rate provides more than enough time to accomplish both the zeroing and vector drawing functions in one instruction cycle.

G. SYMBOL GENERATION

Quite often, graphical displays are designed with symbol generation and vector generation being treated as entirely separate problems. By sharing certain circuits between vector and symbol generating tasks we have achieved important cost reductions.

Given the basic incremental beam positioning system which has been described, all that is needed to create a dot-matrix symbol generator is a pair of counters to step the beam through a matrix pattern, and a memory to hold the specific intensity pattern for each symbol, which causes the CRT beam to blank and unblank as it moves through the matrix. The matrix pattern includes intersymbol spacing, i.e., it leaves the CRT beam in the proper screen position to plot the next symbol, so that no beam repositioning is needed between symbols in a text line. Initial positioning for the first symbol in a line is established by a previous vector or set-point, thus there is full flexibility in choosing symbol locations.

Because we are limited to telephone line transmission rates, plotting speed is not a problem, and a 7 x 9 dot matrix was chosen to display the 94 printable symbols of the ASCII code with reasonable fidelity. This requires a 96-word, 63-bit read-only memory for the symbol patterns (a blank symbol is used for "space," and a filled-in pattern for "delete"). The symbol memory is a 6720-bit diode matrix array contained on a single one-half-inch square integrated-circuit chip.[4,7]

H. COMMUNICATIONS AND CONTROL

The function of the communications and control portion of ARDS-II is to accept a continuous serial bit stream from the telephone line, and decode this into commands and data to enable the machine to perform useful functions--such as drawing a line or printing an alphanumeric symbol. For practical reasons, transmission is in the form of fixed-length "characters" with "stop" and "start" bits to insure continuous synchronization between source and receiver. The character set chosen is the American Standard Code for Information Interchange (ASCII) which will be used in the new Project MAC MULTICS System. This choice was also prompted by the desire to build a console that would be compatible with other computer systems. In the ARDS-II, therefore, characters are 10 bits long (with a start bit, 7 data bits, a parity bit, and a stop bit).

The ARDS-II is designed to work with a five-character sequence to define vector commands, and a one-character sequence to define text commands. A 10-bit input shift register, driven by a clock that is re-synchronized by each new character, serially stores an entire character. When the register is loaded, the data bits are immediately shifted out to one of five 7-bit registers. A mod-5 character counter gates the first character into the command register. If this character is one of the 96 text symbols of the ASCII code, the symbol generator is activated, and the symbol is plotted on the display screen at the current beam location. At the same time the character counter is reset and a new character is assembled in the buffer.

If the command character is one of the two ASCII control characters, GS or RS (the Set Point and Vector commands, respectively), vector plotting is required and the next two characters will be loaded into the X (horizontal) registers and the following two characters into the Y (vertical) registers. When the character counter indicates all five data registers are filled, the vector generator is started, and the resulting vector is plotted during the time required for the input buffer register to be filled again (8.8 milliseconds with the present 1200 baud transmission). Since only 22 bits are required to specify a vector, there are available bits for specifying whether to intensify or not, and for specifying magnification (dot spacing).

I. GRAPHICAL INPUT

For graphical input, one would like to be able to move a pointer or "cursor" over a stored picture and yet not store the image of the cursor. Fortunately, the characteristics of the screen we are using are such that there is a stable gap between the "image-visible" intensity level and the "image-storage" level. Also, there is a maximum writing rate beyond which storage cannot occur. Therefore, by lowering normal intensity and moving the beam through the cursor pattern rapidly, a very visible but nonstoring cursor is available.

The cursor pattern is generated locally, and its position on the display screen is controlled by means of a hand-held box that is moved about on a surface. This box, similar to a device called the "mouse" by its developers at Stanford Research Institute,[5] has two potentiometers mounted at right angles to each other. Wheels attached to the potentiometers contact the surface, and resolve the motion of the box into two orthogonal components which are fed as voltages to the CRT deflection inputs. Thus, the cursor on the screen "follows" the motion of the "mouse."

At the request of the operator (pushbutton control), an analog-to-digital converter digitizes the vertical and horizontal components of the cursor's position and transmits them to the computer. The operator's program can then interpret these position values as it sees fit--as end points of lines when in drawing mode, as a pointer in locating one of a number of displayed objects, and so on.

J. CONCLUSIONS

We have described the need for an inexpensive remote display console for computer time-sharing systems and we have described our approach to a solution. There are numerous other applications for low-cost displays and there are also many other approaches to solutions.[6,8] One point is evident. The impact of graphic I/O on the computer community is just beginning to be felt, but until low-cost graphic displays become readily available, this impact will be greatly impeded.

Even when such equipment becomes available, extensive system software is required to make it useful. For time-sharing systems the problem of easy, rapid recall of previous display is particularly important on a "soft copy" console. Much work is also needed in development of graphical languages and application programs. How well software problems such as these are solved will largely determine the effectiveness of low-cost display devices.

We are confident that both hardware and software problems will be solved and that the low-cost remote display console will soon become a powerful new tool for man/computer communication.

APPENDIX A

ARDS-II PHOTOGRAPHS

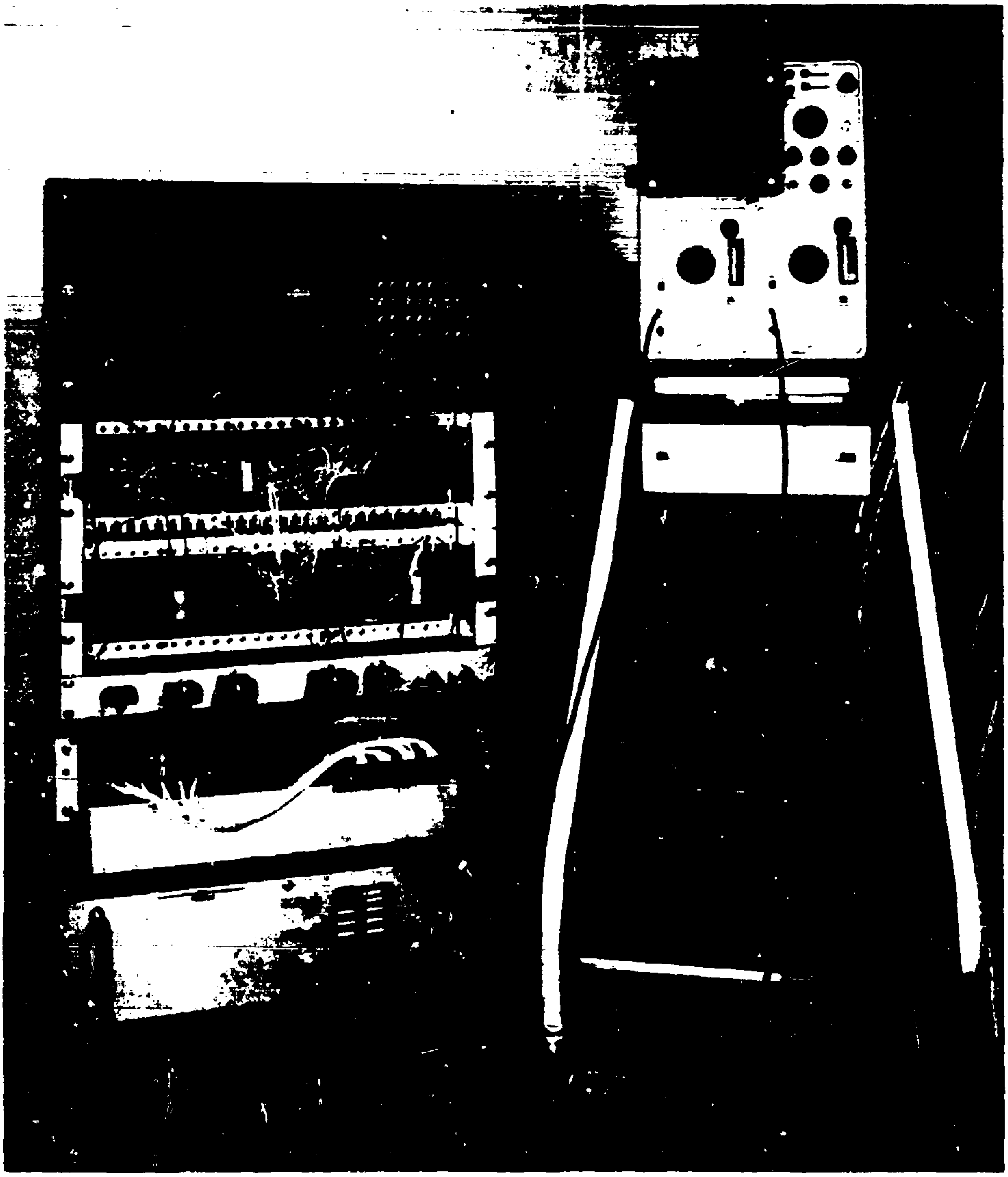


Fig. A.1 ARDS-II with Tektronix 564 Storage Oscilloscope

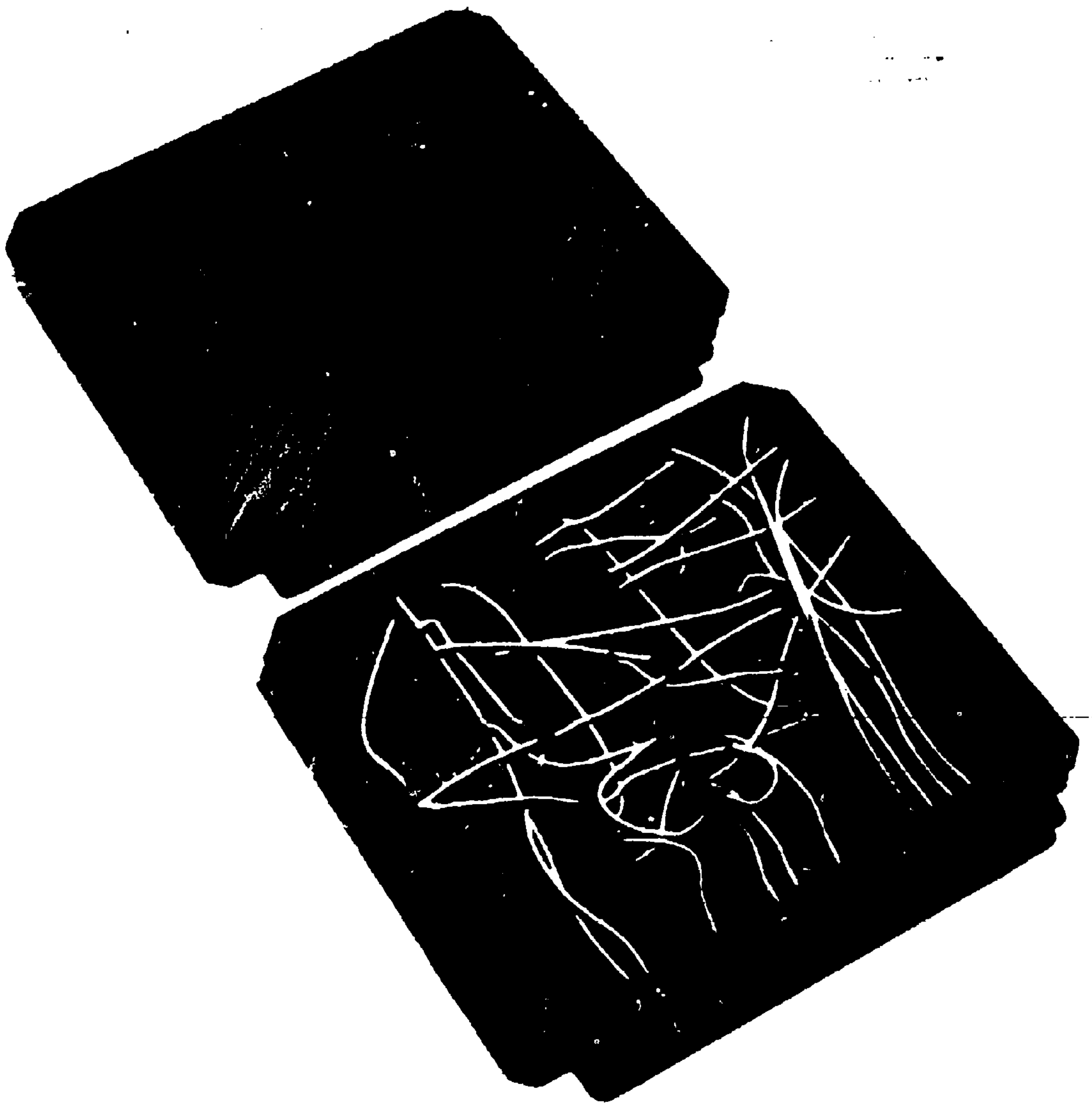


Fig. A.2 A Typical Logic Card from ARDS-II

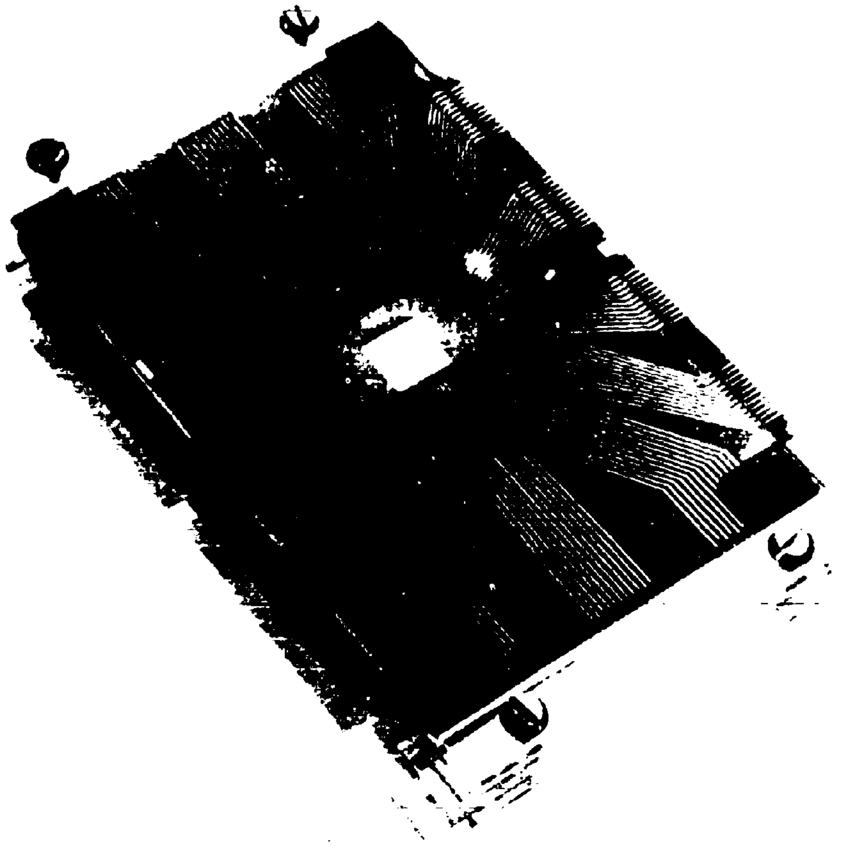


Fig. A.3 Autonetics Diode Matrix 96 x 70 Array

Fig. A.4 Read-only Memory for Character Generator with Diode Matrix, Sense Amplifiers and Selection Logic

APPENDIX B

CODE FORMAT FOR ARDS-II

General Format

ARDS-II has been designed to be compatible with the American Standard Code for Information Interchange (ASCII).* The ASCII format is basically designed for transmission to and from symbol-oriented devices such as teleprinters and magnetic tape drives, and provides a standard symbol set, printer-controls (backspace, carriage return, etc.), formatting controls (start text, end of record, etc.), and communications controls (enquiry, acknowledge, end of transmission, etc.). ARDS-II, which has a graphic as well as a symbol capability, requires pure binary data when in graphic mode, and special control codes beyond those for which there is room in the standard set. The ASCII standardizing committee has not included any specific provisions for transmission of graphic controls or binary data, however there are numerous possibilities for incorporating these features within the standard code set. For ARDS-II, certain of the standard ASCII format control characters have been given special meanings, and a class of "key" characters has been defined for additional control functions. Also, a "6-bit mode" of binary data transmission has been chosen which avoids conflict with standard ASCII control codes, and permits transmission through code-sensitive equipment.

Figure B-1 shows the ASCII code. The standard control codes are those in the first two columns (b7 = b6 = 0), and these divide into two groups: communication controls, and device controls (note that all have an assigned ASCII meaning). All others are symbol codes, except for DEL (delete) which can be interpreted as either a symbol or a control. The graphic symbols and numerals in the third and fourth columns (b7 = 0, b6 = 1) we have defined as key characters. These can be specially interpreted by ARDS-II to provide the necessary additional controls, as will be discussed.

---

* Communications of the ACM, Vol. 8, No. 4, April 1965, pp. 207-214.

Bit Positions b7 b6 b5 → / b4	b3	b2	b1	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
0	0	0	0	NUL	DLE	SP	0	@	P	`	p
0	0	0	1	SOH	DC1	!	1	A	Q	a	q
0	0	1	0	STX	DC2	"	2	B	R	b	r
0	0	1	1	ETX	DC3	#	3	C	S	c	s
0	1	0	0	EOT	DC4	$	4	D	T	d	t
0	1	0	1	ENQ	NAK	%	5	E	U	e	u
0	1	1	0	ACK	SYN	&	6	F	V	f	v
0	1	1	1	BEL	ETB	'	7	G	W	g	w
1	0	0	0	BS	CAN	(	8	H	X	h	x
1	0	0	1	HT	EM	)	9	I	Y	i	y
1	0	1	0	LF	SUB	*	:	J	Z	j	z
1	0	1	1	VT	ESC	+	;	K	[	k	{
1	1	0	0	FF	FS	,	<	L	\	l	|
1	1	0	1	CR	GS	—	=	M	]	m	}
1	1	1	0	SO	RS	.	>	N	^	n	¬
1	1	1	1	SI	US	/	?	O	_	o	DEL

Control Characters

Communication Control Characters

Key Characters

Second Category Representing Argument (Includes the Control Character DEL)

Control Character DEL Included in the Argument Set

Fig. B.1 USASCII Character Assignments

ARDS-II is built to operate from a character-asynchronous data modem and it is therefore necessary to identify the beginning and end of a character and to establish proper synchronization. This is done by adding a Start bit (0) and Stop bit (1) to each 8-bit character, as illustrated below:

0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Parity	1
Start Bit			Time ⟶						Stop Bit

ARDS-II operates on a mode-control basis, i.e., it is placed into a particular mode by receipt of a unique code, and will stay in this mode until it receives a code placing it into a different mode. The presently available modes are Symbol, Set Point, Extended Vector and Short Vector, but the code set has the capacity for adding many more modes. Table 1 shows the present mode control codes, which make use of the "separator" codes of ASCII.

Table 1

Mode	Control Character	Octal
Symbol	FS (Or any control code except GS, RS, US)	034
Set Point	GS	035
Extended Vector	RS	036
Short Vector	US	037

Symbol Mode

Symbol mode is the "normal" mode, and the Reset button forces the equipment to this state. In it, ARDS-II acts like a teletypewriter. It will display whatever symbol is called for and space one symbol position right, ready for the next symbol. For the DEL (delete) code, it will print a "blob", approximately filling a character position, and space right. ARDS-II will also respond to the control characters CR (carriage return), BS (back space), FF(erase), and ENQ (enquiry). FF will cause erasure of the entire screen but will not move the beam position. ENQ causes ARDS-II to return a fixed 6-character sequence which is its unique identification code.

Set Point Mode

This mode allows positioning of the beam to an absolute coordinate location. It is normally used to locate the start of a picture part, or simply to plot points on the screen. The point may be intensified or not, under control of the program. When the control character GS puts ARDS-II into Set Point Mode, it expects to receive a string of 4 characters which it interprets as binary data. In order to be transparent to any communication channel (which may be sensitive to communications control characters), all binary data characters have bit 7 a ONE. This leaves six binary bits of information per character, or a total of 24 binary bits for the Set Point Command. These are interpreted by ARDS-II as shown below:

	Start	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Parity	Stop	
Character 1	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	X data
2	0	2^5	2^6	2^7	2^8	2^9	INTENS	1	P	1	
3	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	Y data
4	0	2^5	2^6	2^7	2^8	2^9	--	1	P	1	

Addressable locations are from +1023 to -1023 in X and Y. Because the screen is rectangular, with the long dimension in the vertical, actual "on-screen" locations extend from -511 to + 511 in X, and from -899 to +511 in Y. Location 0, 0 is screen center in X, and slightly above center in Y. The point specified may be left blank or intensified under control of bit 6 of character 2.

ARDS-II will stay in Set Point Mode as long as it receives binary characters, and it will interpret and plot each set of four characters as a new point. If it receives a nonbinary character (b_7 = 0), it will leave Set Point Mode and enter the mode specified by the nonbinary character (note that there are 64 such characters). If this character is GS, it will return to Set Point Mode. If it is RS or US it will enter Extended Vector or Short Vector mode, respectively. If it is any other of the control codes (note that there are 29 of these) it will enter Symbol mode.

The remaining 32 nonbinary characters (ones with $b_7 = 0$, $b_6 = 1$) are called Key Characters, and are free to designate new modes. As of this writing, no specific assignments have been made, but they will be used to control other devices (e.g., cursor control or a hard-copy printer) or to modify existing modes (e.g., perhaps dotted lines or double-size characters). Binary characters may follow a key character and be used as arguments in the same manner they are in Set Point and Vector modes. An ARDS-II terminal will enter the mode specified by the key character and remain in that mode until a new control or key character is received, at which time it will go to the new mode specified. If ARDS-II receives a mode-control character which it cannot interpret, it will simply ignore the binary data stream until the next mode-control character. By this mechanism, a very large order code is provided so that a wide variety of special equipment can be controlled using the same data format.

Extended Vector Mode

The control character RS causes ARDS-II to enter the Extended Vector Mode, in which it interprets a string of four binary characters ($b_7 = 1$) as data for drawing a vector relative to the present beam position. The data is interpreted as shown below:

	Start	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Parity	Stop	
Character 1	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	ΔX data
2	0	2^5	2^6	2^7	2^8	2^9	INTENS	1	P	1	
3	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	ΔY data
4	0	2^5	2^6	2^7	2^8	2^9	--	1	P	1	

Since the ΔX and ΔY components are each specified by 10 bits (plus sign), vectors in this mode may be drawn with up to ±1023 increments in X and/or Y. The line may be blank or intensified, as specified by bit 6 of character 2. Each succeeding group of four binary characters is interpreted as a new vector until a mode-change character is received.

<u>Short Vector Mode</u>

For many picture purposes, a number of short lines may be needed, e.g., for drawing circles or connecting points on a graph. In such cases, the Extended Vector format with its 10-bit capability is wasteful of communication capacity, both in the computer and in the data link. Thus a Short Vector mode has been provided.

The control code US causes ARDS-II to enter the Short Vector mode, in which it interprets a string of two binary characters (b_7 = 1) as data for drawing a vector relative to the present beam position. The data is interpreted as shown below:

	Start	b_1	b_2	b_3	b_4	b_5	b_6	b_7	Parity	Stop	
Character 1	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	ΔX data
Character 2	0	SIGN	2^0	2^1	2^2	2^3	2^4	1	P	1	ΔY data

Vectors in this mode may be up to ±31 increments in X and/or Y. Only visible lines may be drawn, since there is no spare bit to be used for intensity control. Each succeeding group of two characters is interpreted as a new vector.

BIBLIOGRAPHY

1. Corbató, F. J., Vyssotsky, V. A., "Introduction and Overview of the Multics System," AFIPS Conference Proceedings, Vol. 27, Spartan Books, Inc., 1965.

2. Ossanna, J. F., Mikus, L. E., Dunten, S. D., "Communications and Input/Output Switching in a Multiplex Computing System," AFIPS Conference Proceedings, Vol. 27, Spartan Books, Inc., 1965.

3. Stotz, R. H., "Man Machine Console Facilities for Computer-Aided Design," AFIPS Conference Proceedings, Vol. 23, Spartan Books, Inc., 1963.

4. Electronics Magazine, May 30, 1966, pp. 152A - 152D.

5. English, W. K., Englehart, D. C., Huddart, B., "Computer-Aided Display Control," Final Report, Contract NAS 1-3988, Stanford Research Institute, Menlo Park, California.

6. Culler, G. J., Fried, B. D., "The TRW Two-Station On-Line Scientific Computer," Symposium on Computer Augmentation of Human Reasoning, Edited by Sass, M. A., Wilkinson, W. D., Spartan Books, Inc., 1965.

7. Cheek, T. B., "Design of a Low-Cost Character Generator for Remote Computer Displays," Technical Report MAC-TR-26' (Thesis), Project MAC, M.I.T., March, 1966.

8. The Computer Display Review, Edited by Gilmore, John T. et al., Charles W. Adams Associates, Cambridge, November 30, 1966.

AD
Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE by R. H. Stotz and T. B. Cheek, July, 1967, 27 pages incl. illus. and table. (Contract Number Nonr-4102(01), M.I.T. Project Number DSR 79471). Technical Memorandum ESL-TM-316.
Unclassified Report

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays -- a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of
(over)

UNCLASSIFIED

UNCLASSIFIED

AD
Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE by R. H. Stotz and T. B. Cheek, July, 1967, 27 pages incl. illus. and table. (Contract Number Nonr-4102(01), M.I.T. Project Number DSR 79471). Technical Memorandum ESL-TM-316.
Unclassified Report

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays -- a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of
(over)

UNCLASSIFIED

UNCLASSIFIED

AD
Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE by R. H. Stotz and T. B. Cheek, July, 1967, 27 pages incl. illus. and table. (Contract Number Nonr-4102(01), M.I.T. Project Number DSR 79471). Technical Memorandum ESL-TM-316.
Unclassified Report

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays -- a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of
(over)

UNCLASSIFIED

UNCLASSIFIED

AD
Electronic Systems Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139
A LOW-COST GRAPHIC DISPLAY FOR A COMPUTER TIME-SHARING CONSOLE by R. H. Stotz and T. B. Cheek, July, 1967, 27 pages incl. illus. and table. (Contract Number Nonr-4102(01), M.I.T. Project Number DSR 79471). Technical Memorandum ESL-TM-316.
Unclassified Report

The advent of time-shared computer systems has created a need for a flexible and relatively low-cost communication terminal for remote computer access. Most time-shared systems now use mechanical teletypewriters which are slow and unable to present graphic displays -- a serious limitation in many sophisticated computer applications. The best candidate for a teletypewriter replacement appears to be a CRT console with an alphanumeric keyboard input which can connect as a "stand alone" unit to a standard telephone line. In a joint effort, the Electronic Systems Laboratory and Project MAC at M.I.T. have undertaken to design such a console with production cost objective of
(over)

UNCLASSIFIED

UNCLASSIFIED