DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE COMPUTER DISPLAYS

by

THOMAS BURRELL CHEEK

S.B.E.E., Massachusetts Institute of Technology
(1964)

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
February, 1966

Signature of Author Thomas B. Cheek
Department of Electrical Engineering, January 21, 1966

Certified by John E. Ward
Thesis Supervisor

Accepted by Truman S. Gray
Chairman, Departmental Committee on Graduate Students

DESIGN OF A LOW-COST CHARACTER GENERATOR FOR REMOTE COMPUTER DISPLAYS

by

THOMAS BURRELL CHEEK

Submitted to the Department of Electrical Engineering on January 21, 1966 in partial fulfillment of the requirements for the degree of Master of Science.

ABSTRACT

A requirement exists for a low-cost remote display terminal with alphanumeric and line-drawing capabilities for use with time-shared computer systems. This thesis, conducted as part of the overall remote display design project, was undertaken to investigate novel approaches to character generation, with the goal of drastically reducing present-day costs for such devices.

A survey of existing devices and character generation techniques was carried out, and a design approach was chosen which takes advantage of mass-fabrication techniques. This includes using a five-by-seven dot matrix raster and a resistor array "read-only" character memory for the 96 printable symbols of the Revised Proposed ASCII Code. Circuits designed included a dot matrix generator, and a resistor array memory with selection logic sense amplifiers, and a shift register output buffer.

An experimental character generator with an eight-word memory was built, largely using integrated circuits and was found to work as desired. It is concluded that the design approach will yield a character generator that is of low enough cost to find wide use in remote computer terminals.

ACKNOWLEDGEMENT

The author wishes to express his sincere appreciation to Mr. John E. Ward for help and suggestions both in the writing and in the supervising of this thesis. Mr. Robert Stotz also deserves thanks for his advice and encouragement during this project.

Credit is due Mr. Harold Tonsing and the Drafting Room Staff of the Electronic Systems Laboratory for their work in preparing the drawings and photographs in this report.

Special appreciation is extended to Mrs. Laurel Retajczyk for her assistance in the typing of this thesis.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

CHAPTER I

BACKGROUND

A. USE OF REMOTE TERMINALS IN TIME-SHARED COMPUTER SYSTEMS

Before time-shared computer systems were built, the standard procedure for using a computer for problem solving was to write a program, transfer it to a deck of punched cards, submit it to a central computer center, and after a wait of five to twenty-four hours (while the program was processed along with many others), the problem solution or "print-out" was picked up at the same central station. Because it was seldom that a program worked correctly the first time, the program print-out usually needed to be quite extensive to discover exactly what the program did.

It was long obvious that the incredible speed of computers was not being well utilized because of the bottle-neck created by trying to get information into and out of the machine. It also became obvious that there were a number of interesting things a computer could be used for that would require a faster information channel between the user and the computer; i.e., one that could give responses within seconds rather than the usual hours.

At this point, the concept of time-sharing was introduced and several experimental computer systems were set up.[1] In general, these consisted of connecting a number of teletypewriters to a computer and having the computer service the commands entered from these "remote terminals" so quickly that it appeared to each user at a keyboard that he was the only one using the computer. These teletypes could properly be called remote terminals

[1] Superscripts refer to numbered items in the Bibliography.

because they were connected through the standard telephone system and could be located almost anywhere--classrooms, offices, homes, even distant countries.

The fast computer response greatly reduced wasted time by the user since he could quickly find and correct mistakes and otherwise "de-bug" programs. It also reduced the need for huge amounts of essentially useless print-out, for the user could trace his program step by step. This was very fortunate also since the teletypes used as remote terminals could not print nearly as fast as the large line printers used in computer centers.

However, even though the use of teletypes greatly diminished the man-computer information barrier, it was felt by many that a more desirable remote computer terminal needed to be developed. There were two major reasons for this. Because of inherent mechanical limitations, the teletypewriters (and similar electric typewriters) cannot print faster than about ten characters per second. This is far slower than the reading speed of a user and if the user is simply searching for some piece of information, this slow rate can be quite annoying. In addition, it was felt that to limit the computer output to only the alphanumeric characters that can be printed was too severe a restriction; pictorial information such as graphs, charts, and drawings would often be the output preferred by a user.

B. DEVELOPMENT OF PICTORIAL DISPLAY TERMINALS

For some years, cathode ray tube (CRT) display units have been attached to computers. The early units were "point-plotting" machines that required the computer to continuously supply information concerning the beam position. These machines were slow; a typical delay of 35 microseconds was required to plot each point. Thus a complicated display could monopolize the entire capacity of even a large computer and the image output would "flicker" because of the slow display rate. Even so, some very interesting projects were done using such displays.[2] These projects encouraged work to develop better terminals.

A good example of a highly versatile display station is the ESL Display Console located at Technology Square as part of M.I.T.'s Project MAC.[3,4] It was developed to produce flicker-free displays that would not require an entire computer to maintain. This station consists of a special-purpose display generating computer with two sixteen-inch CRT's, and is connected to Project MAC's IBM 7094 computer by a high-speed data link. The display logic accepts digital commands from the 7094 which cause it to generate the proper analog voltages to drive the CRT displays to make meaningful patterns.

For text display, a special "character generator" is included; this unit accepts six-bit binary words to specify which of sixty-four alphanumeric symbols is desired and then by referring to its internal memory, generates the proper voltages to cause the CRT to trace out the character.

This station also has a number of other features to facilitate making interesting and useful displays.

In a sense, this station respresents an ideal remote computer terminal and, because of its versatile display, has gained wide acceptance from such people as civil engineers, electrical engineers, mathematicians, and even biologists.[5] Normally, these people are reluctant to spend large amounts of time converting standard computer print-out into meaningful forms--pictures, graphs, etc. With the ESL Console, this is no problem. However, certain factors prevent the wide-scale use of such sophisticated displays as general purpose remote terminals. Expense is one. In the overall M.I.T. time-sharing system, there are more than 200 terminals and it is likely the number will increase. If a complex and expensive machine such as the ESL Console were to be used as replacements for all existing terminals, the cost of the terminals would far exceed the cost of the main computer.

Another drawback is that a high-speed, multi-circuit data link must be used between the ESL Console and the main computer; this would restrict the number of locations where the machine could be used. A third problem is that the present system requires a certain amount of main computer memory and some processor time to maintain a display; a large number of such displays would place a heavy and non-productive load on the central computer system. It appears that a limited number of these highly sophisticated displays will be useful in a computer system but cannot be used as general replacements for teletypes.

C. STUDY OF REQUIREMENTS OF A REMOTE TERMINAL AND SOME EARLY EXPERIMENTS

1. General Considerations

Fortunately, most computer users would be satisfied with a computer terminal which had characteristics somewhere between the low-speed, text-only, format of the teletypewriters and the versatility of the complex ESL Display Console. Realizing this, attempts have been made to design a compromise system.

With a somewhat informal study of the needs of the typical user and an insight into some of the physical limitations of building remote terminals, a better idea of what a general purpose remote terminal should be has evolved.[6] What appears to be needed is a unit that can display both graphical information and text, with text display much faster than present teletype speeds. In addition, the total terminal cost must be low enough to allow its use in large numbers. To be able to locate the terminals as freely as they are presently, the data link connecting the terminal to the computer should be a telephone line.

2. Limitations of Existing Approaches

A number of manufacturers have developed so called "inquiry units" for use as remote computer terminals. Generally, these relatively low-cost machines can be driven over telephone lines and utilize a CRT for display, but can only generate text and very limited graphical displays in a typewriter-like format; they could be called "electronic typewriters". Even though these units can display text rapidly, the fact that they lack the ability to provide flexible pictoral displays make them unsuitable for general purpose remote terminal.

3. Potentialities of Direct View Storage Tubes

At the present time, only CRT's offer the speed, flexibility, and relatively low cost needed for high-performance terminals. However, one of the big problems of CRT's is that, in general, the data displayed must be regenerated and fed to the screen at least 30 times a second to avoid annoying image flicker. To do this, some sort of high-speed memory is needed--and this is usually quite expensive. Also, very high speed electronics and deflection systems are required. On the other hand, there is one class of CRT's known as direct view storage CRT's, in which an image traced by the electron beam is visually stored on the screen and will remain for a considerable period of time (up to one hour) unless it is erased.[7]

Using such a tube, it is not necessary to regenerate data; thus no high-speed data memory is needed and writing speed does not have to be above the flicker rate. Certain disadvantages do occur in terms of resolution, brightness, and screen size with available tubes, but it appears that enough research is being done on these problems that in the near future, sufficiently good storage CRT's will be available.[8]

Because the advantages of storage CRT's in not requiring a high-speed memory and associated high-speed picture generation hardware, the ESL Display Group has employed a Tektronix 564 Storage Oscilloscope in several tests of remote terminals. These tests have been designed to discover the best means of transmitting graphical data over the narrow bandwidth of telephone lines.

Fortunately, besides not requiring a regeneration memory, storage CRT's allow an image to be "built up" slowly from a narrow bandwidth input without any complicated circuitry. Therefore, one of the simpliest schemes for using a storage CRT at a remote location is to send the horizontal, vertical, and intensity control signals along three separate telephone lines to the CRT. These signals cause the electron beam to trace out an image. This arrangement has been tried using the ESL Console to generate the needed signals but suffers from three restrictions.

First, because three signals are required simultaneously to control the CRT, three lines are needed for operation. (For a remote CRT display to be as flexible as a standard teletype terminal, it is desirable that it require only one line.) Second, picture quality suffers because of different delays on different lines and because of line noise. Third, because of bandwidth limitations on the telephone lines (approximately 1 Kc), the writing rate, particularly for text displays is too slow.

D. IMPROVED REMOTE DISPLAY TERMINAL

In an effort to overcome the problems outlined above, it was suggested that a remote storage-tube terminal be built with local picture generating capability, somewhat like the ESL Console.[9] Data could then be sent in digital form, which is more noise immune than analog transmission. Horizontal, vertical, and intensity control signals could be sent sequentially and stored at the terminal end until all three were available--thus requiring only one telephone line.

Finally, instead of continuously sending the absolute beam location to the terminal, only changes in beam location would be sent,thus reducing the amount of redundant information sent and effectively increasing writing speed.

These techniques increase the amount of reliable data sent over a communication channel in a given time period but require considerable data processing at the receiver end of the channel. Therefore, the question arose as to whether it would be possible to build the needed circuitry at a low enough cost to be reasonable in a remote terminal. Thus, a design project was undertaken to implement the scheme shown in the block diagram of Fig. 1. This equipment, except for the character generator, was built and tested in the summer of 1965.

Briefly, operation of the vector generator is as follows. Data is received serially in digital form from a single input line. A nine-bit data buffer register stores the incoming signal until it is full and then rapidly transfers its contents to either the x, y, or

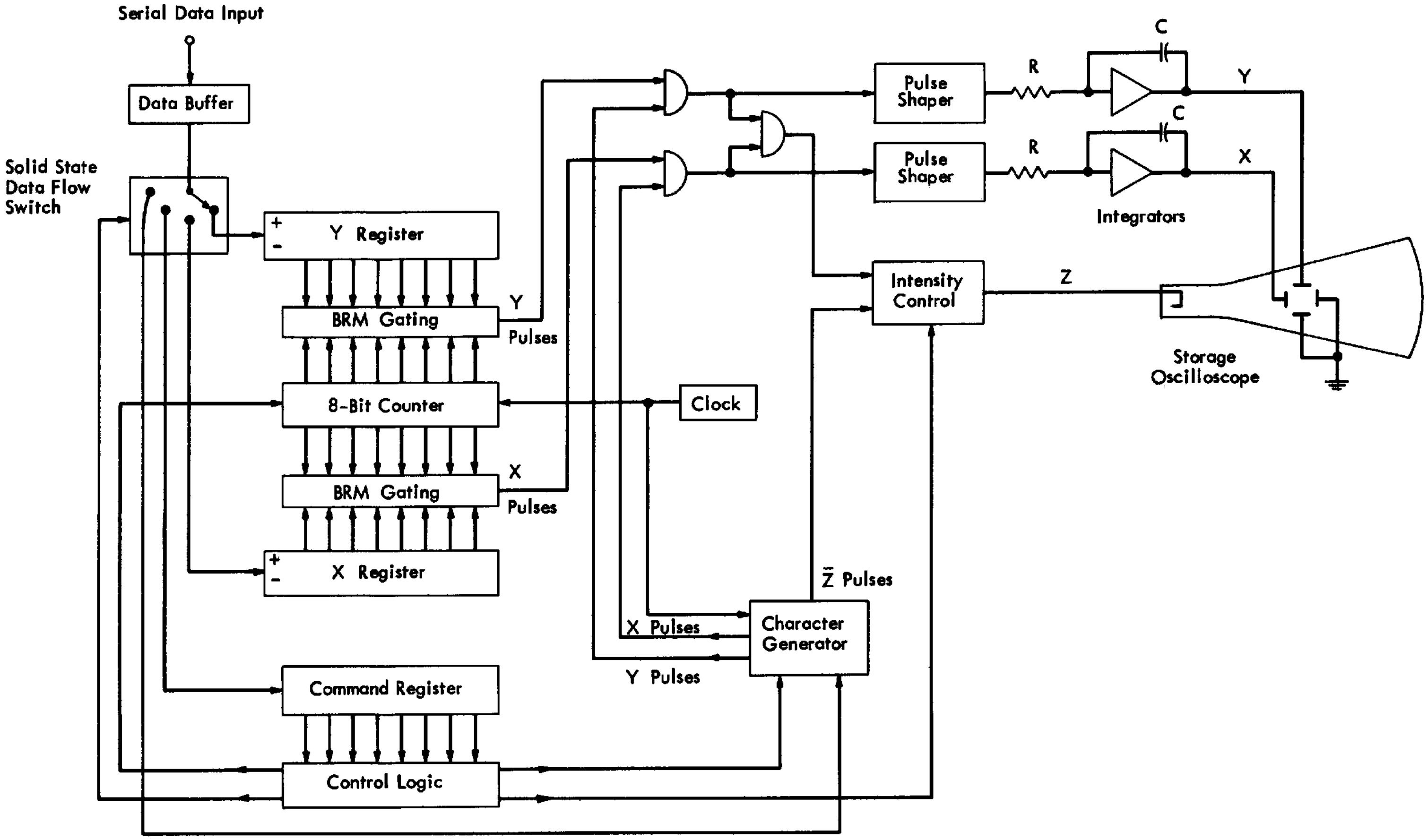


Fig. 1. Remote Display Console with Vector Generator and Character Generator

command flip-flop registers, depending on the position of the data flow switch. The three registers are filled sequentially, starting with the command register. When all are filled, the eight-bit counter associated with the x and y registers is allowed to count. Two binary rate multipliers (BRM's) are controlled by the data in the x and y registers. The output of each BRM is a pulse train containing as many pulses as the binary count in the associated register.

The two pulse trains are used to trigger special pulse generators that put out precise pulses with constant width and constant amplitude. These pulses feed operational amplifiers connected as integrators. The output voltage levels of the amplifiers change as each incoming pulse is integrated. These voltages are used to deflect the beam on the storage CRT. After each new voltage increment appears on one of the integrating capacitors, an intensify pulse is generated and a dot appears on the CRT screen. Therefore, a pulse train produces a series of closely spaced dots that define a straight line. The angle and length of this line depend on the relative magnitudes of the binary numbers in the x and y registers. Also, since the pulse generator can produce either positive or negative pulses, lines may be drawn in any direction. After a line is complete, the electron beam remains at its last location until three more binary words are loaded. At this point, a new line is initiated, starting where the last one left off. By continuing in this fashion, line drawings of any complexity can be made up.

In addition to lines, the command register can be used to program the control logic to reset the integrators to zero, to disable the intensify circuits, and perform other special tasks. Lines are drawn as a series of points spaced about 0.005 inches apart (on a 5" CRT); points can be plotted at a rate of one every twenty microseconds. Presently, a "word" of nine bits can be received in five milliseconds, and since three words must be received in order to draw a line, one line can be drawn every fifteen milliseconds.

E. AN ADDITIONAL PROBLEM

It was known from the beginning that the vector generator described in Section D would not be able to effeciently handle text display. Even though characters could be drawn using short line segments, this method would be much too slow; characters would need five to ten line segments--requiring 75 to 150 milliseconds per character. This is roughly the speed of a teletypewriter.

The solution seemed to be to include a character generator in the remote terminal. As in the ESL Console, this generator would accept a single binary word to specify a desired character. Then, by referring to an internal memory, the machine would produce the appropriate voltages to cause the image of the character to appear on the CRT. Since one word is received every five milliseconds, 200 characters per second could be written on the CRT--20 times as fast as a teletypewriter.

It was this need for a character generator that would meet the special requirements of a versatile, yet low-cost remote computer terminal that motivated this thesis.

In Chapter II, a review is given of existing character generation schemes; some advantages and disadvantages of each type are given. Chapter III outlines in detail how the final generation technique was chosen while Chapter IV discusses actual circuit construction. Finally, Chapter V gives some conclusions about this project.

CHAPTER II

CHARACTER GENERATOR TECHNIQUES

A. INTRODUCTION

After the decision was made that it was necessary to include a character generator in the Remote Display Terminal, it was then necessary to investigate the various ways of building such generators and to see what machines were commercially available.

It was soon discovered that a large variety of generation schemes exist--each having certain advantages and disadvantages.[10] To compare the existing generation techniques, it is desirable to identify the elements and operating parameters that are common to all. There are two basic elements associated with character generators: 1) generating logic and 2) character set memory.

In the simplest terms, it is the function of the generating logic to take information from the character set memory and use it to form a character on the screen of a CRT.

The character set memory stores information about the shape of characters so that when a six- or seven-bit binary word is accepted as a character selection input, the memory "programs" the generating logic to produce the desired character.

Usually, the generating logic produces horizontal and vertical deflection voltages to move the electron beam of a CRT; often a beam intensity voltage is also produced.

Important parameters of character generators are: rate of generation (characters per second), quality of characters, size of character set, reliability, and, of course, cost. For some applications, such other factors as physical size and ease of changing the character set are also important. As a rule, as one or more parameters improve, others suffer. Cost in particular is likely to increase as speed, quality, and character set size increase.

B. GENERATION TECHNIQUES

The different approaches to character generation fall into two basic classes--signal generation and beam forming.[11] The signal generation schemes use the kind of generating logic outlined earlier; vertical and horizontal deflection signals and beam intensity control signals are produced to control the electron beam in the usual way to trace out characters.

Beam forming is a very different method. With beam forming, the cross sectional area of the electron beam is distorted in such a way that when the beam strikes the phosphor screen, it forms a character. This is usually accomplished by passing the electron beam through a mechanical mask located within the CRT. Though the beam-forming generator--really a special purpose CRT--has some advantages in systems requiring high-speed character generation, it is not well suited for use in a remote terminal. The CRT used is very specialized and rather expensive, and since the CRT is one of the most likely elements to age and need replacing, this is a serious drawback. In addition, complex circuitry is needed for the additional deflection stage needed to pass the electron beam through the proper hole in the character mask, and then re-align it on-axis for the main deflection system.

Turning again to the signal generation schemes for character formation, we find that there are two main groupings: raster types and function types. In the raster types, a standard deflection pattern or "raster" is generated for all characters; this pattern is designed so that by selectively intensifying certain portions of the pattern, any character required can be displayed. The pattern used may be a rectangular array of dots, a series of vertical or horizontal lines, or a lissajous type figure. The advantage of this method is that by using the same raster for all characters, the vertical and horizontal deflection voltages are always generated the same way and no memory is required to program these voltages individually for each character. Thus the memory contains only intensification data. This reduces memory capacity requirements and lowers machine cost.

The reduced memory requirements are partially off-set by the need for a relatively complex raster generator which has to be designed to go through the required sequence of horizontal and vertical deflection voltages to produce the raster pattern. The disadvantages of such a system are that generally many unnecessary beam movements are required, reducing character repetition rate. Also, character quality (readability) may be low and the characters can have a rather mechanical look to them, unless a large raster is used.

However, if a large raster is used, the character set memory size must be increased and the advantage of a lower cost memory is lost.

The sports scoreboards that use a number of lights to display numbers are an example of a minimal raster display.

The functional generators avoid the wasted motions of the raster generators by programming the deflection voltages for each individual character, tracing out the character directly. This is a considerably faster technique and produces very good quality characters, but also requires a much larger memory--usually vertical, horizontal, and intensity signals are stored. On the other hand, the vertical and horizontal drive circuits are somewhat simpler than the raster types. The raster circuit has to "remember" the raster pattern, whereas the function type is continuously controlled by the memory.

Several different versions of the function character generators exist. In some, horizontal and vertical deflection voltages are stored in pairs: the generating logic sequentially samples these pairs and the electron beam jumps to a new position each time a new pair is sampled. Usually, the number of pairs is limited to less than sixteen, but this is seldom a restriction on the types of characters that can be generated.

Another type of functional generator is the "stroke" generator, in which characters are drawn as a sequence of short lines. The memory supplies the orientation of each of the lines. Normally, eight possible orientations are stored and nine to twenty line segments are used.

In addition to the basic techniques outlined above, a number of variations exist that have certain advantages for certain applications.

C. CHARACTER SET MEMORIES

Another way of distinguishing between different types of character generators is to compare the types of memories used. As with generation schemes, one finds that many different types of character memories. Information can be stored in either analog or digital form and can be stored for either serial or parallel read-out.

1. Read-Write Memories

The same type of ferrite core memories that are widely used in computers have been used for character generators also. Such a memory can be "written" as well as read and thus offers the advantage of being easily changed to hold different character sets. However, for memories as small as character sets, normal read/write core memories are not economical when compared to other types. In addition, for many character generator applications, they are too slow. Finally, the fact that this memory is not a permanent memory means that some provision for initially "filling" it must be provided.

Sequential memories can be used for character pattern storage in some types of character generators,though for high speed generators, they are too slow. Sonic and magnetostrictive delay lines are both possible choices for this type of memory, and can hold fairly large amounts of information at low cost per bit.

Along with standard core memories, however, sequential memories share the mixed blessing of being "re-writable". This allows easy changing of character set but requires means of periodically filling the memory. In the case of delay lines, a re-filling is necessary whenever power to the unit is interrupted for any reason.

Since information is only available serially from such memories, speed of read-out is very limited and complicated circuitry may be required.

2. Read-Only Memories

Because of the various drawbacks to read-write memories, character generators usually incorporate "read-only" memories which combine reasonable cost, high speed, and permanent storage. As the name implies, new information cannot easily be entered into such memories, so character sets are fixed.

Included in this class are the various matrix memories.[12] In general, these consist of a matrix with a series of horizontal "word" lines and a series of vertical "bit" lines. When used in a character generator, a word line corresponds to a character in the set. The memory is normally binary, storing a pattern of ones and zeros for each character. To store a "one" at a particular bit location of a particular word, a passive circuit element is connected at the junction of the word and bit lines. To store a "zero", no element is connected.

The memory is read by applying a voltage to one of the word lines, and monitoring the output of all the bit lines. A high voltage indicates a "one", and a low voltage indicates a "zero". Circuit elements that can be used are diodes, resistors, capacitors, or inductors.

One ferrite-core memory that overcomes some of the problems of read-write memories is the "rope memory". This consists of a small number (one for each bit of the storage word length) of relatively large aperture ferrite cores that have a large number of wires (one for each stored word)threaded through them. An individual wire is woven through the series of cores in such a manner that it may or may not pass through a given core. If a "write" current pulse is sent through this wire, all the cores through which it passes are set to the "one" state while the others remain in a "zero" state. The pattern of ones and zeros can then be read out by appropriate circuitry. Normally, for each character in a set, there is a wire that is woven through the series of cores. In one standard configuration, 64 wires are woven through 35 cores. When any one of the 64 wires is pulsed, a particular pattern of ones and zeros is set up in the 35 cores; this pattern is then used to program the generating logic to produce a character.

Since the character patterns are determined by the wire threading, there is no need to fill the memory (but it is no longer possible to change character sets, unless the cores are re-threaded). Also, because no coincident-current selection is involved, read-out signal levels can be much higher than for standard core memories and less expensive sense amplifiers are required. In general, rope memories offer a less flexible but more economical character storage for character generators than normal core memories.

3. Special Purpose Memories

The preceding memory techniques are general-purpose in the sense that they are basically digital in nature and any type of information could be stored in them. However, some special memory techniques have been developed especially for character generation. One example has already been discussed; the mechanical mask used in beam-forming CRT's is the memory for that type of generator.

Another interesting example employs a special T.V. "camera" tube, called a monoscope, in a raster type generator. Instead of picking up an external field of view as normal vidicons, the monoscope has images or "targets" built inside its faceplate. If these targets are shaped like characters, they constitute the character memory. In use, the electron beam of the monoscope is trained at the desired character; then the electron beams of both the monoscope and the display CRT are synchronously swept through a series of short vertical or horizontal lines. The beam intensity of the display CRT is controlled by the monoscope signal output, which in turn is controlled by the particular character that is being swept.

D. SUMMARY

The large number of commercially available character generators range in price from about $2000 to more than $10,000, depending on which features they emphasize. A good example of such a machine is the Straza Character Generator used with the ESL Console. It has a repertoire of 64 high-quality characters, and can generate them at a rate of 100,000 per second.[13] However, it costs more than $8000. Since it has been suggested that the total cost of the entire remote terminal be no more than $5000,[14] (including vector generator, CRT display tube, and other required hardware <u>plus</u> a character generator), it appeared that a new type of character generator--less expensive than existing ones-- would have to be developed. Several factors suggested that an original design would be desirable.

First, it was noted that most of the commercial machines were designed to be used with CRT's that require the displayed picture be refreshed continually. For a text display of 1000 characters, the character generator used with such a CRT must produce 30,000 characters a second if the display is to be refreshed at least 30 times a second (to avoid image flicker). However, since the proposed remote terminal is to use a storage CRT, no regeneration is required and the character generator need produce only 200 characters per second. It was hoped that this lower speed requirement would help reduce costs. Another reason for designing a new machine was that Project MAC has standarized on the 7-bit Revised ASCII Character Set (which has 96 printable characters) for future terminals.[15] Few commercial machines offer character sets larger than 64 without essentially paralleling 64-character units, with attendant large increase in price. Some cannot be used with sets larger than 64 without extensive modifications.

For these reasons, it was decided to choose a design approach for building a new type of character generator that would meet the performance requirements of a remote computer terminal and be inexpensive enough to be compatible with the cost goals of such a terminal.

CHAPTER III

SELECTION OF A NEW DESIGN APPROACH

As discussed in Chapter II, study of the various techniques used in character generation indicated that a fresh design approach was needed. In this chapter, we will consider the advantages and disadvantages of several possible types of low-cost character memories. Reasons for choosing a resistor matrix memory are discussed, and a general outline of a proposed raster-type generating logic technique is given. Particular emphasis is given to compatability with the vector generator and possibilities for sharing circuit elements.

A. REVIEW OF REMOTE TERMINAL REQUIREMENTS

The requirements included use the revised ASCII character code with 96 printable characters, and the ability to generate 200 characters per second. Other desirable characteristics were that several format sizes for characters could be generated and that special, non-standard characters could be externally programmed. There was no particular concern about character quality other than that the characters be easily legible. Above all, the generator must be of low cost; initially, it was decided that a figure of $500 to $1000 would be considered acceptable. Of course, any design would have to be compatable with the vector generation scheme as outlined in Chapter I.

B. A LOW-COST, MECHANICALLY ROTATING MEMORY

As was pointed out earlier, one of the most obvious reasons for the high cost of commercial character generators is that these machines are usually expected to work at high speeds--generating 30,000 to 100,000 characters a second--so that CRT image regeneration can take place. Since the generator required for the proposed remote terminal need only produce 200 characters per second, it was hoped that by sacrificing high speed, an inexpensive machine could be built.

One place where a lower speed requirement could result in a substantial savings was in the character set memory. It was suggested that a low-speed, mechanically-rotating, sequential memory could perhaps be used.

It was hoped that such a memory would be very inexpensive to construct. In addition to low cost for the memory itself, the fact that such a memory would be sequential would be an additional advantage over parallel memories. This is true because in all signal generating type character generators, information to control the CRT electron beam must be supplied to the generating logic in serial form. Therefore, for parallel read-out memories, some sort of buffering, such as a shift register, is necessary to convert the memory output to serial form. With a sequential memory, however, no buffering is needed and the cost of this additional circuitry is avoided.

One possible sequential memory for use in a character generator is a magnetic drum, but after investigating these memories, it was found that the commercial units available were too expensive to be used--mainly because they stressed a read-write capability and large bit-capacity.

To find a lower cost substitute, it was decided to consider building a special rotating disk memory that would be limited to read-only capability and have a small-bit capacity. One plan was to use a photographic disk with information recorded optically on co-axial tracks along its circumference. Another possibility was to use magnetic or electrostatic recording as storage modes.

This memory disk would be driven by a motor much like a phonograph record, and a number of "read stations" would be mounted near the disk to convert the stored information to electrical signals.

As with any sequential, circulating memory, this proposed disk system presented problems concerning access time and read-out time. Because of limitations on the writing speed of the storage CRT that will be used, a small-size (0.1 inch high) character cannot be written in less than roughly 300 microseconds; for large-size (0.4 inch high) characters, this writing time cannot be less than 1000 microseconds (one millisecond).

To avoid the cost of extra circuitry, it was necessary to use a minimum of data buffering--both on the memory selection logic and on memory read-out. This meant that the one-millisecond CRT writing rate would be the required memory read-out rate also. On the other hand, incoming character commands occur every five milliseconds, and with a one-millisecond read-out time, the maximum allowable access time would be four milliseconds.

During the four milliseconds, the entire character set memory stored on the disk would have to pass under a memory read station. The faster the disk rotation, the less time is required for a given point on the disk to reach the read station. However, if a single read station is used, a complete rotation in four milliseconds requires the disk to rotate at 15,000 RPM--a rather high speed for mechanical systems. If four reading stations were placed at ninety degree intervals around the disk, a speed of only 3750 RPM would be required for a given point to reach the nearest station in a maximum of four milliseconds. Alternately, if the character set memory were copied several times around the disk, rotational speeds could be reduced. However, bit-packing densities limit how many times the memory may be copied.

The need was, of course, to use as few reading stations as possible to reduce costs, and yet keep the rotational rate at a reasonable value.

In addition to character-set storage, timing and address tracks (which would also require associated read stations) would have to be included on the disk.

One attractive possibility for this type of memory was to store analog signals with a technique much like sound tracks are recorded on motion picture film; by using three tracks, the horizontal, vertical, and intensity control signals could be recorded directly. This would require very little circuitry for the generating logic and character quality could be high. On the other hand, a large number of read stations would be needed.

Another technique that would require considerably less memory capacity (and therefore fewer read stations and switching circuitry) would be to use a minimal raster approach. This would also allow character data to be stored digitally. Of course, the generating logic would be more complex.

The major difficulty with the disk memory approach is that it would be mechanical and would have mechanical limitations. For a high-speed disk system, a well machined housing and drive motor would be needed. Alignment of read stations and other mechanical adjustments would periodically be required and in general, reliability would probably not be as good as an all-electronic character generator. Thus attention was turned to non-mechanical approaches.

C. ALL-ELECTRONIC MEMORIES

In studying various types of purely electronic memories that might compete economically with the mechanical disk type, it was necessary to price not only the memory itself, but also the sensing (read-out) circuitry and the selection logic.

For delay line memories, selection logic can be quite trivial; a counter driven by a precise clocking signal can keep track of which word is available at the output.

The read-out amplifier is relatively complex, but only one such amplifier is needed. However, the cost of the overall memory, the problems of refilling it (as mentioned in Chapter II), and the problem of access time made this type of memory seem unattractive.

Ferrite core "rope memories" offer certain advantages in terms of random-access reading (no access-time problems), read-out levels, and selection logic, but the quoted cost of $350 to $500 (without selection logic) was somewhat higher than was thought desirable.

The special-purpose character memories such as beam forming type and the monoscope type, while feasible, were felt to be ill-suited for use in the proposed remote console. They would not fit in well with the vector generator, and would require special power supplies and other control circuitry which would make them much too expensive.

Various matrix memories were considered: inductive, resistive, diode, and capacitive elements can be used in such read-only memory arrays. Inductive arrays offer advantages in ease of word selection and can have (in some applications) high enough output signal levels to eliminate the need for sense amplifiers. However, such arrays are difficult to fabricate and are therefore more expensive than other arrays. At the opposite extreme, capacitive arrays are quite easy to fabricate but have low output levels.[16]

Diode matrix arrays are straightforward to work with, and have excellent output levels. However, at the time the memory survey was made, there were no commercial techniques for economically producing large diode arrays. Early calculations showed that a minimum character memory would require between 2000 and 3000 bits--this would mean using roughly 1000 to 1500 diodes (assuming that about half of each stored character word consists of "ones"). It appeared that the price of components and installation would be between $.15 and $.20 per diode; therefore memory component cost, not counting selection logic or sense amplifiers would be as high as $300.

Resistor arrays do not offer nearly as well defined output levels as a diode array because of "sneak paths" through unselected words. Moveover, as the array size increases, the output levels decrease rapidly. On the other hand, resistors are cheap--typically no more than $.03 a piece in quantity. Parts cost for a 3000-bit memory would be roughly $45.00. In addition, it was known that some work has been done to fabricate low-cost arrays of resistors, using deposited film techniques.[17]

The promise of a very low-cost resistor array was encouraging enough to motivate research into how big a disadvantage the low output levels of such a memory would be. Initial investigation showed that while rather small output levels were available, the low speed requirements made it possible to use inexpensive circuits to act as sense amplifiers. Selection logic also could be built cheaply. When compared to a possible diode matrix that would require less complex sense amplifiers, indications were that the savings on resistor memory itself would more than pay for the higher circuit costs of the sense amplifiers for the resistor matrix. In addition, the overall costs of a memory made with ordinary techniques appeared to be competitive with the proposed disk memory; if it were found that resistor matrix could be fabricated automatically, its costs would probably be much less.

However, there was one big drawback to using a matrix memory: read-out from such a memory is essentially parallel and, as pointed out earlier, memory data must be fed to the generating logic serially. Therefore, expensive buffering must be used with a parallel read-out memory. To remove this restriction on the use of parallel memories, it was decided to "share" some of the circuitry in the vector generator to perform the buffering function. By sharing already existing circuits, system cost does not increase; thus parallel read-out memories can be used without excessive cost.

A side advantage of using a buffer for the memory is that this buffer can be filled not only by the character memory but also from the external data input. This allows the character generator to work in a special mode in which non-standard symbols can be externally programmed.

Because of the encouraging economic comparison and because a character generator using a resistor matrix memory would be compatable with the vector generator and system timing considerations, it was decided to build an experimental character generator using a resistor matrix as its character-set memory.

D. GENERATING LOGIC

The type of generating logic to use depends, of course, on which type of character set memory is chosen and how much emphasis is placed on character quality. Since it was decided to use a resistor-matrix, which is a purely digital memory, and since character quality was not of prime concern, the obvious choice for the generating logic was to use a minimal raster technique.

The normal mode of the vector generator is to draw lines by a series of points or dots. Thus, the most compatable raster scheme to work with this generator is a rectangular array of dots. The simpliest array that will still give the needed readability for all characters is an array of 35 dots--seven dots high, five dots wide. All characters can be made up from such an array by selectively intensifying specific points. An example of such a dot array or matrix is illustrated in the upper right-hand corner of Fig. 2.

A larger array--say nine-by-six or ten-by-seven-- would produce better looking characters but would require considerably more memory capacity; a smaller array would not provide enough detail to distinguish clearly between characters.

Figure 2 shows in block diagram form the make-up of the proposed seven-by-five dot matrix character generator. Three main divisions are identified; a dot matrix generator, a character memory, and a digital-to-analog converter.

Since the character generator and vector generator are never used at the same time, it was natural to consider using the same circuits to serve both generators. Thus, it was hoped that by time-sharing common circuit functions, considerable savings in cost of circuits would result.

Because of this desire to time-share circuits, the elements in Fig. 2 were chosen to maximize the similarity of certain circuit blocks of the vector generator and of the character generator. It will be noted that the digital-to-analog converter is identical to the pulse integrator circuits of the vecot generator. Also, that the Modulo-5 and Modulo-7 counters could be used as part of the eight-bit counter used in the vector generator.

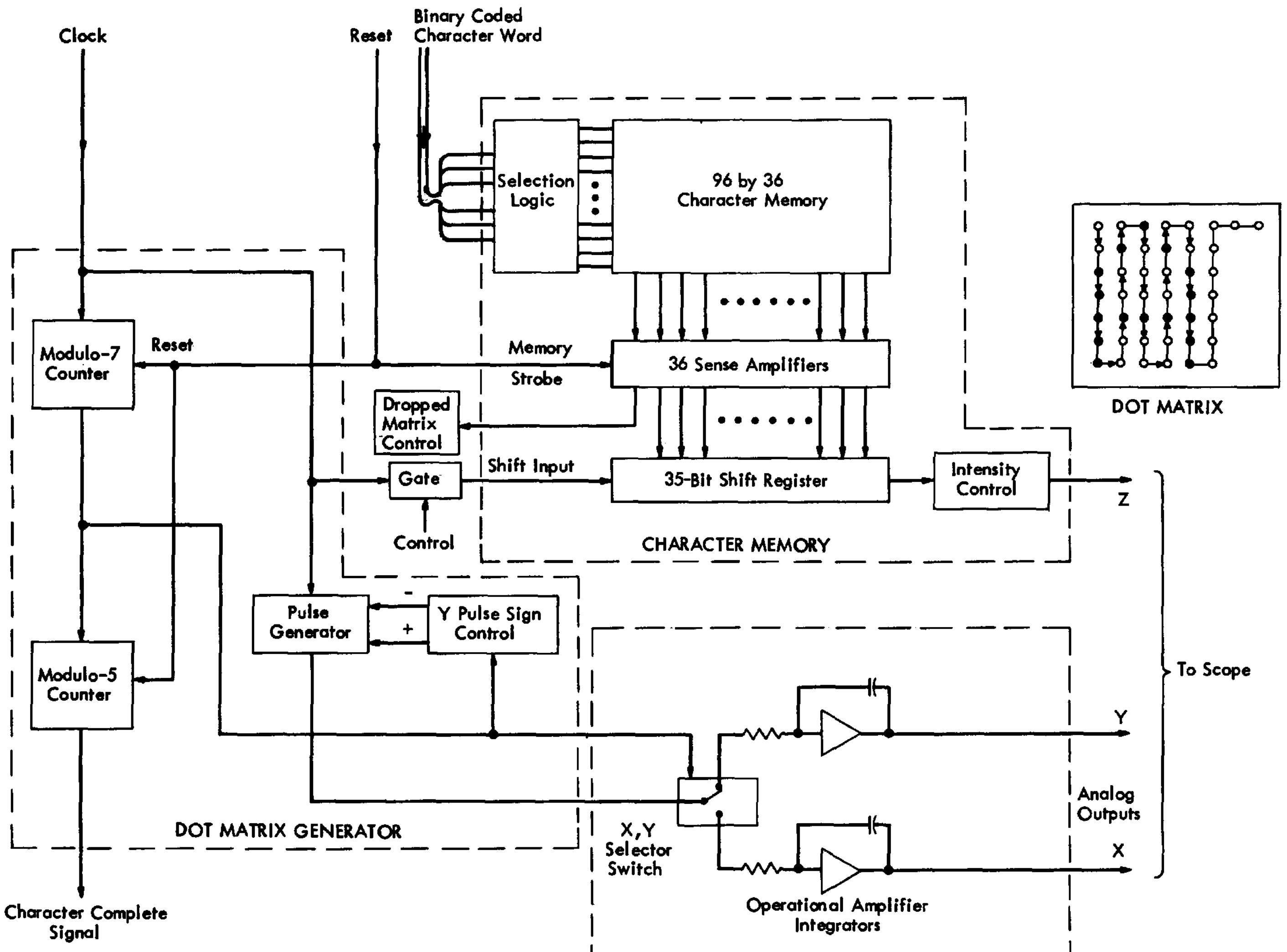


Fig. 2. Block Diagram of Character Generator

In operation, an 8-bit binary word fed to the selection logic is decoded to specify one of the 96 possible character word lines. Thirty-six bit output lines feed the digital word containing the intensity information of the selected character into 36 sense amplifiers. When a "reset" pulse is applied to the character generator, the 35-bits of the character-word are read into the 35-bit shift register. (The thirty-sixth bit is a special control bit which will be discussed later.) After the shift register is loaded, clock pulses feed both the dot matrix generator and the shift register; the matrix generator produces horizontal and vertical deflection signals to. move the electron beam through the pattern shown in the upper right-hand corner of the figure.

In synchronization with the pulses that move the beam, the pattern of "ones" and "zeros" stored in the shift register are shifted to the right to sequentially feed the Intensity Control Circuit. In this way, the information in the character memory is used to control which points in the generated matrix are intensified.

For simplicity, some of the needed circuitry is not shown. For example, provision is made for an additional nine clock pulses (which are never intensified) to move the electron beam into position to draw another character. Also, certain lower-case characters, such as "g", "j", "p", "q", and "y", extend below the base line upon which all the other characters are written and thus lie outside the standard matrix. To draw these characters, it is necessary to start the matrix two points lower than usual. Therefore, the thirty-sixth bit, called the Dropped Matrix Control bit, starts the matrix at the lower position; it is used only for characters that extend below the base line.

The modulo-5 and modulo-7 counters keep track of where the beam is in the x and y directions, respectively. After each set of seven counts in the y direction, the beam is incremented in the horizontal (x) axis by one step and the direction of the beam motion in the vertical (y) axis is reversed. When the modulo-5 counter is full (after five x increments), a signal is generated to stop the clock. When a new character is loaded, the process beings again.

It would have been possible to read out the memory word in segments--say six six-bit blocks--and use only a six-bit shift register instead of a 35-bit one. However, it turns out the x, y, and command registers of the vector generator (shown in Fig. 1) can be built as shift registers; to do so requires a shift register totaling 27 bits long. Therefore, by "time-sharing" the 35-bit shift register between the character generator and the three registers of the vector generator, there is little waste involved in using a 35-bit shift register, and the complex switching involved in reading out segments is avoided.

Since all the major circuit elements except the memory and its selection logic can be shared, they are in a sense "free". Therefore, the cost of the character generator depends mainly on the cost of the selection logic and memory, plus the cost of the circuitry needed to switch the function of the various major circuit elements.

CHAPTER IV

DESIGN OF THE EXPERIMENTAL CHARACTER GENERATOR AND TEST RESULTS

A. DESIGN APPROACH

After the basic approach of using a resistor matrix memory and a dot raster generator was decided upon, it was time to begin actual circuit design. It was planned to build an experimental character generator in two steps. First, a test generator with a simplified raster and very limited memory size would be built to test all the basic circuits. After that, a full-scale machine with a five-by-seven dot raster and a 96-character memory would be constructed.

Recent reductions in the price of epoxy encapulated integrated circuits have made these miniature circuits competitive (and often cheaper) than comparable circuits made from discrete components. Future price projections show a further downward trend in integrated circuit prices. For this reason, plus the obvious advantages of smaller size, it was decided to use these circuits for most of the construction of the logic design.

The units chosen are made by Fairchild Semiconductor Corporation as part of its commercial Micrologic series of integrated circuits. They use resistor-transistor logic (RTL) and are rated for use up to two megacycles. Three types were used: 1) a type 923 JK Flip Flop with set, reset, toggle, and clear inputs, 2) a type 914 dual NOR gate package with two inputs per gate, and 3) a type 900 driver package which acts as an inverting power driver. All are packaged in eight-lead, modified TO-5 epoxy cases. For normal operation, they require a 3.6 volts power supply and have logic levels of 0 to 0.2 volts for a "zero" and +1.0 to +2.0 volts for a "one". Fan-out and fan-in are very limited on these units, but with some care, the units work very well, and in large measure, they contributed to the success of the project.

For applications where the integrated circuits were not suited, a low-cost epoxy encapsulated silicon transistor was chosen (Type 2N2923). Although it is not intended for use in switching applications, this unit was found to work well in most of the low-speed circuits described here. In a limited number of cases where a good switching transistor was needed, a 2N3569 silicon transistor was used. By having such a small number of different types of circuit elements a number of design problems were eased and procurement of parts was simplified.

Even though the basic design of the character generator was selected so that circuits could be shared between the character and vector generators, it was decided to avoid design problems in construtting the experimental character generator by making it almost indøpendent of the vector generator. However, clock and reset pulses were derived from the vector generator, and the digital-to-analog converters of the vector generator were used to transform the pulse train outputs of the character generator into the required analog deflection voltages. The diagram of Fig. 1 shows the relationship (including inputs and outputs) of the two generators. It is proposed to build at a later time a new character-vector generator that would take advantage of the circuit-sharing techniques mentioned in Chapter III.

As has been mentioned, it was decided as the first part of the project to build a small test character generator to check out all the basic circuits before building a full-scale machine. A character generator using as a raster a three-by-five array of dots and having a character set memory of eight words was chosen. This choice was motivated by the fact that this was the smallest pattern on which the numbers 0 through 7 could be legibly plotted.

To construct such a generator, it was necessary to build a dot-matrix generator and a resistor-matrix memory, including sense amplifiers and a shift register. In addition, because the logic levels of the vector generator were 0 and -3 volts and the levels of the character generator were to be 0 and +1 volt, level converters to couple the two generators were needed. A power supply to drive the circuitry was also required, and since it involved no difficult construction problems, it was decided to build the supply big enough to drive either the test generator or the final generator.

B. DOT MATRIX GENERATOR

As outlined in Chapter III, it is the purpose of the dot-matrix generator to generate signals to cause the electron beam of the storage CRT to move through the desired five-by-seven pattern of points. Figure 2 shows a block diagram describing how this is done, while Fig. 3 shows how the actual integrated circuits were connected to perform this function. (Power supply connections are not shown to avoid unnecessarily complicating the diagram.)

As a review of Fig. 2, we note that there are two counters, a Y-Pulse Sign Control, and a Pulse Generator in the Dot Matrix block. In addition, the x, y data flow switch is controlled by the Matrix Generator.

The Modulo-7 counter counts the number of vertical (y) increments and supplies a pulse to change the direction (or sign) of the vertical movement after every seven pulses. The modulo-5 counter keeps track of the horizontal (x) increments and supplies a "Character Complete" signal after the entire dot pattern is finished.

These two counters are shown in Fig. 3 as the two four-bit counters; the counter to the left corresponds to the modulo-7 counter and the one to the right corresponds to the modulo-5 counter.

Basically, each counter is an ordinary four-bit counter and the two can easily be connected together to form the eight-bit counter needed in the vector generator. To make the left counter work as a modulo-7 counter, a pair of gates located in the "Reset Control" block monitor the state of the counter. When the counter, working in the normal binary manner, reaches a value of seven, a reset pulse is generated by the Reset Control and the counter returns to the zero state. At this point, the counter will start counting again, and continue until a count of seven is again reached, etc.

A similar situation exists for the right-hand (modulo-5) counter. A pair of gates monitor the state of this counter and produce a Character Complete signal when a count of five is reached. However, no reset pulse is generated at the end of five counts since this counter need not reset until a new character is started.

By using this reset-to-zero technique, it is possible to use the counters as ordinary binary (modulo-16) counters by simply disabling the reset circuits. Also, it is very easy to change the counting base of the counters by simply opening or closing the switches indicated on the inputs of the count-monitor gates. In this way, by properly setting six

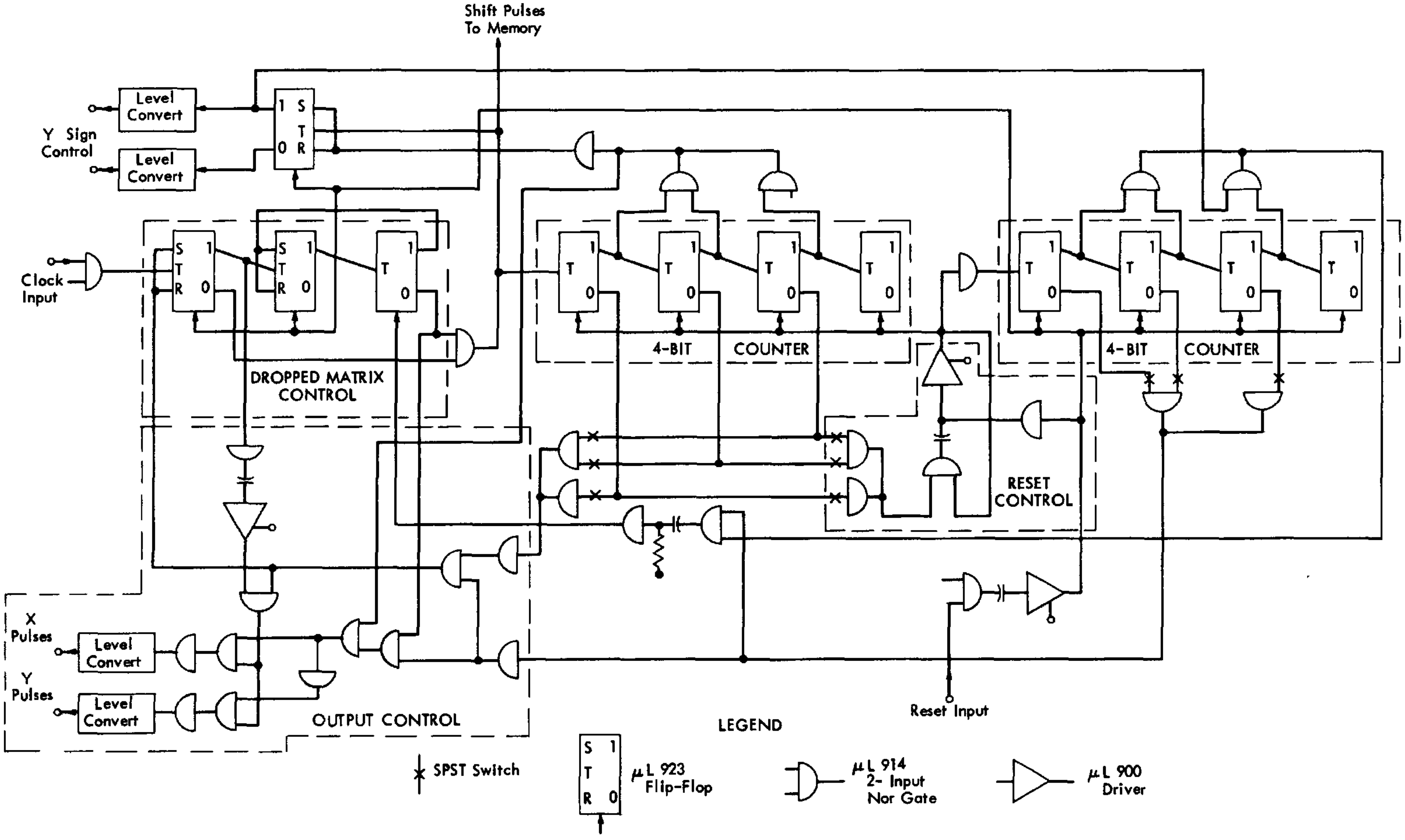


Fig. 3. Dot Matrix Generator

switches (indicated by x's in Fig. 3), it was possible to generate any desired dot array from a one-by-one to a seven-by-seven point matrix. This made it possible to use the same dot matrix generator for both the three-by five test generator and the final five-by-seven generator.

The flip-flop associated with the "Y Sign Control" output is connected so that it will change state every time the modulo-7 counter is reset to zero; also gating associated with the "Output Control" block shunts a pulse that would normally feed the y pulse output to the x pulse output.

There are two additional points which were discussed briefly in conjunction with the simplified diagram of Fig. 2. First, it was decided that the beginning of the design project that it would be desirable, at the completion of each character, to automatically position the electron beam of the CRT so that it would be possible to draw the next character without inserting any additional line commands. In this way, the full channel bandwidth could be used for transmitting characters alone. (Of course, at the ends of text lines, the beam would have to be re-positioned to the start of the next line--somewhat like a carriage-return on a typewriter.)

In a normal five-by-seven array, the final point is plotted at the lower right-hand corner when the first point is plotted at the top left-hand corner. Therefore, some provision must be made to return the beam to the top of the matrix and then move it to the right (to the starting position for the next character). One easy way to do this is to increase the size of the array to six-by-seven; in such an array, the beam ends the pattern one point to the right of the top right-hand corner of the character. Next, two extra pulses are used to force the beam two additional spaces to the right, yielding an inter-character spacing of three dots. Of course, the extra points are never intensified, and therefore require no memory control. The entire matrix pattern is shown in Fig. 2 and also in Fig. 10.

Another problem occurs when the generator is required to plot certain lower-case letters, such as "g", "j", "q", "p", and "y". These letters are unusual in that they extend below the normal base line for alphanumeric symbols. This means that they also extend below the normal dot matrix. A suggested solution to this problem was to start the matrix at a lower point for these characters. Therefore, an extra bit was included in the character set memory to indicate when the matrix should be lowered, and a "Dropped Matrix Control" was included in the matrix generator. This logic circuit adds an extra two pulses to both the beginning and end of the normal matrix. Examples of both the normal matrix and the dropped matrix are shown in Fig. 10,

C. THE RESISTOR MATRIX MEMORY

A section of the proposed resistor matrix memory is shown in the top part of Fig. 4. A word line corresponds to a printable character and the number of bits per word corresponds to the amount of information stored about the shape of each character. The test character generator was constructed with 8 Word Lines and 15 Bit Lines. The final experimental generator was to have 96 Word Lines and 36 Bit Lines.

1. Storage Mechanism

Information is stored in binary form--"ones" and "zeros"--by connecting "bit-resistors" between Word Lines and Bit Lines. A 47 k resistor across a junction indicates storage of a "one"; no resistor indicates a "zero" is stored.

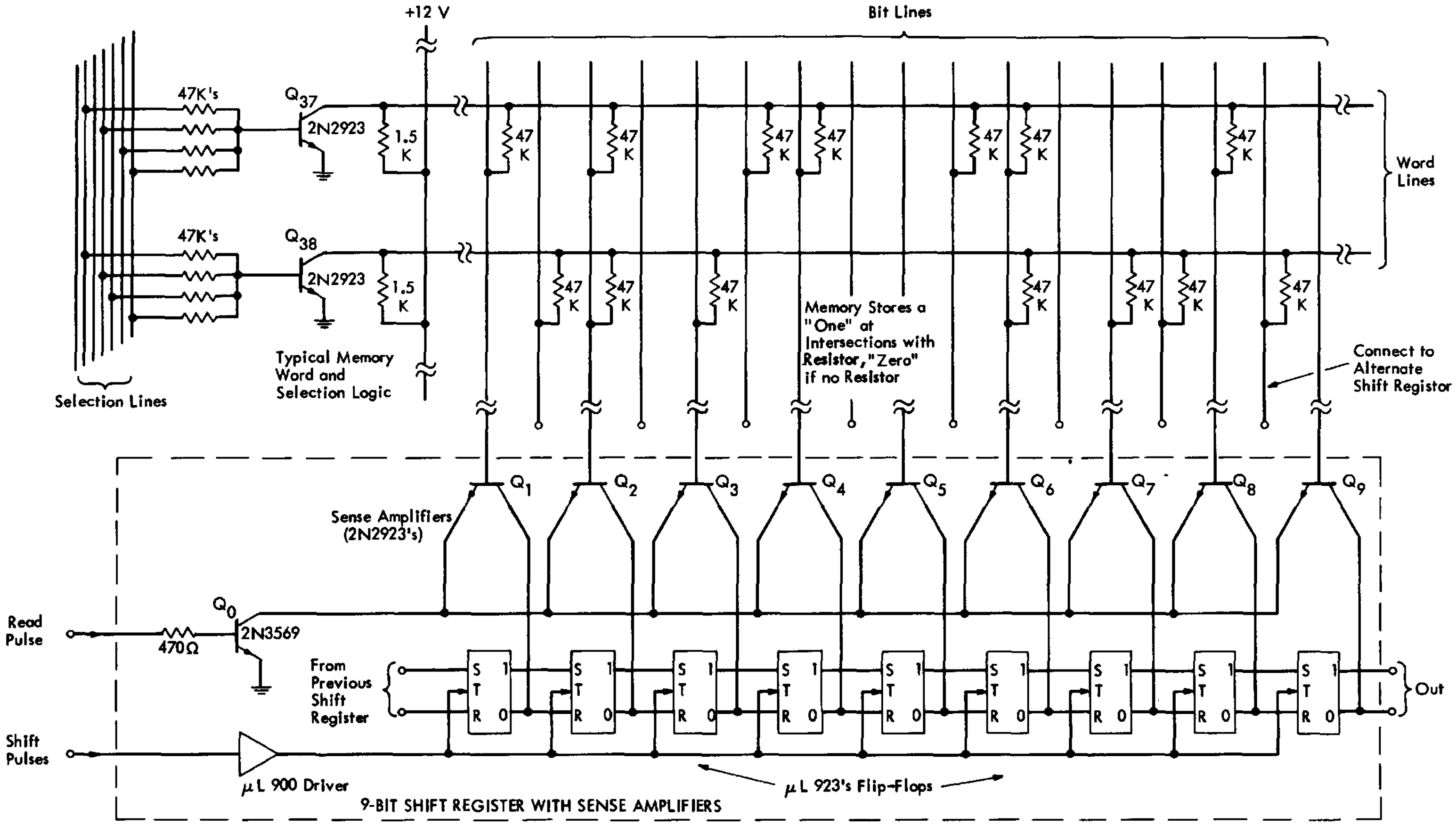

+12 V
Bit Lines
47K's
Q37
2N2923
1.5 K
47 K
Word Lines
47K's
Q38
2N2923
1.5 K
Memory Stores a "One" at Intersections with Resistor, "Zero" if no Resistor
Typical Memory Word and Selection Logic
Selection Lines
Connect to Alternate Shift Registor
Q1
Q2
Q3
Q4
Q5
Q6
Q7
Q8
Q9
Sense Amplifiers (2N2923's)
Read Pulse
470Ω
Q0
2N3569
From Previous Shift Register
S 1
T
R 0
Out
Shift Pulses
μL 900 Driver
μL 923's Flip-Flops
9-BIT SHIFT REGISTER WITH SENSE AMPLIFIERS

Fig. 4. Diagram of Resistor Matrix Memory, Sense Amplifiers, and 9-Bit Shift Register

2. Selection Logic

As shown in Fig. 4 , each word line is driven by a transistor connected as a four-input NOR gate (Q_{37} and Q_{38}, for example). If the voltage applied to any of the four 47 k base resistors by the selection lines is high, the gate transistor will be saturated and the collector output voltage will be low--about 0.1 volts. Alternately, if all four inputs are low, the transistor will be cut off and +12 volts will be applied to the associated Word Line through a 1.5 k load resistor.

Seven selection lines are used to control which word is selected. The line furthest to the right in Fig.4 is used as a "group selection line" in the 96-word memory (as will be discussed later) but is always grounded in the 8-word test memory. The remaining six lines are grouped into three pairs; each pair represents a binary bit and its complement (i.e., one line of the pair is at a high voltage while the other is low.) By properly connecting the NOR gates to these three binary pairs, the gates serve as a three-bit decoding network to select one of the eight words in the memory. In operation, only one of the selection transistors is cut off (giving high output voltage) while all the others are saturated.

3. Read-Out Mechanism

The result of the selection process is that only one word line has a high voltage on it, the exact voltage depending on how many 47 k "bit-resistors" are connected to the line. However, the voltage will be between +6 volts (for a resistor on every line) and +12 volts (no resistors).

The Bit Lines serve as the output lines of the memory and are connected to the Sense Amplifiers. When a Word Line is selected (i.e., it has a high voltage applied to it), those Bit Lines which are connected to that particular Word Line by bit-resistors have current supplied to them through the resistors. Those Bit Lines with no connecting resistors have no current flowing into them. The current in a connected bit line causes the voltage on this line to increase by an amount depending on the input impedance of the Sense Amplifier connected to the line, and on how many other bit-resistors are connected to that Bit Line. Note that since all the unselected Word Lines are virtually at ground potential, all the bit-resistors connecting a Bit Line with unselected Word Lines acts as a resistive load on the Bit Line, reducing its voltage output.

In the 8-word memory for the test generator, there could be as many as seven unselected bit-resistors connected to a Bit-Line. In this worst case--and not counting the load of the Sense Amplifier--the voltage output would be one-eighth the voltage on the selected Word Line. Therefore, the output voltage would be between 0.75 volts and 1.5 volts. Of course, for memories with more words, this worst-case voltage output would be further reduced.

Bit Lines with no bit-resistor connected to the selected Word Line have a voltage output of about 0.1 volt, the transistor saturation voltage of the unselected Word Lines. Therefore, the output Sense Amplifiers for the 8-word test memory have to distinguish between 0.1 volt for "zero" output and 0.75 volts for a "one" output.

D. SHIFT REGISTER AND SENSE AMPLIFIERS

The use of integrated circuit flip-flops made the construction of the shift registers quite simple. As can be seen from Fig. 4 all that was required was to connect the outputs of each flip-flop to the set and reset inputs of the next flip-flop. A common clock line is connected to all stages and is driven by a Type 900 Power Driver. Once the shift register is loaded with a pattern of ones and zeros, a pulse train supplied to the Shift Pulse Input causes the pattern to move to the right--each bit appearing in turn at the Output terminals. The shift register is automatically cleared (all flip-flops reset to "0") during the readout process by connecting the Reset input of the last stage of the shift register (the flip-flop furthest to the left) to the + 3.6 volt supply. This means that a series of shift pulses will cause the entire register to end up in the reset state, i.e., with all "0" outputs at +1 volt.

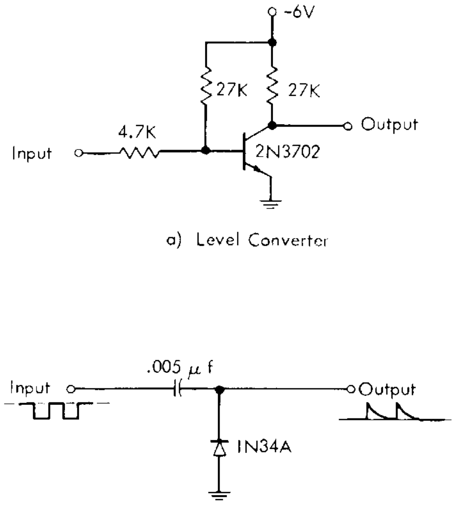
The register is loaded as follows. The Type 923 Flip-Flops are an unbuffered design, and can be set or reset by grounding the appropriate output terminals. Therefore, to load the shift register, the Sense Amplifiers (the 2N2923 transistors labeled Q_1 through Q_9 in Fig. 4) act as switches which can ground the "0" outputs of individual flip-flop in the register setting them to "1".

The emitters of all of the Sense Amplifiers are connected to the collector of the 2N3569 Read Amplifier (Q_0). Only when this transistor is saturated and acts as a virtual short circuit can the Sense Amplifiers ground the shift register outputs. Therefore, the Sense Amplifier's are only enabled when a Read Pulse is received. Whether a given Sense Amplifier will act as an open or closed switch when enabled depends on whether its base input from the matrix memory is at a "high" voltage (0.75 volt) or a low voltage (0.1 volt). A high voltage saturates the transistor setting the associated shift-register flip-flop, while a low voltage keeps it back-biased.

E. LEVEL CONVERTER CIRCUITS

Two types of level-converter circuits were needed: one to change incoming signals from levels of 0 and +1 volt to 0 and -3 volts, and one to convert -3 volt inputs to +1 volt outputs. Five of the former converters are needed to feed signals from the character generator to the vector generator; two for y-pulse sign control, one for the x-pulse output, one for the y-pulse output, and one for intensity control. Referring to Fig. 5a, it can be seen that an input of +1 volt applied to the 4.7 k base resistor will back-bias the 2N3702 transistor, causing its output voltage to go negative (diode clamps in the vector generator limit this negative excursion to -3 volts). An input of 0 volts allows the current through the 27 k base resistor to saturate the transistor, giving an output voltage of less than 0.1 volt.

The two inputs from the vector generator to the character generator are the clock pulses and the reset pulses. Since both signals are pulse trains and since their amplitudes are greater than that required by the character generator circuitry, the simple pulse inverter shown in Fig. 5b was all that was required. The 0.005 mf capacitor changes to three volts through the diode when the input goes negative. As the input goes positive, the diode is back-biased, and a + 3 volt pulse appears at the output.



a) Level Converter

b) Pulse Polarity Converter

Fig. 5. Logic Interfacing Circuits

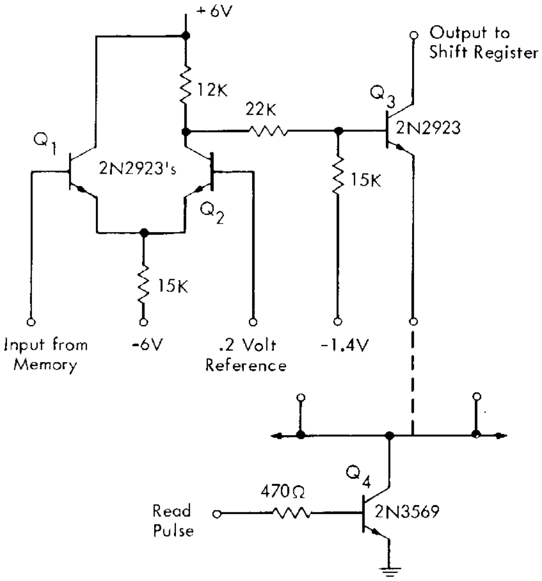


Fig. 6. High Gain Sense Amplifier

F. POWER SUPPLY DESIGN

The power supply design was quite straightforward. Three separate supplies with a common ground were needed: 36-volt 2-ampere supply to power the integrated circuits; 15-volt supply to feed the resistor matrix memory as well as drive the voltage-control circuitry associated with the 36 volt supply; and a -6 volt auxilary supply to bias the logic converter interfacing circuits. The overall circuit schematic is shown in Fig. 7.

Only the 3.6-volt supply required regulation. A six-transistor series-regulator circuit was built using the base-emitter breakdown voltage of a 2N3638 transistor as a reference voltage to control the output. Close attention to the routing of ground leads and high-current carrying leads helped guarantee a ripple-free and stable supply. Output impedance was measured to be less than 0.1 ohms.

The other supplies were simple silicon diode rectifiers with capacitor filtering. The entire power supply was constructed on a 3" x 7" x 12" aluminum chassis. A fuse, pilot lamp, power switch made up the only controls.

G. RESULTS OBTAINED WITH THE TEST GENERATOR

After the design and construction of all the needed circuits, the first tests involved use of the three-by-five dot raster and the 8-word 15-bit memory. The picture shown in Fig. 9 illustrates how numbers produced with the test generator looked when stored on the screen of the 5" Tektronix Type 564 Storage Scope.

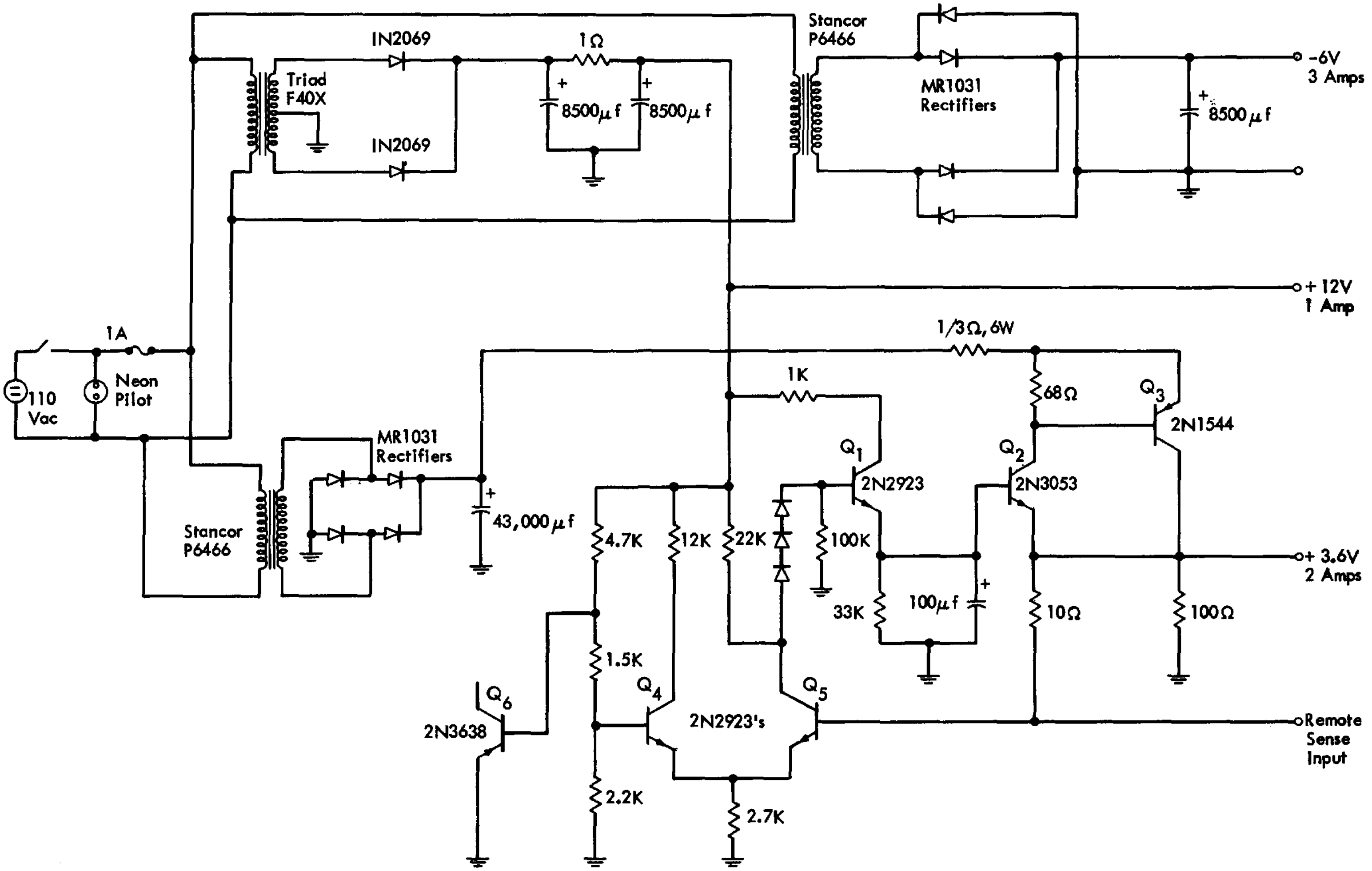


Fig. 7. Multi-Voltage Power Supply

H. DESIGN OF THE FULL-SCALE GENERATOR

The results obtained with the tést generator were so encouraging that plans for building a full-scale machine were started. Three problems were anticipated concerning the larger machine: 1) a 36-bit shift register would be harder to construct than the 15-bit one used in the test generator, 2) a larger memory matrix would make read-out signals much lower than the eight-word memory tested, and 3) the selection logic would have to be enlarged to select one of the 96-word rather than one of eight.

1. Thirty-six Bit Shift Register

The problems with building a larger shift register were of a practical nature rather than of a conceptual one. Physically, it was more desirable to build two 18-bit shift registers side-by-side than to build one long 36-bit register. In addition, the type 900 Power Driver circuits are not capable of driving more than 16 flip-flop stages. Because of these problems, it was decided to build four nine-bit shift registers, each identical to the one shown in Fig. 4. These would be used to make up two 18-bit shift registers (by connecting two 9-bit units in series) which would then be mounted in parallel. Switching logic on the outputs of the two 18-bit shift registers would be used to "inter-weave" the outputs to simulate a 36-bit shift register. The connections used are shown in Fig. 8. With the exception of these variations, the shift register and sense amplifiers operated as described earlier.

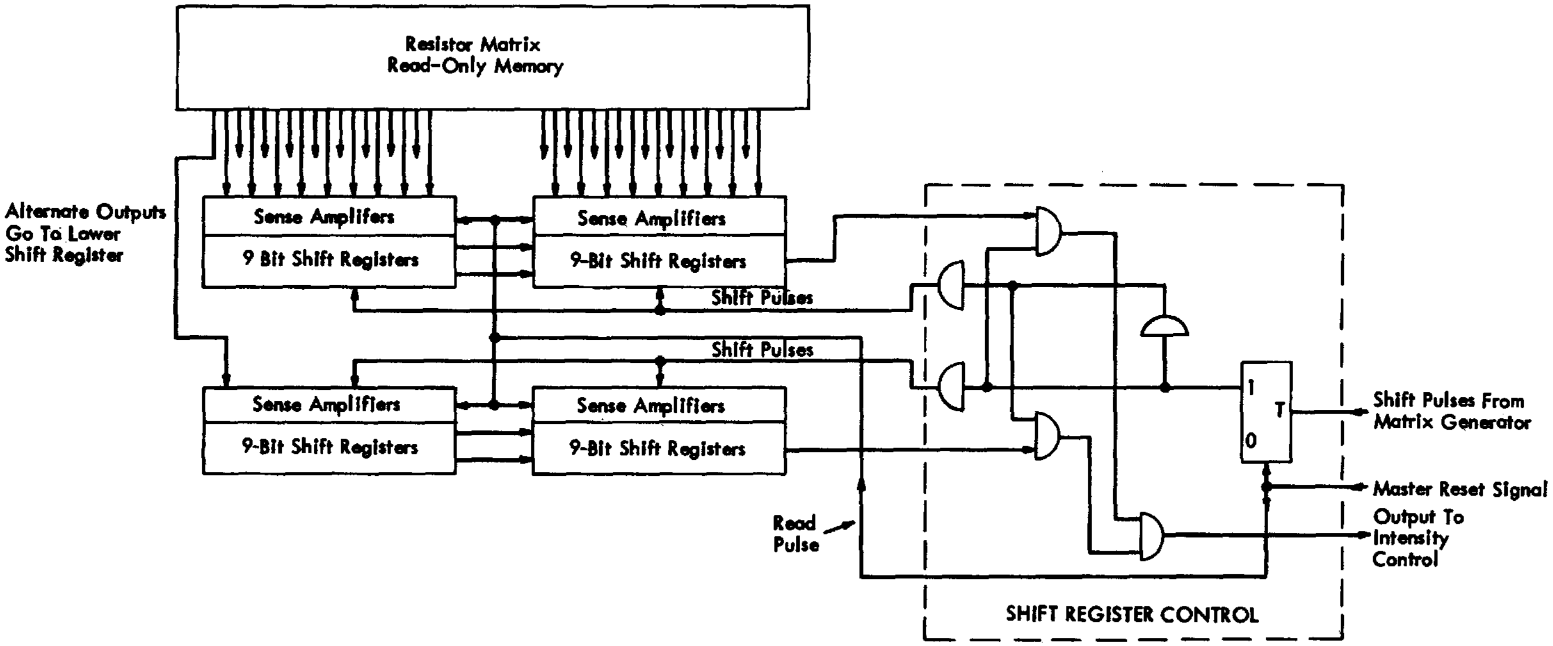


Fig. 8. 36-Bit Memory Read-out System

2. Extended Selection Logic

Since the eight-word selection logic already developed worked well, it was decided to break the character set memory into 12 groups of eight words each. Of the seven binary bits used to select one of the 96 characters, the first four bits would feed the "primary selection logic" to decide which one of the 12 groups would be selected, while the last three bits would decide which one of the eight words of that particular group would be selected. The seventh selection line shown in Fig. 5 and mentioned as part of the Resistor Matrix description would thus be used to select one of the 12 groups of eight; the remaining six would select one word in the group, as previously described.

3. High-Gain Sense Amplifier

As described earlier, the output voltage for a "one" in a resistor matrix memory goes down as the number of words goes up. For a 96-word memory, in the worst case, the output voltage is 1/96 the voltage applied to a Word Line. For 12 volts on a Word Line, the output is only 125 millivolts. This level is too low to drive the sense amplifier shown in Fig. 4. Therefore, a new amplifier shown in Fig. 6, was designed.

It has a differential input stage to maintain temperature stability and to compensate for off-set voltage of the memory caused by the saturation voltage of the selection transistors. The amplifier was designed to switch states for a 50-millivolt change in input.

There was some concern for the different values of V_{BE} of a randomly selected group of transistors; if the differences in V_{BE} were comparable to the signal levels received at the memory output, the amplifiers would be unreliable. However, it was found by testing a substantial number of transistors that the maximum variation to be expected was only 30 millivolts. Since input signal levels are to be 100 to 200 millivolts, there should thus be no trouble in obtaining positive switching. Moreover, the temperature coefficients of the transistors used were not expected to deviate more than 0.2 millivolts per degree Centegrade; therefore,no temperature instability should occur over the expected operating temperature range.

The output of the differential stage is fed through a voltage divider to match the required voltage level of the output transistor (Q_3). This transistor operates in the same manner as the simple sense amplifier described earlier. Again, the amplifier is "read" by grounding the emitter lead of Q_3 whenever Q_4 is saturated by a Read Pulse.

I. TESTS OF THE FINAL CHARACTER GENERATOR

Even though the extended selection logic scheme and improved sense amplifier were designed and tested, they were not included in the final bread-boarded circuit because of a lack of time. Also, as explained in the next chapter, it appeared that a better way to construct the memory would soon be possible. Therefore only eight characters (the single quote and the letters A through G) were included in the expanded resistor matrix memory. Nevertheless, it was felt that this machine included all the principles needed to demonstrate the feasibility of this particular approach to character generator design.

Characters that were generated and stored on a Tektronix 564 Storage Oscilloscope are shown in Fig. 11. By using the small characters shown, it appears it will be possible to store at least 600 legible characters on the 8 cm by 10 cm screen of this particular storage CRT.

The photograph in Fig. 12 shows the experimental set-up with the vector generator mounted in the relay rack at the right, the Storage Oscilloscope in the middle, and the experimental character generator with its power supply on the table at the left.

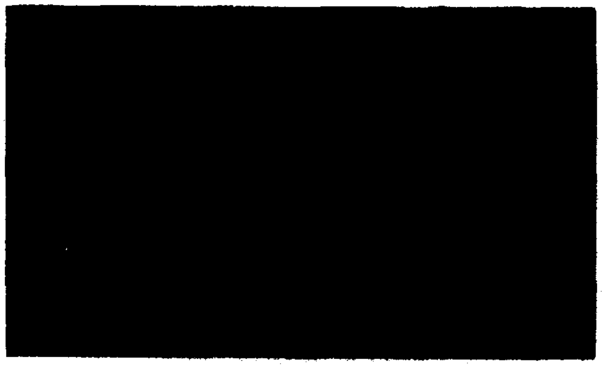
The close-up picture in Fig. 13 shows details of the bread-board construction. In the lower part of the picture is the 36-bit shift register and eight words of memory. The dot matrix generator is located in the center portion. At the top is the 15-bit memory, selection logic, and shift register.

Fig. 9 Numbers produced using 3 x 5 dot array.

Left: Normal raster

Right: Dropped raster for special lower case letters.

Fig. 10 Complete 5 x 7 dot array with beam re-locating steps.



Top: Enlarged characters, showing individual dots making up characters.

Bottom: Characters as they will normally appear in text display (3/32" high)

Fig. 11 Characters produced using 5 x 7 dot array.

Note: All pictures taken on a Tektronix 564 Storage Oscilloscope operated in the storage mode. Pictures taken 5 to 15 minutes after images were drawn.

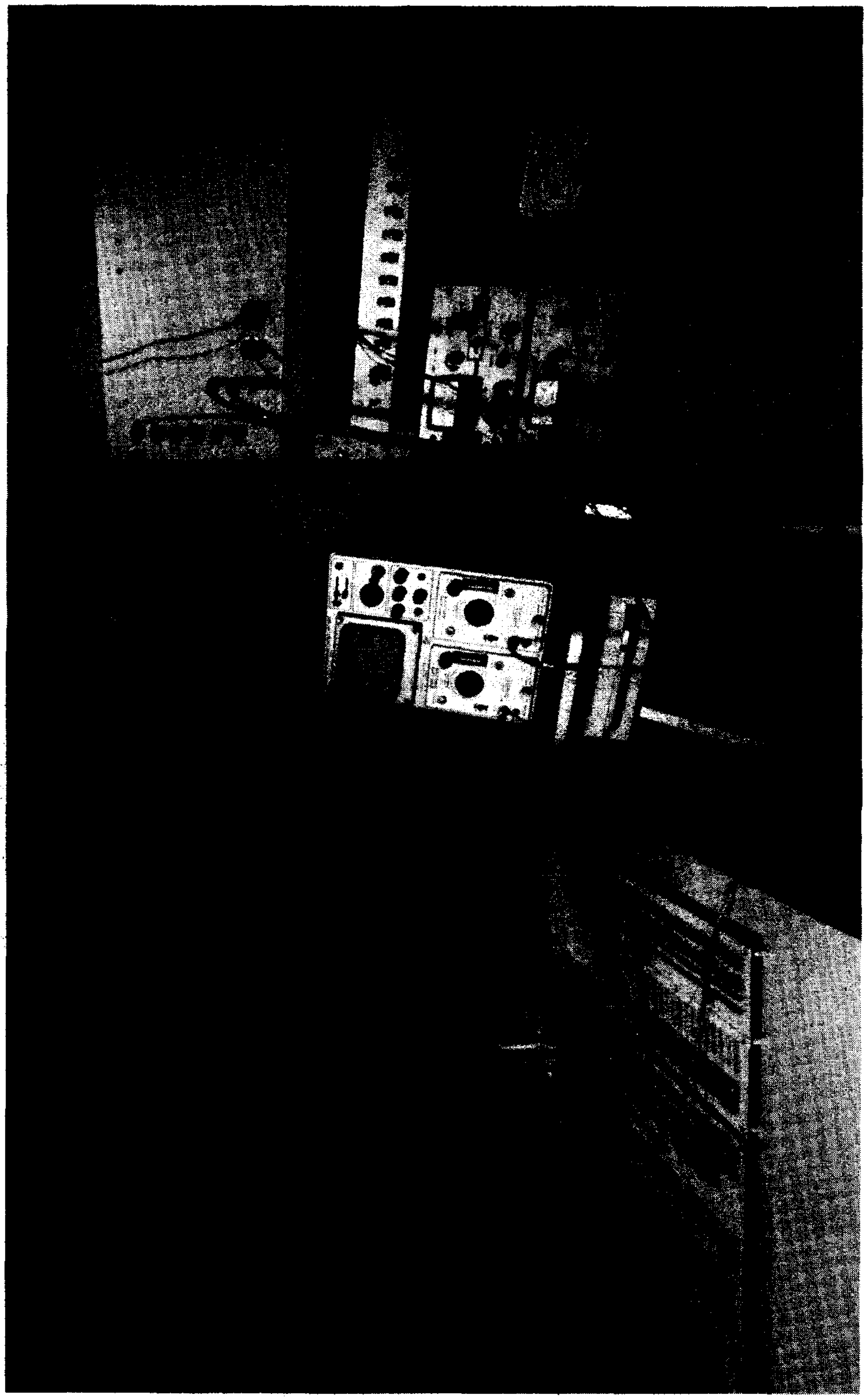


Fig. 12. Photograph of Complete Experimental Remote Terminal

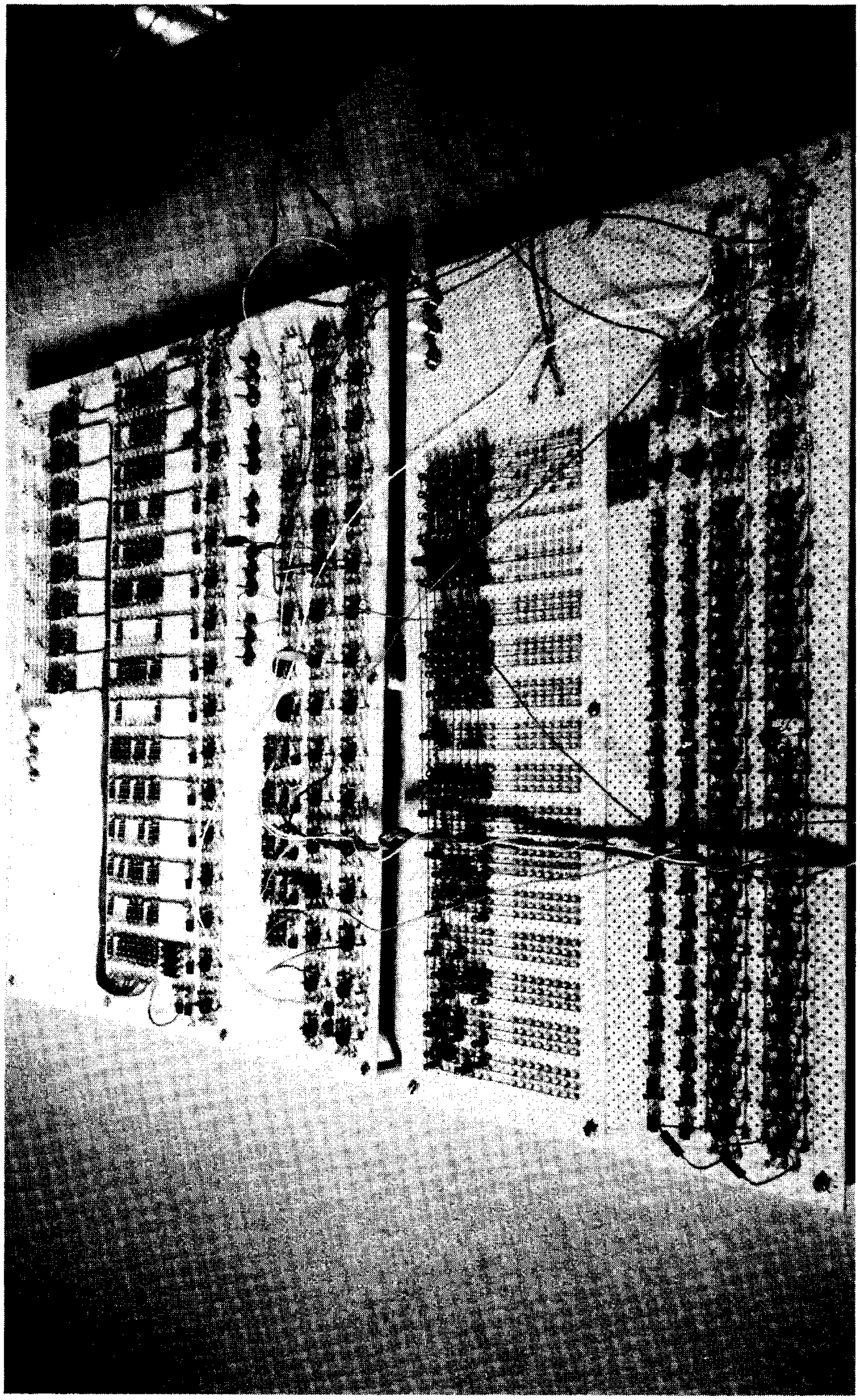


Fig. 13. Photograph of Bread-Board Character Generator

CHAPTER V

CONCLUSIONS

One of the major problems in any design project, particularly in the electronics field at the present time, is that a "best" design is highly elusive. This is because advancing technology is forever changing the relative values of using one approach versus another. The designer is forced to decide whether to base his design on present "state-of-the-art" achievements in technology or whether to second-guess what advances will occur in the near future and use this "advanced" technology as a basis for design.

Changes in technology can be of two types: 1) new products can be developed that either perform better than existing types or perform in entirely new ways or, 2) advances in manufacturing techniques can cause the prices of existing products to drop. To a limited extent, both types of change were anticipated in this project. As will be described, some of these expected advances came to fruition much more rapidly than expected, even while the unit was being designed and built.

It can be seen from the photograph in Fig. 11 that, for all intents and purposes, the experimental design produces quite legible characters. Because of its all-electronic nature, the machine should be very reliable. In addition, the operating speed is highly conservative--the machine could easily generate 10,000 characters per second. Special characters can be externally programmed. Moreover, by designing the character generator to use circuits already available in the vector generator (creating a vector-character generator), the cost of adding character generation to a remote terminal should be a minimum.

The main costs would be a certain amount of control logic, a character-set memory, word-selection logic, and sense amplifiers. Parts costs are estimated to be under $200 and it is believed construction costs should not exceed $400 for a total device cost of $600.

However, two recent developments make it appear that a much lower figure will be possible almost immediately. As was mentioned in Chapter III, it was hoped that there would be a way of automatically fabricating the resistor memory array. Towards the end of this project, it was learned that a division of United-Carr, Inc. has developed a technique of chemically depositing nickel-alloy resistors directly on printed circuit boards at a projected cost of $.005 per resistor in completed arrays. The technique also includes a way of depositing two insulated layers of copper conductors which can interconnect with the nickel alloy resistors. In addition, after the resistors and conductors have been fastened to the printed circuit board, other components such as transistors, capacitors, etc. can be mounted and dip-soldered as with an ordinary printed circuit board.

This promises to be a way in which the selection logic, resistor array, and sense amplifiers could all be constructed very cheaply--not only would there by a reduction in parts costs, but more importantly, there would be a large reduction in labor costs.

A second advancement that promises to be of possible value is the development of "silicon-on-sapphire" (SOS) diode arrays by the Autonetics Division of North American Aviation, Inc. Basically, these are arrays of diodes made by conventional integrated circuit techniques, but using a sapphire substrate to gain device independence. Arrays of 3000 to 6000 diodes are easily possible, with a packing density of over 15,000 diodes per square inch. Also, SOS construction techniques offer the possibility of simultaneous fabrication of the selection logic, the diode array, and the output sense amplifiers, all on the same sapphire substrate.

Most important, this SOS technique promises to be very low cost--somewhere between $.01 and $.03 per diode. Recall that the only reason that a resistor array was chosen in place of a diode array was lower cost. With this new development, the cost disadvantage of diodes has apparently disappeared. Not only would such a diode array with built-in selection logic reduce parts and labor costs, but the higher output levels of a diode array would eliminate the need for a three-transistor sense amplifier--a single transistor would do.

Since both the deposited-nickel-resistor and the SOS-diode arrays offer certain other advantages, which there has not yet been time to evaluate, it is still too early to say which approach is most desirable. However, it is certain that either approach will substantially lower the costs of the character-set memory. In view of this, it may be desirable to use some of the savings to increase the character dot raster to six-by-nine points (or even larger) to obtain higher quality characters. At any rate, it appears that ultimately the cost of a combined character-vector generator in some sort of "production" quantities will be no more than $100 to $300 more than a vector generator alone. Since the design goal was to reduce cost to $500 to $1000, it is concluded that this part of the low-cost remote display project has been highly successful.

BIBLIOGRAPHY

1. Fano, R.M., Project MAC: Progress to July 1964, Massachusetts Institute of Technology, Preface.

2. I.E. Sutherland, Sketchpad: A Man-Machine Graphical Communication System, M.I.T., Lincoln Laboratory Technical Report 296, January 30, 1963.

3. Stotz, R.H., and Ward, J.E., Operating Manual for the ESL Display Console, ESL Internal Memorandum 9442-M-129, Project MAC Internal Memorandum MAC-M-217, March 9, 1965.

4. Stotz, R.H., Computer Display of 3-Dimensional Figures, Electronic Systems Laboratory Report, ESL-TM-167, March, 1963.

   Stotz, R.H., "Man-Machine Console Facilities for Computer-Aided Design", AFIPS Conference Proceedings, Volume 23, 1963 Spring Joint Computer Conference, Detroit, May, 1963, pg. 323-328.

5. Ward, J.E., "Display Systems", Electronic Systems Laboratory-Annual Report, (July 1964 to June 1965) pg. 31-35.

6. Stotz, R.H., Gronemann, U., Ward, J.E., Specifications for a Dataphone-Driven Remote Display Console for Project MAC, Project MAC Memorandum MAC-M-243, June 24, 1965.

7. Instruction Manual for Tektronix Type 564 Storage Oscilloscope, Tektronix, Inc., Beaverton, Oregon.

8. Interview with Tektronix Representatives at Room 833, 545 Technology Square, on November 2, 1965.

9. ESL Display Group, Development of a Remote Digital-Dataphone-Driven Display Terminal, Un-numbered Project MAC Memorandum, May 24, 1965.

10. Machover, Carl, "Converting Data to Harman-Interpretable Form", Data Systems Design, September, 1964.

11. Ibid

12. Lewin, M.H., "A Survey of Read-Only Memories", AFIPS Conference Proceedings, Volume 27, Part I, Fall Joint Computer Conference, 1965, pg. 775.

BIBLIOGRAPHY (Continued)

13. Instruction Manual, Model 11-64 Symbol Generator, Straza Industries, Las Vegas, Nevada.

14. Stotz, R.H., Gronemann, U., Ward, J.E., op. cit. (MAC-M-243) Point II-A-(11), pg. 2.

15. American Standards Association, "Proposed Revised American Standard Code for Information Interchange", Communications of the ACM, Volume 8, Number 4, April, 1965.

16. Lewin, M.H., op. cit., pg. 4.

17. Ibid, pg. 3.