NASA TECHNICAL NOTE

NASA TN D-5474

c. 1

NASA TN D-5474

A DEVICE FOR PRODUCING BOTH HARD COPY ALPHANUMERIC LISTINGS AND DIGITAL DATA PLOTS

by John W. Oglesbee

Lewis Research Center
Cleveland, Ohio

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • OCTOBER 1969

1. Report No.
NASA TN D-5474

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle A DEVICE FOR PRODUCING BOTH HARD COPY ALPHANUMERIC LISTINGS AND DIGITAL DATA PLOTS

5. Report Date
October 1969

6. Performing Organization Code

7. Author(s)
John W. Oglesbee

8. Performing Organization Report No.
E-4863

9. Performing Organization Name and Address
Lewis Research Center
National Aeronautics and Space Administration
Cleveland, Ohio 44135

10. Work Unit No.
120-27

11. Contract or Grant No.

12. Sponsoring Agency Name and Address
National Aeronautics and Space Administration
Washington, D.C. 20546

13. Type of Report and Period Covered
Technical Note

14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

A high-speed printer was modified to perform the additional function of x-y plotting of digital data. The basic printing mechanism and its modification to a plotting device are discussed. A small digital computer was selected as a control device, and the special software developed for this application is described. In the plotting mode, the device is capable of producing a typical (7-in.-square) plot in about 25 seconds. The plot is constructed from a matrix of dots, with a resolution of 500 dots across the width of the plot. When not plotting, the device can be used as a 300-character-per-second printer.

17. Key Words (Suggested by Author(s))
High-speed digital plotter-printer

18. Distribution Statement
Unclassified - unlimited

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price*
Unclassified	Unclassified	25	$3.00

*For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151

A DEVICE FOR PRODUCING BOTH HARD COPY ALPHANUMERIC LISTINGS AND DIGITAL DATA PLOTS

by John W. Oglesbee

Lewis Research Center

SUMMARY

A high-speed printer was modified to perform the additional function of x-y plotting of digital data. The basic printing mechanism and its modification to a plotting device are discussed. A small digital computer was selected as a control device, and the special software developed for this application is described.

In the plotting mode, the device is capable of producing a typical (7-in. -square) plot in about 25 seconds. The plot is constructed from a matrix of dots, with a resolution of 500 dots across the width of the plot. The number of points to be plotted does not affect the plotting time, and several variables may be plotted simultaneously using different plotting symbols for each variable. The plotter generates any of a number of grid patterns, complete with titles, axis labels, and numerical scales. When not plotting, the device can be used as a 300-character-per-second printer.

INTRODUCTION

An IBM 360/67 time-shared computer system has been under implementation at the Lewis Research Center since 1967. One primary purpose of this system is to provide remote computer facilities in several test area control rooms. A need was anticipated for a high-speed graphics printing device to provide for plotting of computed data at these remote terminals. A commercially available high-speed printer was purchased and modified to satisfy this requirement. The modification has resulted in a low-cost device which is useful in many applications where plotting is required. The device has a distinct advantage over other plotters in that its function as a high-speed alphanumeric printer has also been retained.

The unmodified printer was designed to write on electrosensitive paper using a moving conductive stylus. Characters were made uɒ from elements of a 5 by 7 dot matrix which was printed by applying current pulses to the scanning stylus. A special translator was required for use with the printer to translate ASCII (American Standard Code for Information Interchange) input information into the pulse sequence required by the printer.

Modification of the printer to a plotting device required development in three areas. The paper advance mechanism of the printer was modified to allow printing in the blank areas forming the vertical spaces between alphanumeric lines. The synchronizing system used with the moving stylus was improved with a resulting improvement in the resolution of the printing device. A small digital computer was used to replace the original character translator allowing the generation of additional printing elements required for plotting.

This report describes the printer and the modifications that enable it to print graphic information. The printer must be supplied by a specialized sequence of electrical pulses to perform the plotting operation. One method, described herein, utilizes a small digital computer. The computer software developed for this application is described as well as an interface unit required between the computer and the printer.

The developments described herein are representative of a system meeting the minimum requirements of a useful printing and plotting device. In developing this system, the capability of the printer as an element for producing rapid hard-copy plots has been demonstrated. Example plots showing the quality and type of output available from the device have been made (see fig. 9).

With the capabilities of the device thus demonstrated, a wide area of applications remains to be investigated. In many cases, the device will find applications in existing computer oriented systems without additional computer hardware.

BASIC PRINTING MECHANISM

Printing Process

Printing is done on electrosensitive paper by several continuously scanning print styluses. Dark areas (dots) are produced on the paper by applying current pulses to these styluses at specified times during the scanning process. Current pulses flow from a commutator bar to each print stylus, through the paper, and are returned to ground by a metal wiper on the back side of the paper. The timing and control of these current pulses are handled by the digital computer and its interface unit.

Mechanical Features

The printer is shown in figure 1, and the major mechanical features of the printing mechanism, including the detail of a print head, are illustrated in figure 2. Seven print styluses are mounted on each print head. Since each alphanumeric character is constructed of a matrix of dots 7 units high, an entire line of alphanumeric characters can be printed with each scan of a print head. Four print heads are mounted on a continuously moving belt. The distance between the print heads on the belt is such that only one head is in contact with the paper at any given time. As one print head leaves the right edge of the paper, the next head on the belt moves into position at the left edge of the paper. A continuous horizontal scanning process is thus achieved without retrace delay.

Paper Advance

Vertical spacing of lines on the paper is achieved by continuously advancing the paper during the scanning process. Two different paper speeds are used to provide proper line spacing for either the alphanumeric printing mode or the plotting mode.

A faster paper speed is used in the alphanumeric printing mode that allows the paper to advance a distance equivalent to the character height plus a vertical (between line) space during the time interval of each scan. This speed results in proper vertical spacing of alphanumeric lines, without using extra time to scan the space areas between the lines.

In the plotting mode, the entire area of the paper must be scanned. A slower paper speed must be used so that the paper advances a distance equivalent to the height of the plotting head during the time interval of each scan. The unscanned space between lines is therefore eliminated, and plotting may be done anywhere on the paper. Vertical spacing of alphanumeric lines printed in this mode is achieved by printing characters only during every other scan. The alphanumeric printing speed is thus reduced by a factor of 2 in the plotting mode, but this is necessary to allow complete scanning of the paper.

The effect of the vertical paper motion during each scan can be minimized if the print head belt assembly is skewed slightly with respect to the horizontal. Ideally, the skew angle should be set so that print head motion has a vertical component equal to the speed of the advancing paper. However, exact correction at each of two different paper speeds is not possible, and a compromise must be made. This compromise results in a slight deviation of the printed lines from true horizontal, but this error is not noticeable to the eye.

INTERFACE UNIT

The interface unit contains a master clock which provides the basic synchronization for both the computer and the printer. The unit also provides for control and information transfer between the two units. A block diagram of the interface unit is shown in figure 3. The 525-word sections of the computer memory are used as storage buffers and are shown on the block diagram.

Information is transferred from the computer to the interface unit by using a computer subroutine. The timing of the transfer is controlled by the master clock, with each clock pulse causing a program interrupt and execution of the subroutine. One word of information is transferred with each execution of the subroutine. The diagram of figure 3 includes blocks indicating the program interrupt function and the transfer subroutine.

An output gate controls the information transfer between the interface unit and the printer. Timing for this gate is again under direct control of the master clock.

The sync decoder provides a synchronizing pulse each time a print head reaches the left paper edge. This pulse starts the master clock and initiates the printing process for each line. The problem of synchronization is discussed in more detail later in this report.

The control decoder recognizes special computer commands which are used to start or stop the printer in either of its modes, or to turn off the master clock.

Definition of Image

The term "dot location" is used in this report to refer to a point of the printed output where a dot may appear. Any particular dot location may or may not contain a dot. Each printed line consists of 3675 dot locations arranged in a matrix 7 locations high and 525 locations wide. A one-to-one correspondence is made between the dot locations of each printed line and the bits of storage in the computer buffers. Each column of the printed matrix corresponds to a word of storage in the computer buffer, with the seven dot locations of each column corresponding to the seven bits of each buffer word. A one bit in a buffer indicates a dot is to be printed in the corresponding dot location, while a zero indicates that no dot is to be printed.

The pattern of dots which makes up each printed line is thus represented in the computer memory by a pattern of binary bits. The term "image" will be used below to indicate any such pattern of bits in memory. Image may refer to a single character or symbol, or to an entire line of characters.

Computer Buffers

The two buffers provided in the computer memory are used alternately. An image of the line being printed is held in one of the buffers until the entire line has been printed. At the same time, the image of the next line to be printed is generated in the opposite buffer. At the start of a new line of printing, the buffer functions are reversed so that the image just generated is now held during printing of the new line. The two buffers allow the computer operations of generating an image to be time independent of the operations of printing an image.

Printing a Line Image

Printing of each new line is initiated when a print head reaches the left edge of the paper. A synchronizing pulse is generated at this time which starts the master clock. The clock is then free-running for the remainder of the line and produces one pulse corresponding to each column of the line matrix.

Each clock pulse generates an interrupt and forces execution of the transfer subroutine. A new word is fetched from the appropriate buffer and transferred to the interface unit where it is held temporarily. When the print head reaches the location where the next column is to be printed, the word is gated into the printer and the column is printed. This process is repeated with each new clock pulse until all 525 columns have been printed. A column counter in the computer detects the last column of the line and causes a "stop clock" command to be executed. The clock stops, and the printing of the line is completed.

This sequence of events is illustrated by the timing diagram of figure 4. The line represented is a letter "T" followed by all spaces. The contents of the buffer for this line (the line image) is shown in table I.

Synchronization Difficulties

A synchronizing pulse from the printer is necessary to determine the exact time at which the pulse sequence for each line is to begin. The printer was originally equipped with a cam-operated switch to provide this synchronization. The cam was geared to the main drive sprocket of the print head belt and adjusted so that the switch opened whenever a print head reached the left most column position. While this method was adequate for alphanumeric printing, the errors in timing were not tolerable for plotting applications. Figure 5 shows an example of an early plot made using the cam-operated switch for synchronization. The uneveness in the vertical lines of the plot is a result of inaccurate synchronization.

Investigation revealed that the synchronization errors arose from several sources: backlash in the gears driving the cam, eccentricities in the gears, backlash between the drive sprocket and belt holes, and the print heads shifting slightly in their mounting positions with respect to the belt. These synchronization errors were eliminated by changing to a system of electrically sensing the first contact between the print head and the paper surface.

Synchronization by Electrical Sensing

A short time interval exists at the end of each line in which none of the print heads is in contact with the paper. This interval occurs after a print head has moved beyond the right edge of the paper, but before the next head has reached the left paper edge. The original synchronization switch was readjusted to open near the center of this time interval. The timing of the switch was sufficiently accurate to ensure contact opening within this time interval.

The opening of the synchronization switch was used to turn on all seven print head styluses simultaneously. At the instant of the switch opening, none of the styluses is in contact with the paper and no current flows in the print head circuits. However, as soon as a print head contacts the left paper edge, current flows through the head styluses and conductive paper to ground. A current sensing circuit was added to the printer power supply to detect the start of this current flow.

The current-sensing circuit generates a pulse which starts the master clock. The same pulse is used to turn off the current to the print head styluses. Since the styluses are turned off as soon as current is detected, nothing significant is printed in the process. Synchronization is established by this method, however, and the remainder of the line then is printed, as described earlier. The improvement in resolution resulting from this synchronizing method is evident from a comparison of figures 5 and 6.

Other Timing Errors

A close examination of figure 6 will reveal that while the synchronization at the left edge of the paper is adequate, there is still a slight uneveness in some of the vertical lines. This uneveness can occur only as a result of variations in either the belt velocity or clock pulse rate. The belt velocity was checked by using a strobe light and was found to vary periodically, while the clock was found to be stable. Further examination determined that the belt speed variations were caused by eccentricities in the gears between the drive motor and the drive sprocket.

The belt speed variations can probably be reduced by using higher precision gears in the belt drive mechanism, and the slight uneveness of vertical lines on the plot can, therefore, also be reduced. This has not, as yet, been implemented.

SOFTWARE DESIGN

Previous sections have indicated how a line image may be printed once it is available in the computer memory. The remaining problem is to make the line images available as they are demanded by the printer.

One solution to this problem is to provide sufficient memory space so that all line images for an entire plot may be stored in memory before the printer is started. Computer operations during the printing process would consist simply of reading the stored images out of the memory. The technique would be particularly adaptable to a computer with a large available memory space and a direct memory access feature.

If this technique is used, a typical plot might consist of a 500 by 500 unit matrix of dot locations, which would require storage space for a 250 000-bit image (approximately 16 000 16-bit words). The area of core involved would first be cleared to zero. The desired grid pattern would then be stored in the core, along with any numerical scales, axis labels, or titles. Data could be converted to an appropriate "data symbol image" as it was received, with the image then stored in an appropriate location of the core. (The data symbol images are stored nondestructively, i. e., without destroying the grid image or other information already located in the memory.) When all data for the plot were received, the images for the entire plot would be available directly from the memory for printing.

The advantage of this technique is that printing takes place without the necessity of any correlated computations. The major disadvantage is that the size of the plot must be limited by the memory space available for storing images. The computer available for development work had a limited memory (4000 words), and hence this technique was not explored further. The reader should be aware, however, that this technique might prove quite useful in some computer facilities.

A second technique was used with the available computer which requires much less memory capacity but has the disadvantage that it involves a number of computations correlated to the printing process. The technique depends on the fact that most plots are constructed of major areas of blank space. The memory capacity necessary to specify a plot can be minimized by ignoring the blank areas and identifying only those areas which are to be printed. Further reduction in the storage capacity is possible by identifying elements of the printed areas which are repeated in the same form in different locations of the plot. Such elements can be defined once and then used in several different locations.

For example, the same symbol may be used to plot many different data points. All such symbols can be specified by first defining the dot pattern of the symbol, and then listing the different locations on the plot in which this dot pattern is to appear. As another example, a particular grid might consist of a series of horizontal line segments. All such segments can be specified by defining the dot pattern of one line and then listing the locations in which that type of line is to appear.

The construction of a plot can thus be specified by defining the dot patterns of particular elements of the plot, and then indicating the locations on the plot in which these elements are to appear. The computer operates on this information to generate the complete image required for any particular line of the plot.

GENERATING LINE IMAGES

Synchronization to Printer

The operations of generating each new line image take place during the printing of the preceding image. When a new line image has been completely formed, the program waits for the printer to finish the line it is printing. Computations for the next line image then begin simultaneously with the printing of the line just formed. Synchronization between the program and the printer is required only at the beginning of each line, and the only restriction on the timing of computations is that the program be capable of generating each image in less time that is required to print a line.

Summary of Computations

The generation of each line image proceeds through the steps of clearing the image buffer, generating a grid pattern for the line, generating all plotting symbols appearing on the line, and generating any alphanumeric characters appearing in the line. The grid patterns, data symbols, and alphanumeric characters are generated by independent parts of the computer program. The results of each part are superimposed in the common buffer before the printing of the image begins.

The flow diagram of figure 7 illustrates the overall computing process. The major functions of the program are shown on the left of the diagram, while the transfer subroutine is detailed on the right.

The only connection between the main program and the transfer subroutine is through the common image buffers, and through the "end of line" exit from the subroutine. The end of line exit path occurs following printing of the last column of each line and serves to synchronize the program to the printer at the start of each new line.

GENERATING GRID PATTERNS

The commonly used grid patterns are, by their general nature, symmetrical in structure. This makes the grid patterns adaptable to generation by an iterative computer program from a minimum of defining information. For use on a small computer, defining information must be kept compact while at the same time the information must be flexible in allowing a useful variety of grids to be constructed. The method described subsequently strikes a practical compromise between these factors.

A program was developed which generates the grid from top to bottom in sections of seven rows; the sequence required by the printer. It is useful for generating any grid consisting of horizontal or vertical lines or line segments. Different grids may be generated by changing the input information, and the information is sufficiently compact that it is practical to keep a library defining several grids resident in core.

Specifying Grid Structure

For the purpose of specifying a grid structure, the area scanned by the print heads (analogous to "raster area" of television) is divided into its individual columns. The grid is then defined by specifying a particular dot pattern for each column of this plotting area. Each such vertical column extends for the entire length of the plot.

It is often desirable that the grid not extend across the entire area scanned by the print heads. However, since "blank columns" can be processed in the same manner as columns containing other grid information, it is convenient to think of the grid as extending across this entire area. Blank areas (margins) at the left or right side of the plot are then defined by the inclusion of blank columns in appropriate locations of the grid. These blank columns are particularly useful at the left side of the plot to allow blank space for printing alphanumeric information defining axis values.

Although there are many (525) columns associated with a particular grid, there are generally only a few distinct types of columns in any grid. The advantage of specifying a grid in terms of its columns becomes apparent; columns of the same type (i. e., columns with the same dot pattern), while they may appear many times in a particular grid, need only be defined once.

The grid of figure 8, as an example, contains only three distinct column types: type 1, a blank column; type 2, a solid column of all dots; and type 3, a column where dots appear in every 50th location. The grid of figure 8 is easily defined in terms of these column types:

(1) The first 24 columns of the grid are type 1 columns.

(2) The 25th column, and every 50th column thereafter, is a type 2 column.

(3) All other columns of the grid are type 3 columns.

This example illustrates the general method used to specify any grid. The distinguishable columns of the grid are first identified, and each distinct column type is assigned a column type number. The definition of the grid is completed by specifying a column type number for each column of the plot. (It is assumed that, if the column type number is known, the column dot pattern can be constructed for any desired grid line image.)

The problems of specifying a type number for each column of the grid and specifying the dot pattern for each column type are discussed next.

Specifying Column Type Numbers

Column type numbers could be specified by a list of words, with each word of the list containing the column type number appropriate to a particular column of the plot. However, large sections of such a list would contain the same words, since the same column type is often used in many consecutive locations. The list was simplified by defining a word format in which the last 6 bits of each word specify a column type number and the first 9 bits specify the number of times that column type is to be repeated in consecutive locations of the grid. A list of words of this type is shown in table II. This list defines the column type numbers for the grid of figure 8.

Further simplification of the list in table II is possible by observing that words 2 and 3 (001-02 and 049-03) repeat themselves for the duration of the list. Since repetition such as this is characteristic of most grids, a special word containing zeros was used to indicate a section of the list to be repeated. A "zero" word signifies to the program that the next word is to be obtained from the second location of the word list. (The first location of the list is reserved to specify a group of blank columns at the left side of the plot.) The list of table II can be reduced to the four words shown in table III using this technique. The word list of table III is sufficient to define the column type numbers for all column locations of the grid of figure 8.

Constructing Line Image for Grid

The interpretation applied by the program to each word of table III is shown in the last column. The image of the grid for each line is formed by scanning this list of words and identifying a column type number for each column of the grid. As each column type is determined, an appropriate image for that column type, corresponding to the line being processed, is transferred to the buffer. Construction of the grid image is complete when all locations of the buffer have been filled with a particular column image.

A separate memory location is assigned for each different column type which contains a particular 7-bit section of the image of that column type. These memory locations are updated at the beginning of each new line, so that they always contain sections of the image of each column corresponding to the current line. In constructing the grid, as each column type is identified, an appropriate image for that column type is therefore immediately available from one of these memory locations. The information used to update these memory locations is obtained from information specifying the dot patterns for each different column type.

Specifying Column Dot Patterns

The dot pattern of each different column type is defined by a list of special words. Reference is made to these lists at the beginning of each new line, and from the information of each lish the next 7 bits of that column image are derived.

Since most grids contain a large percentage of blank space, many consecutive locations of most column types will be blank. Such groups of blank dot locations are represented by single word, with the word indicating the number of locations involved. In a similar manner, groups of consecutive dots can also be represented by a single word. The two types of groups are distinguished by the sign bit of the word; a "0" sign bit indicates a group of dots, while a "1" sign bit indicates a group of blank dot locations. A single dot or blank dot location is thought of as a "group" of one and is then defined by the same word format.

Words of this format arranged in a list define the dot pattern of each column type. Words of each list are arranged in the order in which different groups of dots or blanks occur in the column dot pattern. As done previously, it is useful to define a "zero" word to indicate a portion of the word list to be repeated. Table IV shows an example of a word list defining a column type where a dot appears in every 50th location.

Program Requirements for Generating Grids

The commands for the program to generate the grid occupied about 100 words of memory space. In addition, the program required storage space for the word lists defining the grid structure. Of the grid types shown in this report, the grid shown in figure 9 required the greatest storage space. This grid required 36 words for definition.

The number of computer cycles required to generate a complete grid image for one line depends on the exact pattern of the image. However, 25 000 computer cycles per line image is typical.

GENERATING PLOTTING SYMBOLS

Each data point to be plotted is represented by one of several possible plotting symbols. Figure 10 shows some examples of the types of symbols which may be used. Each symbol is constructed of a series of dots lying in a matrix of dot locations 5 units square.

Defining Symbol Patterns

The dot pattern of each symbol is represented by a five-word image of the symbol permanently stored in the computer memory. Each of the five words of the image corresponds to one column of the printed symbol. The five significant bits of each word correspond to the five dot locations of each column of the symbol.

Each different symbol is identified by a particular symbol type number. A symbol type number must be specified for each point to be plotted, and the program uses this number in referencing to the location in memory of the appropriate symbol image. If no number is specified with a particular data point, a type zero symbol is assumed and the point is plotted using a type zero symbol.

Specifying Location of Plotted Symbol

The location of each symbol to be plotted must be specified by a horizontal coordinate and a vertical coordinate. Vertical columns of the plotting area are numbered consecutively from left to right. Horizontal rows of the plotting area are numbered consecutively from top to bottom. The horizontal coordinate of each symbol to be plotted is defined by stating the column number of the plot in which the center of the symbol is to be located. The vertical coordinate of the symbol is defined by stating the row number of the plot in which the center of the symbol is to be located.

It is assumed that all input data defining symbol coordinates are received in the form of a row number and a column number. Horizontal or vertical translations to position plotted symbols with respect to the grid, and scaling of coordinate values so symbols will be defined within the area of the plot, are assumed to be done external to the plotting program. It is also assumed that a symbol type number is associated with each pair of coordinates to specify the type of symbol to be plotted.

The information specifying each symbol is stored in two 16-bit words as follows:

S TTTTT CCCCC CCCCC

S RRRRR RRRRR RRRRR

where

S sign bit (unused)

T type number (5 bits)

C column number (10 bits)

R row number (15 bits)

Each symbol to be plotted therefore required two words of core storage for its definition.

Plotting of Symbols

During the generation of each line image, the program examines the words defining symbols to be placed on the plot. The row numbers are used to determine if the vertical position of any symbol is such that a part of the symbol will appear in the line image currently being generated. If this is the case, the symbol type number is examined, and the location of the image defining the symbol dot pattern is determined. The words defining the symbol image are then fetched one by one from memory and transferred into an appropriate location of the buffer for printing. Five words are transferred for each symbol constructed until all relevent symbols have been processed.

Recall that each word of the computer buffer represents one column of the printed line. Horizontal position in a printed line is therefore related to word location in the buffer. The column number of the symbol (horizontal coordinate) is used to define the location of the five words of the buffer into which the image of each symbol is transferred. The horizontal position in which each symbol is printed is defined in this way.

The images defining each symbol dot pattern are stored in memory with the center of the symbol associated with the center position of each line. Symbol image words transferred to the buffer without modification will therefore cause the symbol pattern to be printed in the center of the line. This "standard position" for the center of a symbol may be modified either up or down by shifting the bits of each word of the symbol image before it is transferred to the output buffer. The center of any symbol may therefore be relocated in any row of the line, and vertical resolution of one row is attained.

The number of shifts required can be determined by dividing the row number that defines the symbol location by 7, the number of rows in each line. The remainder after this division specifies in which row of the particular line being constructed the center of the symbol is to be located. From this information an appropriate shift command is derived.

Finally, it should be noted that some symbols may overlap between lines when they are printed. Such symbols are processed twice; once during the construction of each of the two line images. The upper part of the symbol is first constructed in the appropriate line image. The symbol is later processed a second time to locate the lower part of the symbol in the succeeding line image.

The symbol images are thus fetched according to type number, positioned horizontally by addressing techniques, positioned vertically by shifting the bits of the image words, and then stored in the computer buffer prior to printing. The method has proved quite satisfactory for plotting individual data symbols.

Program Requirements for Plotting Symbols

The length of the program required to scan the symbol definitions and transfer appropriate symbol images to the buffer is about 100 words. Each different symbol type used required five words for image definition. The 10 symbols of figure 10, for example, require 50 words of core storage. In addition, each symbol to be plotted requires two memory words to define its location. The program to generate the plotting symbols thus requires 150 memory words plus two additional memory words for each symbol to be plotted.

The maximum number of symbols which may be plotted on any one line is limited by the computer time available to process the symbol images. The computer which was used in developing the plotter had a basic cycle time of 1.0 microsecond. Since each line takes about 300 milliseconds to print, there is time to execute about 300 000 computer cycles during the printing of one line.

Several necessary operations must be performed on each line in addition to transferring symbol images to the buffer. These include clearing the buffer, generating the grid, generating alphanumeric characters, interrupting to output column images to the printer, and other miscellaneous chores. It is estimated that about 50 000 computer cycles per line are required for these operations. With 300 000 cycles available per line, there are about 250 000 cycles per line available for processing symbol images. Processing one symbol image consumes about 250 computer cycles; hence, the maximum plotting density, as limited by the computer processing time, would be 1000 symbols per line.

The preceding density, as dictated by a 1.0-microsecond cycle time, is much higher than would be practically required. However, the relation between computer speed and plotting density has been explained, and this factor becomes more critical with slower computers. For example, a 5.0-microsecond cycle time would allow only 60 000 computer cycles per line, leaving only 10 000 cycles per line for processing symbols (this assumes also that the estimate of 50 000 cycles of "other operations" is precise), and resulting in a plotting density of only 40 symbols per line. This might by unduly restrictive in some applications.

Generating Alphanumeric Characters

Although a translator is available from the printer manufacturer for generating alphanumeric characters, it was more convenient to provide this function through the computer used for plotting. A program was therefore developed in which alphanumeric characters are generated from a list of words in the ASCII format. The program reads down the list of words and generates appropriate characters until the first carriage-return - line-feed word is reached. Generation of further characters is then inhibited until the construction of the next line image is begun. With the start of the next line, the program resumes reading from the point in the word list where it left off, and continues to generate new characters until another line-feed - carriage-return is reached.

Printing of alphanumeric characters from ASCII input information is thus done in much the same manner as with a conventional typewriter. The input information must be available, either in memory or from an external source, as demanded by the printer. Once the printing process for a line begins, the printer cannot be stopped to wait for additional input information. The printer prints text at a rate of 300 characters per second. An example of alphanumeric printing is shown in figure 11.

The printing and plotting programs may be operated simultaneously if desired. Notes, titles, and scales may be printed on plots using the alphanumeric program in this manner. The positioning of the characters on the plot is done in the same manner as used on a conventional typewriter, that is, by the use of carriage return, line feed, and space characters.

It should be noted that, in the plotting mode, the paper speed is slowed so that the vertical spaces between adjacent alphanumeric lines is eliminated. Lines of alphanumeric characters used with a plot must therefore be terminated by a double line feed to provide a vertical space between lines. Each line feed "advances" the paper a distance equal to the character height.

The characters in either the printing or plotting mode are generated by the same program, in a manner similar to that in which the plotting symbols are generated. Each alphanumeric character is made up of a dot pattern constructed on a matrix of dot locations 7 units high and 5 units wide. A five-word image is permanently stored in memory to define the dot pattern of each alphanumeric character. Each word of the image represents one column of the character matrix. The 7 bits of each word correspond to the 7 dot locations of each column of the character matrix.

As each new character is read from the input word list, the ASCII format is decoded and used to reference the memory locations in which the image of the character is stored. The five words of the image referenced are transferred one by one into consecutive locations of the computer buffer. (No shifting of the words is required as was done with the plotting symbols, because the characters are always centered in the line printed.) The process continues until the buffer is full, or until a line-feed - carriage-return character is recognized. Processing of the line image is then terminated, and no further input information is read until construction of the next line image begins.

The program for reading the ASCII input formation and transferring appropriate images to the buffer occupies about 90 memory words. The images for the 64 ASCII characters require an additional 320 words for storage. Each character to be printed must be specified in the input information which occupies 1/2 word (8 bits) per character.

Approximately 10 000 computer cycles are required to construct a complete line of alphanumeric characters. This represents the maximum number of operations required in any one line for generating alphanumeric characters.

CONCLUSIONS

This report has described the modifications made to a commercially available printer that allow it to plot digital data symbols over a reference grid. The modifications have resulted in an economical and mechanically reliable device capable of producing a 7-inch-square hard-copy plot in about 25 seconds. To perform these plotting functions, it was necessary to interface the device to a small digital computer. The basic program developed for this application has served to illustrate the general requirements of a computer system for driving the device. The completed program occupied about 1400 16-bit words, and the maximum plotting density was related to the basic computer speed. A 1.0-microsecond cycle time was more than adequate, although caution is indicated in using a much slower computer. The system described has served to demonstrate the quality and types of plots obtainable from the printer. The usefulness of the printer as a plotting device has thus been established.

Lewis Research Center,
National Aeronautics and Space Administration,
Cleveland, Ohio, August 20, 1969,
120-27.

TABLE I. - CONTENTS OF BUFFER

FOR LETTER "T" FOLLOWED

BY SPACES

Buffer word number	Contents of word	
1	000000000	1000000
2	000000000	1000000
3	000000000	1111111
4	000000000	1000000
5	000000000	1000000
6	000000000	0000000
7	000000000	0000000
.	.	.
.	.	.
.	.	.
525	000000000	0000000

TABLE II. - WORD LIST DEFINING COLUMN TYPES FOR

GRID OF FIGURE 8

Word number	Contents of word		Meaning assigned to each word by program
	Bits 2 to 10 (number[a] of consecutive columns)	Bits 11 to 16 (column type number[a])	
1	0 2 4	0 1	Generate 24 type 1 columns
2	0 0 1	0 2	Generate 1 type 2 column
3	0 4 9	0 3	Generate 49 type 3 columns
4	0 0 1	0 2	Generate 1 type 2 column
5	0 4 9	0 3	Generate 49 type 3 columns
6	0 0 1	0 2	Generate 1 type 2 column
7	0 4 9	0 3	Generate 49 type 3 columns
8	0 0 1	0 2	Generate 1 type 2 column
9	0 4 9	0 3	Generate 49 type 3 columns
10	0 0 1	0 2	Generate 1 type 2 column
11	0 4 9	0 3	Generate 49 type 3 columns
12	0 0 1	0 2	Generate 1 type 2 column
13	0 4 9	0 3	Generate 49 type 3 columns
14	0 0 1	0 2	Generate 1 type 2 column
15	0 4 9	0 3	Generate 49 type 3 columns
16	0 0 1	0 2	Generate 1 type 2 column
17	0 4 9	0 3	Generate 49 type 3 columns
18	0 0 1	0 2	Generate 1 type 2 column
19	0 4 9	0 3	Generate 49 type 3 columns
20	0 0 1	0 2	Generate 1 type 2 column
21	0 4 9	0 3	Generate 49 type 3 columns
22	0 0 1	0 2	Generate 1 type 2 column
Total columns:	5 2 5		

[a]Numbers shown are decimal equivalents for binary contents of each word.

TABLE III. - SIMPLIFIED LIST DEFINING COLUMN TYPES

FOR GRID OF FIGURE 8

Word number	Contents of word		Meaning assigned to each word by program
	Bits 2 to 10 (number[a] of consecutive columns)	Bits 11 to 16 (column type number[a])	
1	0 2 4	0 1	Generate 24 type 1 columns
2	0 0 1	0 2	Generate 1 type 2 column
3	0 4 9	0 3	Generate 49 type 3 columns
4	0 0 0	0 0	Return to word 2 of list and repeat instructions until all 525 columns have been generated

[a]Numbers shown are decimal equivalents for binary contents of each word.

TABLE IV. - LIST DEFINING A COLUMN WITH

DOTS IN EVERY 50th LOCATION

Word number	Contents of word		Meaning assigned to each word by program
	Bit 1 (sign: - blank; + dots)	Bits 2 to 16 (number[a] of consecutive locations in group)	
1	-	0 0 2 4	Generate 24 blank locations
2	+	0 0 0 1	Generate 1 dot
3	-	0 0 4 9	Generate 49 blank locations
4		0 0 0 0	Return to word 2 of list and repeat instructions until plot is finished

[a]Numbers are decimal equivalents for binary contents of each word.

Figure 1. - Printer.

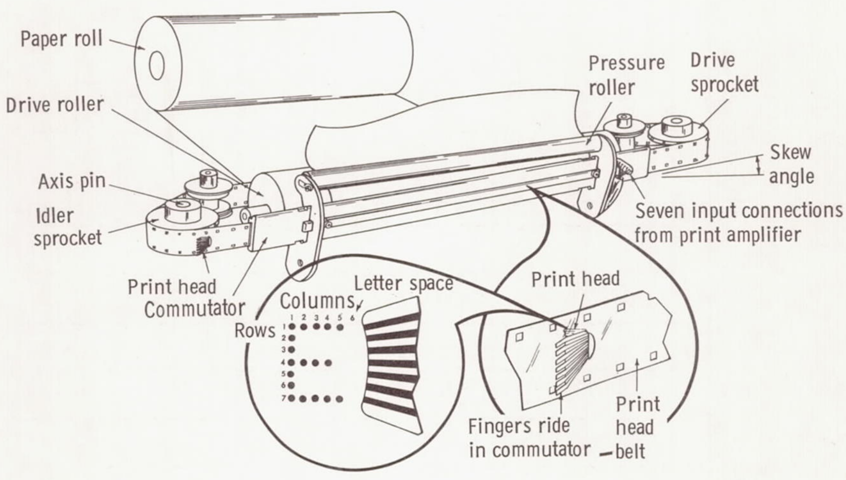

Figure 2. - Printing mechanism.

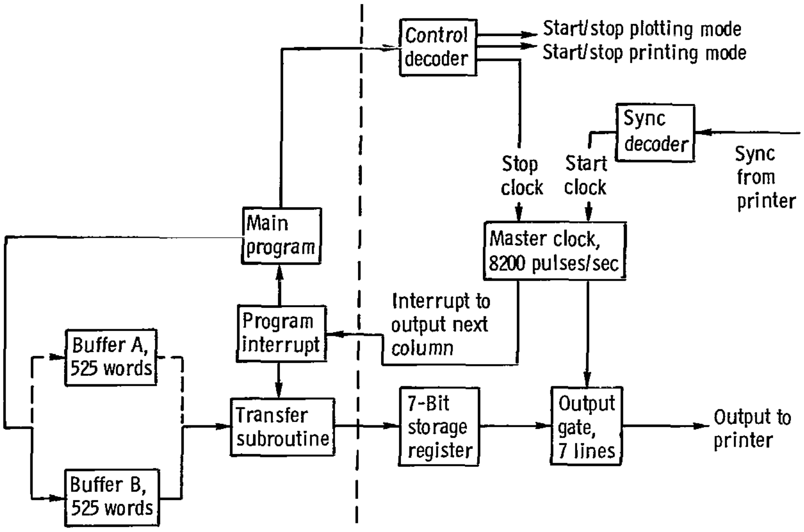

Figure 3. - Block diagram of interface unit.

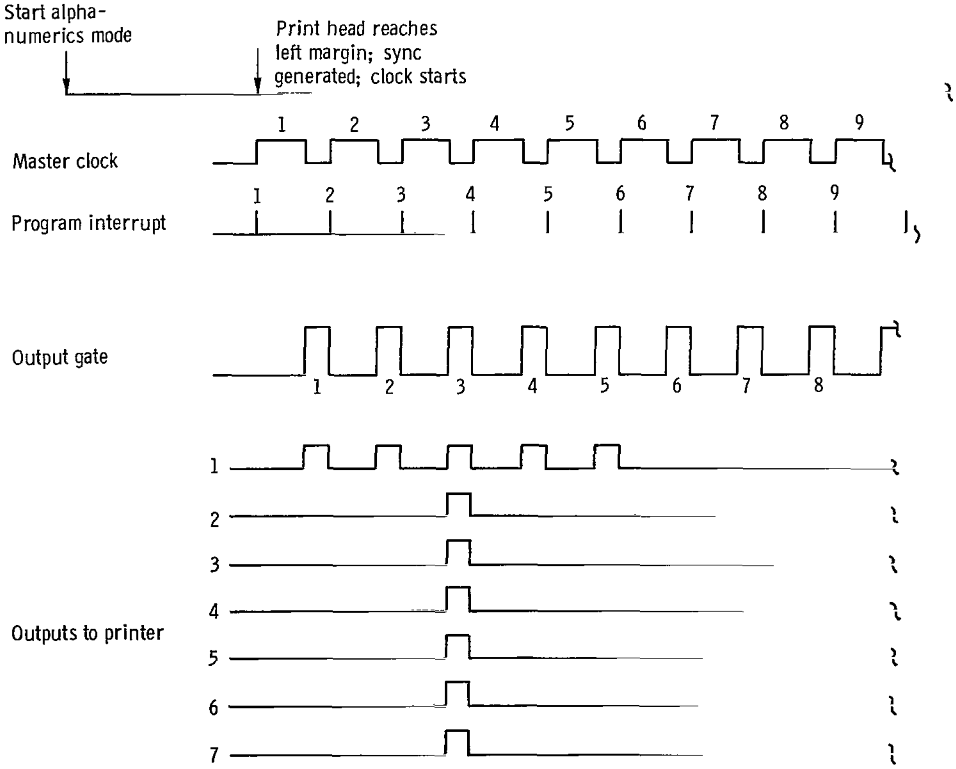

Figure 4. - Timing diagram.

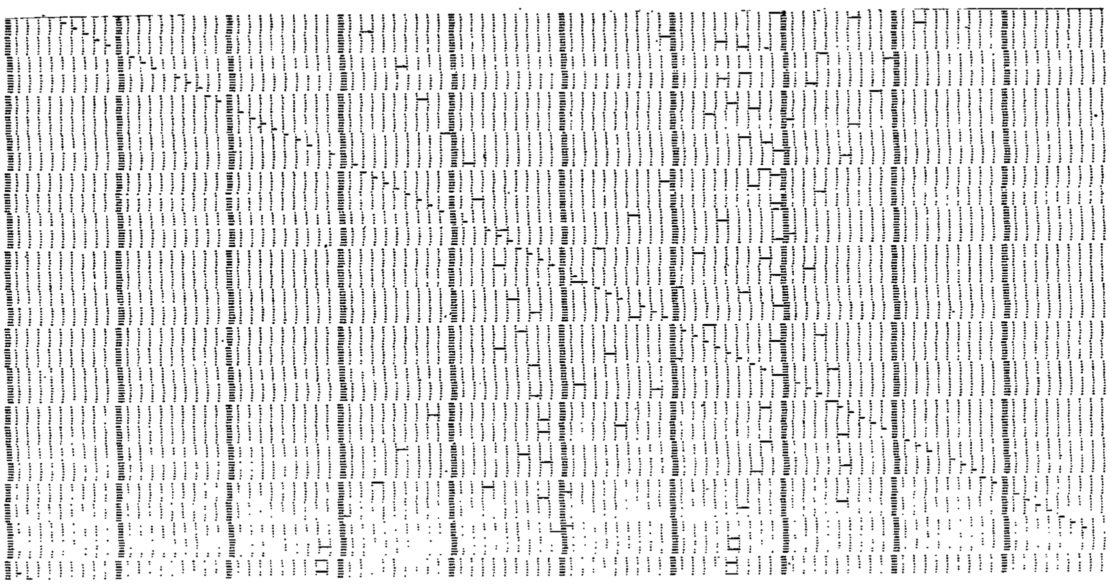

Figure 5. - Example plot using original synchronization.

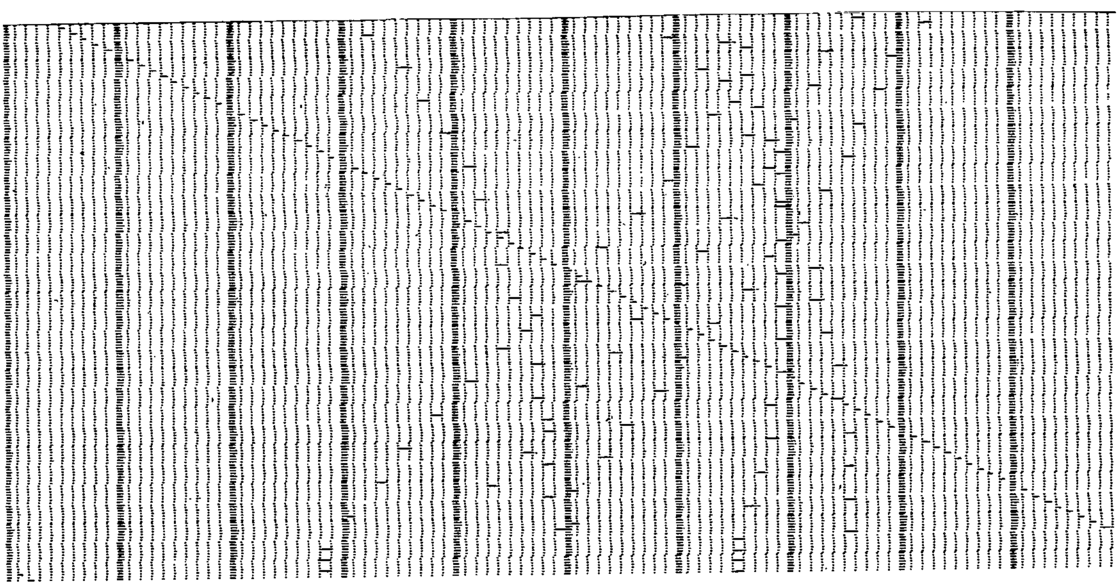

Figure 6. - Example plot using electrical synchronization.

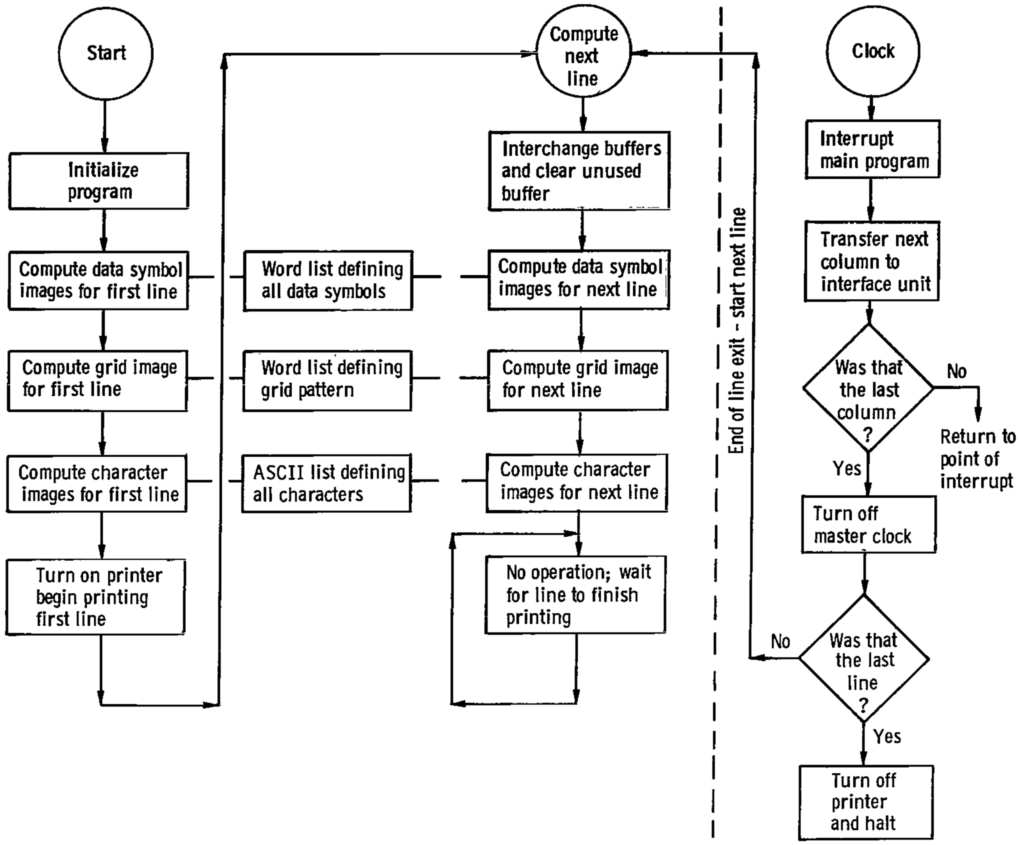

Figure 7. - Flow diagram for plotting program.

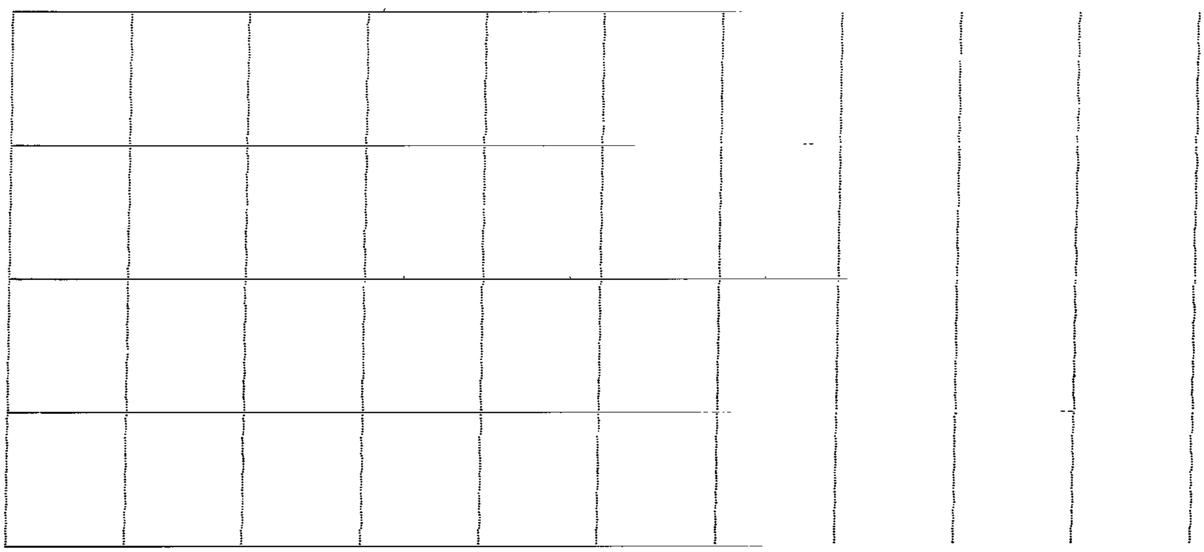

Figure 8. - Example of simple grid.

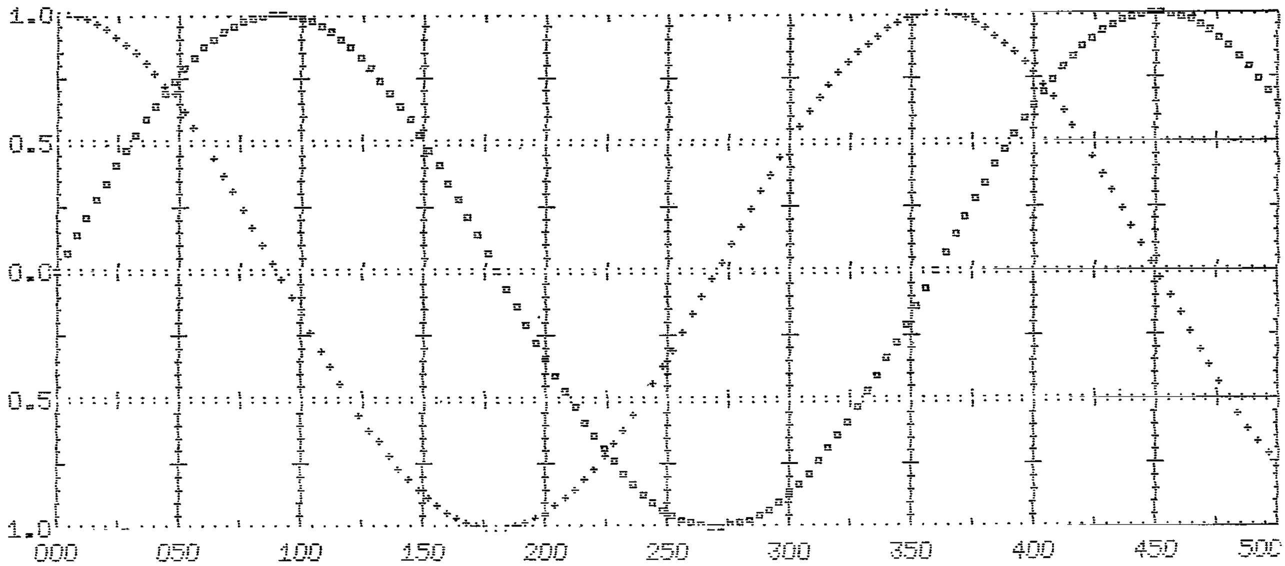

Figure 9. - Plot of sine and cosine functions.

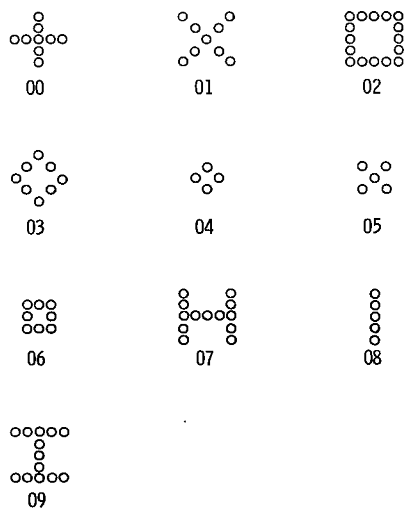

Figure 10. - Examples of symbol dot patterns.

THIS IS A SAMPLE OF THE ALPHANUMERIC PRINT OUT OF THE PRINTER/PLOTTER.
THE MAXIMUM PRINTING SPEED OF THE DEVICE IS 300 CHARACTERS PER SECOND.

AAAAAAAA	BBBBBBBB	CCCCCCCC	DDDDDDDD	EEEEEEEE	FFFFFFFF
GGGGGGGG	HHHHHHHH	IIIIIIII	JJJJJJJJ	KKKKKKKK	LLLLLLLL
MMMMMMMM	NNNNNNNN	OOOOOOOO	PPPPPPPP	QQQQQQQQ	RRRRRRRR
SSSSSSSS	TTTTTTTT	UUUUUUUU	VVVVVVVV	WWWWWWWW	XXXXXXXX
YYYYYYYY	ZZZZZZZZ	11111111	22222222	33333333	44444444
55555555	66666666	77777777	88888888	99999999	00000000
!!!!!!!!	""""""""	########	$$$$$$$$	%%%%%%%%	&&&&&&&&
''''''''	((((((((	))))))))	********	========	@@@@@@@@
::::::::	--------	;;;;;;;;	,,,,,,,,	....	////////
↑↑↑↑↑↑↑↑	←←←←←←←←	++++++++	<<<<<<<<	>>>>>>>>	????????
[[[[[[[[	]]]]]]]]	\\\\\\\\			

IN ADDITION THIS DEVICE ALSO MAKES PLOTS OF DIGITAL DATA.
A TYPICAL PLOT IS SEVEN INCHES SQUARE AND TAKES ABOUT 25 SECONDS TO PRINT.
A HORIZONTAL LINE CONTAINS 500 DOTS; POINTS CAN THEREFORE BE LOCATED
WITH A RESOLUTION OF 1 PART IN 500.
IN THE VERTICAL DIRECTION THERE ARE ABOUT 50 DOTS TO THE INCH.
THE LENGTH OF THE PLOT IN THE VERTICAL DIRECTION IS VARIABLE;
THE RESOLUTION IN THE VERTICAL DIRECTION DEPENDS ON THE LENGTH
OF THE PLOT--MAXIMUM RESOLUTION IS ONE DOT.
ANY NUMBER OF POINTS CAN BE PLOTTED WITHOUT AFFECTING THE OVERALL TIME
TO COMPLETE THE PLOT.
SEVERAL DIFFERENT VARIABLES ARE PLOTTED SIMULTANEOUSLY USING DIFFERENT
PLOTTING SYMBOLS FOR EACH VARIABLE.
A VARIETY OF DIFFERENT GRID PATTERNS CAN BE CALLED FROM A PERMANENT LIBRARY.
SPECIAL GRIDS ARE EASILY PROGRAMMED IF NECESSARY.
ALPHANUMERIC INFORMATION CAN BE INCLUDED ON THE PLOT IF DESIRED:
AXES LABELS
SPECIAL NOTES
TITLES
NUMERICAL SCALES FOR AXES
PLOTS CAN BE MADE WITH ABSCISSA IN HORIZONTAL DIRECTION,
OR WITH ORDINATE IN HORIZONTAL DIRECTION, SO THAT THE VARIABLE
LENGTH AXES CAN BE ASSOCIATED WITH EITHER X OR Y

Figure 11. - Example of alphanumeric printing.

NASA-Langley, 1969 —— 8 E-4863

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

— NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Washington, D.C. 20546