G101—A Remote Time Share Terminal with Graphic Output Capabilities

STEPHEN PARDEE, MEMBER, IEEE, PETER E. ROSENFELD, AND PATRICK G. DOWD

Abstract—A remote terminal with graphical output capability has been designed for use with time-shared computer systems. The terminal contains a small computer to provide local picture storage and the ability to employ picture encoding techniques which greatly reduce transmission times. A storage oscilloscope is used to view the output. Terminal cost is currently about $10,000.

Index Terms—Computer graphics, computer terminals, graphic terminals, graphic transmission codes, mini-computers, remote terminals, storage scope, time sharing.

I. INTRODUCTION

For some time now many users of time-sharing systems have felt the need for graphic terminal facilities that had the following qualities:

1) provides both character and vector information;
2) is a compatible replacement for existing teletypewriter-type terminals;
3) meets ASCII standards;
4) is capable of very efficient transmission of complex pictures over 110 Baud dialed up telephone lines;
5) is relatively inexpensive ($10,000 to $15,000).

In the past the available terminals that provided both character and vector information were either too expensive or were too slow at picture transmission. The G101 described here satisfies all of the above criteria by employing a storage CRT in conjunction with a small computer in place of a wired controller. The storage CRT eliminates the need for refreshing the display. The small computer provides local picture storage plus the ability to employ picture encoding techniques that result in greatly reduced transmission times compared to the use of a hard-wired controller.

The idea of using a local computer in conjunction with large interactive graphic systems has been widely employed [1]–[3]. Similarly the use of a storage CRT for time-sharing terminals has been widespread [4]–[6]. However, at the present time, no commercially available terminal takes advantage of both techniques to provide a flexible, efficient, but still inexpensive, terminal.

This paper will emphasize the particular hardware configuration of the G101, the specific techniques used to compress transmission time, and the compatability features with existing time-sharing systems.

II. HARDWARE CONFIGURATION

The G101 terminal hardware configuration (see Fig. 1) consists of a Tektronix 611 storage CRT, a PDP-8/L computer with an ASR 33 teletypewriter, and a type 103 Data-Phone data set. The data set and storage CRT are interfaced to the computer using Digital Equipment Corporation M-series modules. The scope interface consists of a 10-bit X register and D/A converter, an 11-bit Y register and D/A converter, and a Z-axis control. The data set interface consists of serial-to-parallel and parallel-to-serial converters, timing and control circuits, and a parity check circuit.

Fig. 1. G101 hardware configuration.

The origin of the display coordinate system is in the lower left corner of the display. Because of the aspect ratio of the CRT, values of X can range from 0 to 1023 display units, and values of Y can range from 0 to 1435 display units. The terminal is built into a standard desk-type console as shown in Fig. 2.

Fig. 2. G101 terminal.

III. SYSTEM DESCRIPTION

The G101 terminal operates by receiving strings of ASCII characters and has two basic modes of operation, teletypewriter-
writer mode and graphic mode. In the teletypewriter mode, the local computer acts as a normal transmission line between the teletypewriter and the time-sharing computer. In this mode the G101 responds exactly as a standard teletypewriter.

In the graphic mode, the ASCII characters received from the time-sharing computer are interpreted as either graphical commands or data, are stored in the memory of the local computer, interpreted, and displayed on the CRT.

The switching between modes occurs upon receipt of one of four special control characters. Three of the control characters are used to select one of three possible states within the graphic mode. The graphical character state is entered upon receipt of the FS control character (034). In this state all ASCII characters received are interpreted as standard ASCII symbols and are drawn on the display as such. This allows all 96 printable ASCII characters to be displayed.

Receipt of the ESC control character (033) will put the terminal into the rotated character state. This is identical with the graphical character state except that all characters are rotated 90° in the positive direction.

The third state of the graphic mode is the graphical master state. It is entered upon receipt of the GS control character (035). Once in this state all succeeding ASCII characters are interpreted as graphical operation codes or data. An ASCII character is treated as an op-code if its most significant bit is a 0 and as graphical data if it is a 1. This allows up to 32 op-codes. Since each ASCII character has seven bits (plus parity) and one bit is used to differentiate between op-codes and data, only six bits are available for data. For these op-codes requiring data that exceed six bits, two successive ASCII characters are concatenated as shown in Fig. 3 to provide up to 12 bits of data.

To return to the teletypewriter mode from the graphical mode, the terminal must receive the DC1 control character (021) from the time-shared computer.

IV. TECHNIQUES FOR TRANSMISSION COMPRESSION

The use of a small local computer opens up many possibilities to employ techniques that compress the transmission time for a given picture. In particular, it makes possible the following:

1) the use of graphic subroutines;
2) the ability to replicate at the terminal a string of ASCII characters;
3) the ability to selectively store, erase, and re-display portions of a picture;
4) software expansion of data.

A. Graphic Subroutines

A graphic subroutine is a portion of a picture that may appear many times within the overall picture. Rather than transmit the subpicture each time it is to be displayed, a great deal of time can be saved if the picture is transmitted once as a graphic subroutine, stored in memory, and then invoked as many times as necessary. For example, an application that was to draw logic diagrams might transmit each required logic symbol as an individual graphic subroutine. A particular logic diagram could be constructed by "calling" the appropriate subroutines.

Each time a particular graphic subroutine is called, only two ASCII characters need be transmitted from the time-sharing computer to the G101 terminal. The first character is the "call subroutine" op-code and the second character contains the identifying number that was assigned to the graphic subroutine when it was transmitted to the G101. Therefore, by transmitting two ASCII characters, a portion of a picture that might otherwise require many more characters can be displayed. As an example, consider the simple logic diagram element shown in Fig. 4. The transmission of 36 ASCII characters is required to transmit the complete symbol. By using the subroutine approach, the transmission time is reduced by a factor that approaches 18 to 1 if the symbol appears quite often in the diagram to be displayed.

B. Graphic Replication

Quite often it is desirable to simply repeat a particular portion of a picture a certain number of times. This is accomplished by preceding the characters that define the portion to be repeated by two ASCII characters, the first of which is the REPEAT op-code and the second of which indicates the number of times the portion is to be repeated. The repeated sequence is terminated by the END REPEAT op-code. Because of the memory and processing capabilities of the local computer, it is possible to store the characters that define the portion to be repeated and then to repeatedly scan the characters to develop the desired display.

As an example, consider the problem of displaying the grid upon which a graph is to be plotted. Eight characters are required to define one line on the grid and position the writing beam at a point ready to draw the next line. If the grid is made up of 20 such lines, in say the vertical axis, 160 characters would have to be transmitted. By taking advantage of the repeat features, only 12 characters would need to be transmitted.

C. Selective Storage, Erase, and Re-Display

Two display storage areas are provided in the G101's local memory. These buffers are usually used alternately, thus allowing the user to view either of the two most recently transmitted displays without requiring any additional transmission. However, the user can alter the sequence in which the buffers are used and thus save the contents of one buffer indefinitely. This ability to save a display indefinitely can also be used to reduce transmission time. Consider a circuit analysis program whose output is a series of frequency response plots. A significant part of the transmission...
time for a given plot would be consumed in transmitting the coordinate grid (usually semilog) and axis labels. The curve itself might only represent 10 to 20 percent of the total transmission time. By storing the grid and labels in one buffer and the curves themselves in the other buffer, the display can be erased and a "clean" grid re-displayed from the terminal’s memory, thereby eliminating the re-transmission time.

D. Software Expansion of Data

Rectangles and circular arcs are two elements that often appear in pictures. They can be transmitted as a series of vectors, but this is time-consuming especially for the arcs. In the G101, rectangle and arc op-codes are used to transmit the required parameters. The local computer then generates the required sequence of vectors. For rectangles this reduces transmission times by 5/9. For a circle the improvement is dependent on the diameter and is typically of the order of 50 times faster.

The plotting of data can also be speeded up by defining special op-codes. For point plotting, where the data are spaced equally along the X axis, the use of an op-code that requires specification of the plotting character and X increment only once speeds things up by 4 to 1. For plotting using straight lines to connect the data points, a 2 to 1 improvement can be obtained.

V. G101 Operation Codes

The actual op-codes implemented can also contribute to transmission efficiency if they are properly chosen. In the G101 these op-codes can be divided into four categories:

1) control;
2) beam control;
3) drawing;
4) characters.

A short review of these op-codes may be helpful in understanding the capabilities of the G101. A list of the available op-codes is shown in Table I.

A. Control Op-Codes

There are three op-codes associated with graphic subroutines. Each subroutine is preceded by a SUB ID op-code and a single data character assigning an ID number to the graphic subroutine. A subroutine is terminated by the END SUB op-code. To call a graphic subroutine, the CALL SUB op-code is transmitted followed by a single data character indicating the ID number of the subroutine. A graphic subroutine may CALL other graphic subroutines, and this nesting may be eight levels deep. All subroutines used in a picture must be transmitted at the beginning of the picture.

The end of a particular display is indicated by transmitting the END DISPLAY op-code. The ERASE op-code causes the picture presently being displayed to be erased. The contents of memory are unaffected by ERASE.

Repetition of a sequence of graphic op-codes and data is accomplished by preceding the sequence with the REPEAT op-code plus two data characters indicating the number of times the sequence should be repeated. The sequence is terminated by the END REPEAT op-code.

Control of the buffer used to store a particular picture is exercised through the use of the SELECT BUFFER op-code. The data character that follows this op-code specifies which buffer is to be used and also whether the other buffer is to be protected in the event that the picture being transmitted is larger than the selected buffer. If the unselected buffer is not protected, memory space is dynamically allocated to the selected buffer if needed.

B. Beam Control Op-Codes

The absolute position of the CRT writing beam can be specified by using the SET X, SET Y, or SET XY op-codes. The SET X and SET Y op-codes are followed by two data characters which, taken together, specify the new X or Y coordinate of the beam. If both X and Y are to be modified, the SET XY op-code is used followed by four data characters giving the new X and Y coordinates. The beam may be repositioned incrementally without intensification from its current position by using the INVISIBLE X or INVISIBLE Y op-codes. INVISIBLE X can be followed by one pair of data characters which specify the X increment or by two pairs of data characters which specify X and Y increments. Similarly INVISIBLE Y can use one pair of characters for Y increment and two for Y and X increments. At any time the beam can be intensified by using the INTENSIFY op-code.

C. Drawing Op-Codes

There are a number of op-codes that will cause picture information (as opposed to text) to be displayed on the CRT. The VECTOR and SHORT VECTOR op-codes are followed by an unlimited number of X, Y data pairs. Visible vectors are strung head-to-tail connecting each specified coordinate. The only difference is that SHORT VECTOR restricts X's or Y's to ±31 display units and uses only one data character.

In many situations, it is desirable to specify a series of vectors that alternate between being parallel to the X and

<p>| TABLE I |
|-------------------|------------------|
| <strong>SUMMARY OF OPERATION CODES</strong> |
|-------------------|------------------|</p>
<table>
<thead>
<tr>
<th>Control</th>
<th>Drawing</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUB ID</td>
<td>VECTOR</td>
</tr>
<tr>
<td>END SUB</td>
<td>SHORT VECTOR</td>
</tr>
<tr>
<td>CALL SUB</td>
<td>VISIBLE X</td>
</tr>
<tr>
<td>END DISPLAY</td>
<td>VISIBLE Y</td>
</tr>
<tr>
<td>ERASE</td>
<td>RECT</td>
</tr>
<tr>
<td>REPEAT</td>
<td>ARC</td>
</tr>
<tr>
<td>SELECT BUFFER</td>
<td>POINT PLOT 1</td>
</tr>
<tr>
<td>Beam Control</td>
<td>POINT PLOT 2</td>
</tr>
<tr>
<td></td>
<td>LINE PLOT</td>
</tr>
<tr>
<td></td>
<td>LINE TYPE</td>
</tr>
<tr>
<td></td>
<td>Character</td>
</tr>
<tr>
<td></td>
<td>SIZE</td>
</tr>
<tr>
<td></td>
<td>-------------------</td>
</tr>
<tr>
<td><strong>INVISIBLE X</strong></td>
<td></td>
</tr>
<tr>
<td><strong>INVISIBLE Y</strong></td>
<td></td>
</tr>
<tr>
<td><strong>INTENSIFY</strong></td>
<td></td>
</tr>
</tbody>
</table>
then the $Y$ axis. The visible $X$ and visible $Y$ op-codes accomplish this type of plotting. With these op-codes, it is only necessary to transmit a single data value ($X$ or $Y$) for each vector that is to be displayed. The visible $X$ op-code will cause the first vector in the series to be drawn parallel to the $X$ axis and visible $Y$ will start the series in the $Y$ direction.

The four vector op-codes discussed above can be used to draw almost any conceivable display. However, a number of special op-codes have been included to handle situations that occur frequently. RECT and ARC allow rectangles, arcs, and circles to be specified in a compact fashion. To facilitate the plotting of curves, three special op-codes have been provided. POINT PLOT 1 allows the user to specify a symbol that is to be plotted at a series of points. The data following the op-code is the symbol to be plotted, the $\Delta X$ between successive points, and a series of $Y$ values. LINE PLOT is similar to POINT PLOT 1 except that the symbol is omitted and the points are connected with visible lines. POINT PLOT 2 is similar to the above except that the assumption of equal intervals in the $X$ direction is eliminated and the symbol data character is followed by a series of absolute $X$, $Y$ coordinate pairs specifying where the symbol is to be plotted.

It is often desirable to overlay many figures or graphs on one display. To facilitate this overlaying, seven different line styles are available. To change from one style to another, the line type op-code is used, followed by a single data character that indicates which of seven line styles should be employed. Table II contains the seven possible styles. The solid line is the default style.

D. Character Op-Codes

As described earlier there are two character states, normal and rotated. When the G101 terminal is in either state, all characters received from the time-sharing computer are interpreted as displayable ASCII symbols until another control character removes the G101 from a character state. The size of the characters being displayed is controlled by the size op-code. The size op-code is followed by a single character that indicates a size between 1 and 63 (2 is the normal default size). All characters are displayed as a 5 by 7 matrix of dots.

VI. ADDITIONAL VIRTUES

Since the G101 terminal is based on the use of a stored program computer rather than a hard-wired controller, it contains an unusual amount of flexibility. For instance, by loading different control programs the terminal can be made to simulate the ARDS, Tektronix, or Computek terminals. This also allows for expanding or modifying the G101 op-code set as has already been desirable. Finally, the terminal can be used stand alone for a number of jobs that are within the capability of the PDP-8 computer.

VII. CONCLUSIONS

The G101 terminal was designed to fill the gap between inexpensive displays that were slow and inflexible, and very expensive displays that had the desired speed and flexibility. Since March 1969, it has provided developers at Bell Laboratories with a graphic terminal that allows for simple interaction, is usable with almost any time-sharing system, and makes very efficient use of the limited transmission bandwidth available from most time-sharing systems. At the present time, 11 of these graphic terminals are in use at various Bell Laboratories' locations and have been used on such diverse problems as circuit analysis, lens design, particle motion in magnetic fields, and geometry research.

The cost goal of the G101 was to provide a terminal that could be purchased for between $10,000 and $15,000. This goal has been met by the terminal described in this paper.

REFERENCES