RCSL No: 52 - AA1224

Edition: 1983.11.29

Author: Knud Erik Hansen

Title:

Programmers Reference Manual
for VPB701
Video Graphic Processor board
to RC702/703

A/S REGNECENTRALEN
af 1979

Keywords: RC702, RC703, VPB701, Programmers Reference Manual

Abstract: This paper gives all details about the graphic option to RC702/703, necessary from the programmers point of view.

(56 printed pages).

Printed by A/S Regnecentralen af 1979, Copenhagen

Users of this manual are cautioned that the specifications contained herein are subject to change by RC at any time without prior notice. RC is not responsible for typographical or arithmetic errors which may appear in this manual and shall not be responsible for any damages caused by reliance on any of the materials presented.

RCSL [illegible]2-I 2164

TABLE OF CONTENTS PAGE

APPENDICES:

1. DESCRIPTION 1.

1.1 Introduction 1.1

The VPB701, Video Processor Board, adds bit-mapped raster scan graphic drawing and display functions to RC702/703 microcomputer family. The VPB701 is designed for installation as a piggy-back to the printed circuit board internal RC702/703.

The VPB702 can function in tw› modes:

1. 560x275 pixels B/W graphic video ored with the normal video output from RC702/703, generating a mixed graphic/character picture on the standard RC702/703 monitor.

2. 256x256 pixels color video on an external color monitor. In this mode VPB701 allows selection of up to eight colors each for display and background.

1.2 General Description 1.2

The VPB701 VGC board attached to the RC702/703 host processor board via the standdard eight-bit Z80 databus and uses normal programmed I/O instructions.

The processor passes commands and data to the VPB701 board. The VPB701, upon receipt of the commands, performs all of the tasks needed to manage the display memory. In addition to the display function, the VPB701 board implements high-speed, vector-to-raster scan conversion.

The commands sent to the VPB701 board consists of a series of instructions and their associated parameters. These commands and their associated parameters are used to construct a geometric figure or graphic character into the display memory bit-by-bit. This geometric figure or graphic drawn in the display memory can then be displayed on the monitor. The drawing commands which define the elements of the figure include lines, arcs, circles, rectangles, character/symbol painting, and area fills. The commands also allow user-selected line styles and patterns with which to draw and display figures on the monitor.

The processor (under control of the user˝s graphic software program) performs the preliminary calculations to prepare memory and other associated parameters of the figure to be drawn. The VPB701 calculates the display memory addresses bit-by-bit for the graphic figures to be drawn and completes the drawing without further intervention from the processor.

For example, if a rectangle is to be drawn on the attached monitor, the VPB701 board is given a command which defines the type of drawing followed by the starting address and several other parameters that define the length and height of the rectangle. The VPB701 board is then given the start drawing command to draw the rectangle. The VPB701 board takes the command and draws the figure in the display memory.

1.3 Functional description 1.3

The VPB701 board consists of five logical blocks as shown in Figure 1. These five logic blocks of the VPB701 board are as follows:

- Graphic Display Controller
- Control Logic
- Timing Logic
- Display Memory (16K x 16-bit static RAM array)
- Shift Register and Output Buffer

1.4 Graphic Display Controller

The heart of the VPB701 board is the NEC7220 Graphic Display Controller (GDC) chip. The Graphics Display Controller (GDC) chip is an intelligent microprocessor peripheral chip designed as a raster-scan computer graphics controller. Control of the GDC by the host microprocessor is achieved via an eight-bit bi-directional bus interface. This bi-directional bus interface allows the host processor to send command and parameter bytes to start the drawing process. The host processor begins the process by retrieving status information from the VPB701 board concerning the drawing process. This status information is readable at any time by the host processor. Access to the GDC is coordinated via flags in the status register.

The First In First Out (FIFO) buffer is integral to the GDC and is the GDC´s interface to the microprocessor bus through which all commands and data pass. Access to the 16-byte FIFO is controlled by the host processor via the GDC´s command set. The host processor coordinates the transfers by checking the appropriate status register bits. Commands and parameter bytes are sent to the GDC´s FIFO and are differentiated based on the state of the address bit 0. All commands are written to the GDC. When the write control line is activated, the GDC configures the FIFO as an input buffer to receive the command and the parameter bytes. If a read command is issued to the GDC, the GDC interprets the command, configures the FIFO as an output buffer and places the requested data into the FIFO. The status register indicates that the data is ready to retrieve. The host processor can then retrieve the requested information by activating the read control line.

When the host processor sends a command byte and activates the write control line (IOWR) to the GDC, the GDC interprets the contents written into the FIFO, decodes the command, distributes the succeeding parameter bytes to the proper registers within the GDC, and then initiates the required operation. any parameter byte following a command is truncated by the receipt of the next command.

The host processor (under control of the user´s software) programs the GDC sync logic during initalizations with either a reset command or a sync command and the associated parameters. The GDC generates the sync signals (horizontal and vertical) as well as the blanking signals to the CRT monitor in color mode. In Black/White mode RC702/703 generates the sync pulses.

1.5 Timing Logic 1.5

This circuit uses the dot frequency from RC702/703 to generate internal timing pulses to the VPB701 logic.

In Black/White mode, the sync pulses (VRTC and HRTC) from RC702/703 is used to generate the graphic picture.

In color mode the 7220 is free running, generating the sync pulses by itself.

In Black/White the frequency to the 7220 is 0,7 Us, which gives a RMW (Read-Modify-Write) cycle time of 2.8 Us (or a drawing speed fo 350.000 pixels /sec.).

In color mode the drawing speed is 200.000 pixels/sec.

1.6 Control Logic 1.6

The control logic consists of the Control PAL which aids the GDC in controlling access to the display memory. There are two types of accesses to the display memory; Display cycles and Read-Modify-Write (RMW) cycles. During display cycles, the data at the addressed location is read and sent to the shift register for subsequent display on the monitor. During a RMW cycle, data is accessed from display memory, modified, and written back into the display memory. The modifications done to the data at the addressed location are: complement, set, clear or replace. In Black/White mode the Display cycle is not executed by the 7220, but by hardware circuits controlled by VRTC and HRTC pulses from RC702/703.

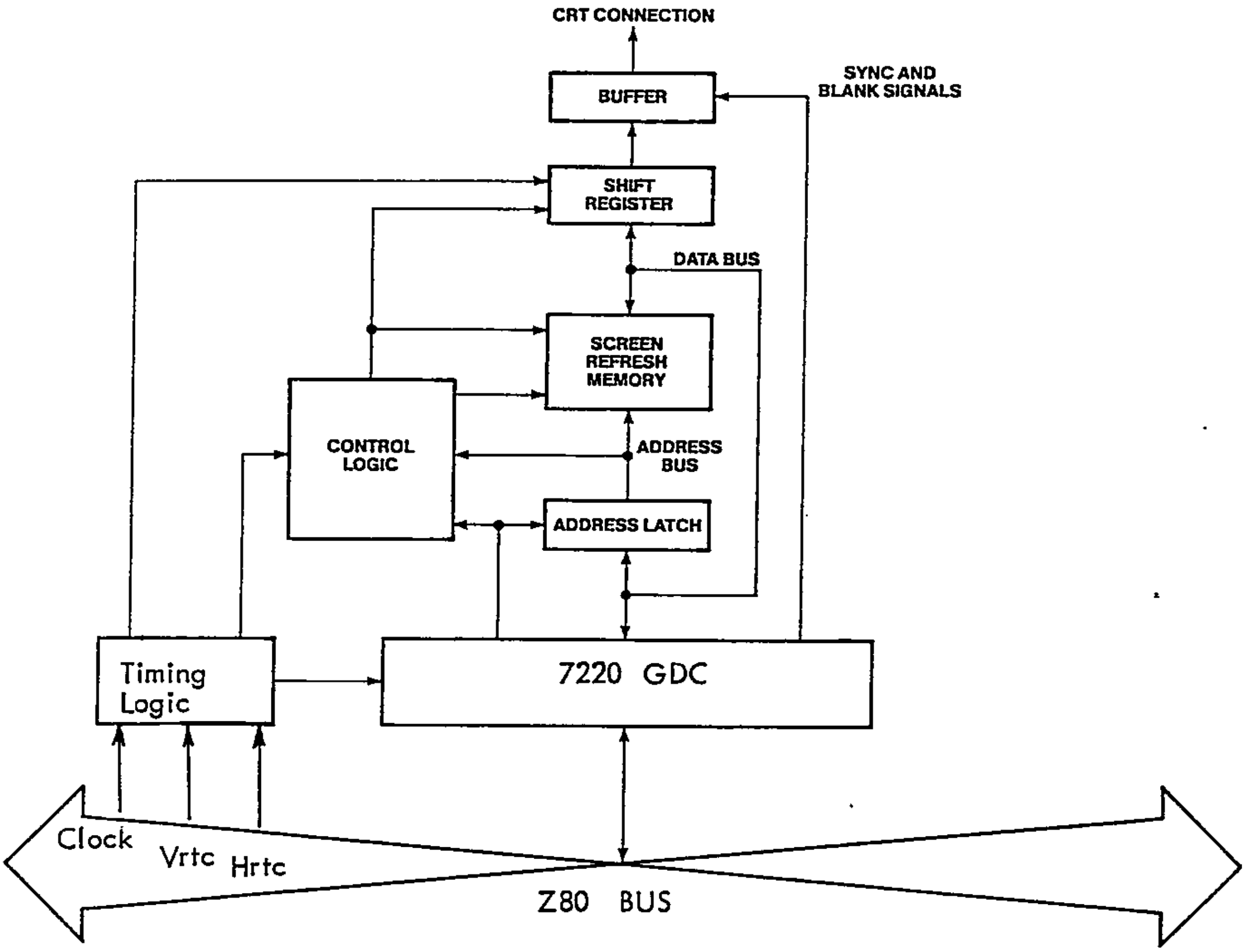


Fig. 1. Functional Block Diagram

1.7 Display Memory 1.7

The display memory consists of sixteen 16K x 1 static RAMs which provide 256K bits of memory arranged as 16K sixteen-bit words. The address of the display memory access is calculated by the GDC and latched in the address latch. The data stored at the addressed location may be modified by the GDC, passed back to the host processor, or displayed on the attached monitor.

When the data stored at the addressed RAM location is to be displayed, the data is parallel loaded into the shift register.

1.8 Shift register and buffer logic 1.8

The Shift Register logic consists of three 16-bit registers (one for each color) and a series of D flip-flops. Each shift register performs the parallel-to-serial conversion that produces the video bit stream sent to the attached monitor. Additional D-type flip-flops are used to resynchronize the video data with the synchronizing signals and blanking signals out to the connector. In the black and white mode, only one shift register is used, and the video from the graphic logic is ored with the video from RC702/703.

2. PROGRAMMING INFORMATION 2.

2.1 Introduction 2.1

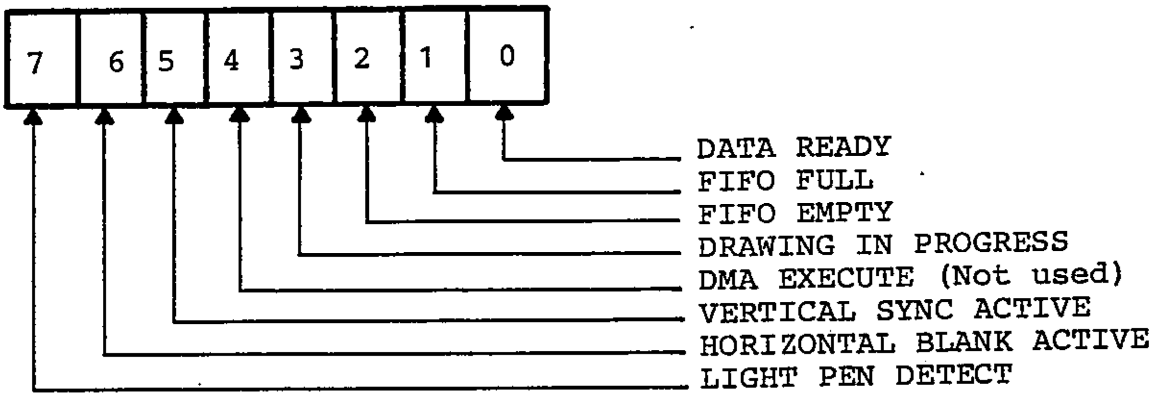
This chapter provides programming information for the VPB701 Graphics Controller board. All on-board functions are controlled by the Graphics Display Controller (GDC) chip.

2.2 Addressing 2.2

The VPB 701 board is accessed in the I/O space of the host processor using read and write command. Figure 2 gives the addresses of the Command, Status and Data ports used by the VPB701 board. Before data or commands are written from the host processor, the host processor should check the status register to ensure that the command/data FIFO is not full. Figure 3 defines each bit of the Status Register.

8 Bit Z80 bus	READ (IN) TO CPU	WRITE (OUT) FROM CPU
C8	status	Data,(Parameters
C9 Hex	Data	Command
CC Hex		Disable graphic video.
CE Hex		Enable graphic video.

Fig. 2. I/O Port Address Locations



Light Pen Detect. When this bit is set to 1, the light pen address (LAD) register contains a value (word address) that the system microprocessor can read. This flag is reset after the three bytes of Light Pen Address (LAD) is moved into the FIFO in response to the light pen read command.

Horizontal Blank Active. A value of 1 for this flag signifies that horizontal retrace blanking is currently underway.

Vertical Sync Active. This flag is a 1 during vertical retrace. The vertical sync flag coordinates display-format-modifying commands to the blanked interval surrounding vertical sync. This eliminates display disturbances.

DMA Execute. This bit is a 0 since DMA is not supported.

Drawing in Progress. This status flag is set to a 1 while the GDC is drawing a graphics figure.

FIFO Empty. This flag and the FIFO full flag coordinate system microprocessor accesses with the GDC's FIFO. When it contains a 1, the empty flag ensures that all the commands and parameters previously sent to the GDC have been processed.

FIFO Full. When this flag contains a 1, it indicates a full FIFO in the GDC. A value of 0 ensures that there is room for at least one more byte. This flag must be checked before each write into the GDC.

Data Ready. When this flag is set to 1, it indicates that a byte is available to be read by the microprocessor. The data-ready bit must be tested before each read operation by the system microprocessor. The flag is reset after the data is transferred from the FIFO into the microprocessor.

Fig. 3. Status Register Bit Definition

2.3 Interrupts 2.3

The VPB701 does not generate interrupt to the host microprocessor.

2.4 Display memory organization 2.4

The on-board bit-map display memory consists of 16K words of 16 bits each beginning at address 0 relative to the Graphics Display Controller (GDC) chip. This memory is read out 16 bits at a time by the GDC and displayed on the screen starting at the top, left corner of the screen and sequences down toward the bottom right corner of the screen. Because the display is read a word at a time, the display resolution in the horizontal direction must always be an even multiple of 16. The least significant bit of the word is displayed first. The GDC chip has a total of 18 address bits, but the VPB7801 board uses the 14 least significant bits. The additional upper bits are ignored, but should be set to zero to avoid complicated memory address calculations.

While in black and white mode, all 16K words of the display memory are in a contigous block which may be written to and displayed. A value of 1 written into the bit map is displayed as an illuminated dot.

When in the color mode, three color planes exist sequentially in memory for purposes of writing, but are displayed simultaneously. The planes are each 4K words long and are in the order of Red, Blue and Green. The Red plane starts at memory address 0; the Blue plane starts at memory address 1000H; and the Green plane at memory address 2000H. A value of 1 written into a given plane illuminates that color dot. Figure 4 lists the color plane combinations for the various colors. If a figure is to be drawn in a color that is composed of two or more primary colors, the GDC chip must be instructed to draw the figure separately for each color plane. This is done by changing the cursor base address and repeating the drawing command and parameters.

COLOR PLANE			DISPLAYED COLOR
GREEN	BLUE	RED	
0	0	0	BLACK
0	0	1	RED
0	1	0	BLUE
0	1	1	MAGENTA
1	0	0	GREEN
1	0	1	YELLOW
1	1	0	CYAN
1	1	1	WHITE

Fig. 4. Color Plane Combinations

2.5 Command Summary 2.5

The following is a summary of all the commands available with the VPB701 board. These commands are organized into four categories: Video Control commands, Display Control commands, Drawing Control commands, and Data Read commands and are described in subsequent paragraphs.

RESET COMMAND	Reset the GDC to its idle state.
VSYNC	Specifies the video display format.
SYNC	Specifies the cursor and character row heights.
PITCH	Specifies the width of the horizontal dimension of display memory.
ZOOM	Specifies the zoom factors for graphics character writing.
PRAM	Defines the starting addresses and lengths of the display areas and specifies the eight bytes for the graphic character.
CURS	Sets the position of the cursor in display memory.
START	Ends the idle mode and unblanks the display.
BCTRL	Controls the blanking and unblanking of the display.
MASK	Sets the mask register contents.
FIGS	Specifies the parameters for the drawing processor.
FIGD	Draws the figure specfied into the display memory.
GCHRD	Draws the graphics character into the display memory.
WDAT	Writes data words or bytes into the display memory.
RDAT	Reads data words or bytes from the display memory.
CURD	Reads the cursor position.
LPRD	Reads the light pen address.

2.6 Video Control Commands 2.6

The Video Control commands are used to initialize the VPB701 board after a power up sequence. Normally, these commands do not need to be repeated after being executed once; however, executing these commands puts the GDC in a known state. The commands are listed in the sequence in which they are normally executed.

2.7 RESET Command (00H) 2.7

The Reset command (Figure 5) is used to initialize the operation and display modes of GDC. It resets the GDC to its idle state and specifies the video display formats when executed with the appropriate parameters. When executed without parametes, it simply causes the display to blank and the GDC chip to remain idle; no previous parameters are changed. To ensure that this command is received VPB701, ignore the status of the First In/First Out (FIFO) register. This command clears the FIFO and terminates any previous command. Figure 6 defines the eight parameter bytes following this command.

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	0	0	--RESET COMMAND (00H)
0	0	0	F	I	0	G	S	P1- Mode of operation
A W								P2- Active Display words/line
VS_L			HS					P3- Horizontal and Vertical Sync Width
HFP						VS_H		P4- Horizontal Front Porch Width
HBP								P5- Horizontal Back Porch Width
VFP								P6- Vertical Front Porch Width
		AL_L						P7- Active Display Lines/Video Field
VPB						AL_H		P8- Active Display Lines/Video Field

Fig. 5. Reset Command and Parameter Bytes

Fig. 6. Definition of Reset Parameter Bytes

Byte Function

1. This byte defines the mode of the VPB701 VGC board as follows:

0	0	0	F	I	0	G	S

I	S		
0	0	=	Non- Interlaced
0	1	=	Invalid
1	0	=	Interlaced Repeat Field for Character Displays
1	1	=	Interlaced

G { 0 = Mixed graphics and character mode
  1 = Graphics mode

F { 0 = Drawing during active display time and retrace blanking
  1 - Drawing only during retrace blanking

2. The value of this parameter byte specifies the number of active words per line (the horizontal dot resolution divided by 16). It must be an even number and two less than the value desired; if a resolution of 256 dots is required, the value given for this parameter = 14 decimal or OEH (256 divided by 16 equals 16; 16 minus 2 equals 14).

3. The value for the upper portion of this parameter byte specifies the horizontal sync width in words. (With a crystal oscillator of 12.6 MHz, one word time is 1.27 microseconds for black and white or 2.54 microseconds for color.) The value must be programmed to one less than the desired value, i.e., if a horizontal sync width of 5 microseconds is required in color mode, the parameter value is 5/2.54 = 2; 2 minus 1 = 1.

4. The value for the upper portion of parameter 3 and the lower portion of parameter 4 is the vertical sync pulse width in lines. This value specifies the number of lines that the VSYNC signal is active. The time period of a line is calculated as LT = WT x (AW + HFP + HBP + HS). Where : LT = line time, WT = word time, AW = active display words/line, HFP = horzontal front porch width (in words), HBP = horizontal back porch (in words), and HS = horizontal sync width (in words).

The value for the lower portion of parameter 4 is the horizontal front porch width in words. This value must be one less than the desired value . If a light pen is to be used, the minimum horizontal front porch width must be six words. This results in a minimum parameter value of five words.

5. The value for parameter 5 specifies the horizontal back porch width in words. It must be one less than the desired value. The minimum horizontal back porch width is 3 words, resulting in a minimum parameter value of 2 words.

6. The value for parameter 6 specifies the vertical front porch width in lines.

7. The value for parameter 7 and the lower portion of paramter 8 specifies the number of lines displayed (the Y resolution) if in the non-interlace mode of operation. If the interlace mode of operation is used, this value is YRES/2.

8. The value for the lower portion of this parameter along with the upper portion of parameter 7 specifies the number of lines displayed (the y resolution). The upper portion of this parameter specifies the vertical back porch width in lines.

In the following table is listed the reset parameters in B/W mode with RC702/703.

```
G   =   1;
I   =   0;
S   =   0;
RD  =   0;
F   =   0;
HFP =   3;
HS  =   1;
HBP =   3;
AW  =  35;
VFP =   1;
VS  =  31;
VBP =   2;
AL  = 275;
```

2.8 VSYNC (6FH) 2.8

This command is used to set the GDC on the board as the master sync generator. Figure 7 shows the VSYNC command.

```
 ---------------------------------
| 0 | 1 | 1 | 0 | 1 | 1 | 1 | 1 |
 ---------------------------------
```

Fig. 7. VSYNC Command

2.9 SYNC (OFH or OEH) 2.9

The SYNC command is used to enable (OF) or blank (OE) the display. This command is also used to control the video display with the same parameters allowed as that for the Reset command. The difference being that this command does not reset the GDC nor does it place the GDC in an idle state. This command and associated parameters are shown in figure 8. Figure 6 defines each of the associated parameter bytes.

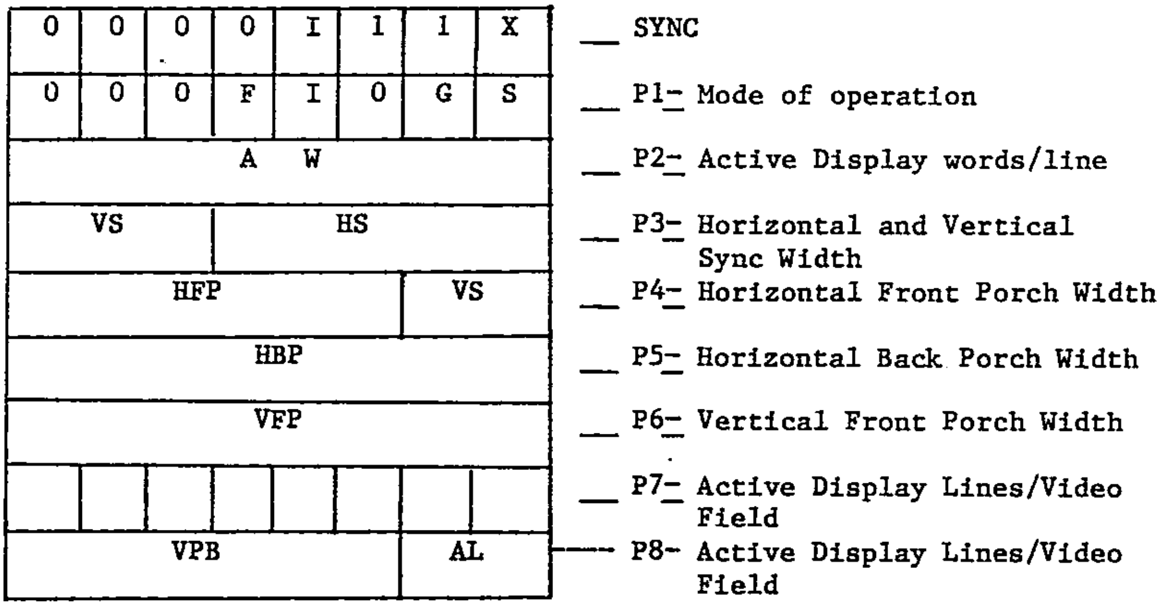

0 0 0 0 I I I X
SYNC
0 0 0 F I 0 G S
P1- Mode of operation
A W
P2- Active Display words/line
VS HS
P3- Horizontal and Vertical Sync Width
HFP VS
P4- Horizontal Front Porch Width
HBP
P5- Horizontal Back Porch Width
VFP
P6- Vertical Front Porch Width
P7- Active Display Lines/Video Field
VPB AL
P8- Active Display Lines/Video Field

Fig. 8. SYNC Command and Associated Parameter Bytes.

2.10 Display Control Commands 2.10

There are six Display Control commands which are used to control the manner in which an image drawn in the display memory is accessed and displayed on the attached screen of the monitor. These six Display Control commands are as follows:

2.11 PITCH (47H) 2.11

This command specifies the width of the display memory in the horizontal dimension. It sets the number of words between vertically adjacent pixels. If the bit map and the display window are the same size, the pitch should be the same as the Active Words per line parameter programmed as part of the Reset or SYNC command (a horizontal resolution of 256 dots results in a pitch of 16) or 256/16 = 16. Note that the value entered for the pitch paramter is not decremented by one as in the parameter value entered for the horizontal resolution parameter for the Reset command. If a reset command is executed and parameter 2 of the Reset command is sent, the Pitch command must be executed and if parameter 2 of the Reset command is sent, the Pitch command must be executed to properly reload the Pitch register within the GDC.

By specifying a pitch parameter which is greater than the Active Words per line, a bit map can be created that is wider than the display surface. Changing the starting address of the display (with the PRAM command) allows the display window to be panned across the bit map. If a wider bit map is used, care must be taken to avoid drawing figures which exceed the available memory or inadvertantly cross from one color plane to another. Figure 9 shows the PITCH command and associated parameter byte. Note that the Active Word (AW) parameter value is two less than the display window width. The Pitch command must be used to set the proper memory width larger than the window width.

0	1	0	0	0	1	1	1	-- PITCH (47H)
								---P1---Number of word addresses of the display memory in the horizontal direction.

Fig. 9. PITCH Command and Associated Parameter Byte

2.12 ZOOM (46H) 2.12

This command is used to specify the multiplier factors used during display and characted drawing and rectangle filling.

The board does not support display zooming; therefore, the upper four bits (nibble) of the parameter must be zero. The zoom magnification factor for a graphic character is set by the lower nibble in integer multiples. A value of 0 sets the multiplication factor to 1; a value of 15 sets the multiplication factor to 16. Figure 10 shows the Zoom command and associated parameter byte.

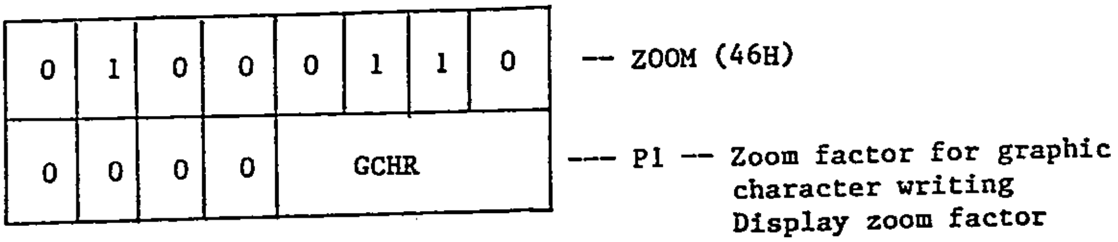


Fig. 10. ZOOM Command and Associated Parameter Byte

2.13 ' PRAM (7XH) 2.13

This command is used for loading the internal 16-byte RAM area of the GDC chip. The lower four bits of this command determine the starting address in the parameter RAM area. The succeeding bytes are stored in predefined locations. The host processor can write into any location of the PRAM and load up to 16 bytes into the parameter RAM at incrementing addresses. The parameter RAM area within the GDC chip is used to store two types of information. It is used to specify the details of the display area partitions, or used to supply the bit pattern for either figure drawing or the graphic character. The sequence of the byte transfers is terminated by the next command byte entered into the FIFO. Figure 11 · shows the PRAM command and associated parameter bytes. Figure 12 defines the parameter bytes associated with the PRAM command. This command may be used to modify the contents of any byte or sequence of bytes starting with the byte specified within the least significant four bits of the command byte. The starting address is used in panning and scrolling. To pan, your PITCH must be greater than the active words per line. To scroll, your starting address is incremented in multiples of your PITCH setting. In color mode, scrolling tends to shift your displayed screen and therefore is not useful.

<table>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td colspan="4">SA</td><td>-- PRAM (7XH)</td></tr>
<tr><td colspan="8">SAD1L</td><td>- P1</td></tr>
<tr><td colspan="8">SAD1H</td><td>- P2</td></tr>
<tr><td colspan="4">LEN1L</td><td>0</td><td>0</td><td colspan="2">SAD1H</td><td>- P3</td></tr>
<tr><td>0</td><td>0</td><td colspan="6">LEN1H</td><td>- P4</td></tr>
<tr><td colspan="8">SAD2L</td><td>- P5</td></tr>
<tr><td colspan="8">SAD2H</td><td>- P6</td></tr>
<tr><td colspan="4">LEN2L</td><td>0</td><td>0</td><td colspan="2">SAD2H</td><td>- P7</td></tr>
<tr><td></td><td>X</td><td colspan="6">LEN2H</td><td>- P8</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P9</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P10</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P11</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P12</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P13</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P14</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>-P15</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>- P16</td></tr>
</table>

Fig. 11. PRAM command and Associated Parameter Bytes

Fig. 12. Definition of PRAM Parameter Bytes

Byte	Function
	The command byte specifies the starting address in the parameter RAM. From the starting address, any number of bytes may be loaded into the parameter RAM at incrementing addresses up to 15.
1 - 4	The first 4 bytes specify the starting address and the lenght of the top portion of the first display area. The starting address uses an 18-bit value, but the board only uses the least significant 14 bits. Normally, a starting address of 0 is used. The length portion of the parameter RAM block is a 10-bit value used to specify the length (in words) of the first (top) display window. When only one display window is used, the length should be set to its maximum value (3FFH). Note that the top two bits of the fourth byte must be 0.
5 - 7	If two display windows are used, these bytes specify the starting address and the length of the bottom display area (or second window) following the same guidelines as listed above.
8 - 9	Bytes 8 through 9 specify the 16 bit pattern information when in the bit-mapped graphics mode. When drawing a line, arc, or rectangle (linear figures), bytes 8 and 9 are used to specify the line style (dotted or dashed).
8 - 15	Bytes 8 through 15 are used to specify the character front bit pattern when drawing a graphics character. A value of 1 enables the corresponding pixel, while a value of 0 suppresses it. Again, a value of 1 in a given bit position enables drawing, while a value of 0 suppresses it.

2.14 CURS (49H)

This command sets the position of the graphics cursor by specifying the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word. The memory address is an 18-bit value, while the dot address within the word is a 4-bit value. This results in a total of 22 bits of pixel address.
Figure 13 shows the CURS command and associated parameter bytes.

0	1	0	0	1	0	0	1	— CURS (49H)
EAD								--- P1--Execute Word Address low byte
0	0	EAD						--- P2--Execute Word Address high byte
dAD				0	0	0	0	--- P3--Graphics Mode only

Word Address - top two bits
Dot Address within the word

Fig. 13. CURS Command and Associated Parameter Bytes

Normally, graphic coordinates are specified by the X and Y coordinates of the Cartesian coordinate system with the origin (X = 0, and Y = 0) at the bottom left corner of the screen. The device driver software must convert these X and y values to an absolute memory address. Note that the memory address starts at the top of the display and increments towards the bottom of the screen, while the Y coordinate starts at the bottom and increases toward the top of the screen. This direction reversal means the Y coordinates must be inverted before the memory address is calculated.

In graphics mode, the cursor word address specifies the word containing the starting pixel of the drawing; the dot address value specifies the pixel within that word. The following equations show the calculation for the absolute memory address.

x = x coordinate. It must be in the range 0<=X <=XRES
Y = Y coordinate. It must be in the range 0<=Y <=YRES

Where: XRES = horizontal resolution in dots
YRES = vertical resolution in dots
Y^{-} = inverted Y coordinate
MA = memory address

Y^{-} = (YRES - 1) -Y
MA = Y^{-} * XRES + X

If color mode is used, the base address of the color plane must be added

RED = 0
BLUE = 1000H words, 10000H bits
GREEN = 2000H words, 20000H bits

Once the 22-bit memory address is calculated, it must be formated by the driver software into the form required by the CURS command.

Memory Address (MA) = ZAABBC

B	B	- Parameter 1 EAD
0 0 A*	A	- Parameter 2 EAD
C	0 0 0 0	- Parameter 3 dAD

A, B and C are 4-bit nibbles
* set the upper two bits of this nibble to 0.

2.15 START (6BH) 2.15

This command starts the display process with the previously defined display conditions. Figure 14 shows the START command. This command is not used in the Black/White mode, because hardware logic external to the GPC take over the display process.

```
---------------------------------
| 0 | 1 | 1 | 0 | 1 | 0 | 1 | 1 |     --start (6bh)
---------------------------------
```

Fig. 14. START Command

2.16 BCTRL (OCH, ODH) 2.16

The blanking control command (ODH) enables and (OCH) disables the display proces, blanking the screen. Figure 15 shows the BCTRL command.

```
---------------------------------
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | X |    --BCTRL (OCH,ODH)
---------------------------------
```

Fig. 15. BCTRL Command

2.17 Drawing Control Commands 2.17

The Drawing Control commands are used to perform bitmap manipulation and to draw figures. These commmand are used repeatedly as figures are drawn.

2.18 MASK (4AH) 2.18

The Mask command loads the 16-bit Mask register which specifies the bit or bits within the currently addressed display memory that are to be modified. The Mask register can be modified using the CURS command to reposition the cursor or modify the drawing. Two parameters associated with the Mask command are used to load a 16-bit value into the Mask register. If normal single pixel at a time graphics figure drawing is desired, there is no need to use this command because the CURS command sets up the proper pattern to address the proper pixels as the drawing progresses. The Mask register should be set to all ones for any word-at-a-time operation. Fiqure 16 shows the Mask and associated parameter bytes. Figure 17 defines the parameter bytes associated with the Mask command.

0	1	0	0	1	0	1	0	--MASK (4AH)
ML								---P1--Low significant byte
MH								---P2--High significant byte

Fig. 16. MASK Command and Associated Parameter Bytes

Byte	Function
1	The value of 1 for each bit defines the bits of the least significant byte in display memory that can be modified.
2	The value of 1 for each bit defines the bits of the most significant byte in display that can be modified.

Fig. 17. Definition of MASK Command Bytes.

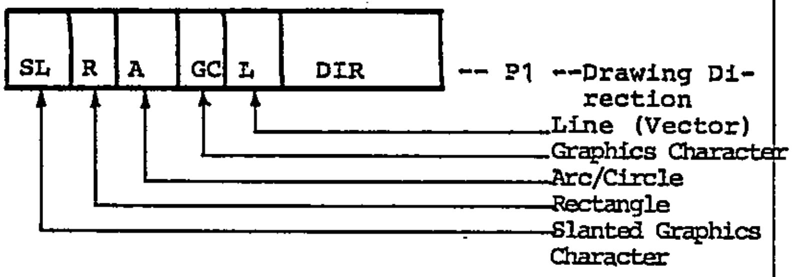
2.19 FIGS (4CH) 2.19

The FIGS command specifies the geometric figure or graphic character to be drawn and is used to set up the parameters needed to modify the bit map. Up to 11 parameters may follow this command. To prepare for graphics drawing, the GDC requires the figure type, the direction, and drawing parameters, the starting pixel address, and the pattern. Once these are stored within the GDC, the Figure Draw (FIGD) command initiates the drawing operation. Figure 18 shows the FIGS command and associated parameter bytes. Figure 19 defines the parameter bytes associated with the FIGS command.

<table>
<tr><td>0</td><td>1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>--FIGS (4CH)</td></tr>
<tr><td>SL</td><td>R</td><td>A</td><td>GC</td><td>L</td><td colspan="3">DIR</td><td>-- P1 Drawing Direction</td></tr>
<tr><td colspan="8">DC</td><td>-- P2 DC Drawing Character</td></tr>
<tr><td>0</td><td>GD</td><td colspan="6">DC</td><td>-- P3 DC Drawing Character (high bits)</td></tr>
<tr><td colspan="8">DL</td><td>-- P4 D Drawing Parameter</td></tr>
<tr><td>0</td><td>0</td><td colspan="6">DH</td><td>-- P5 D Drawing Parameter</td></tr>
<tr><td colspan="8">D2L</td><td>-- P6 D2 Drawing Parameter (Low byte)</td></tr>
<tr><td>0</td><td>0</td><td colspan="6">D2H</td><td>-- P7 D2 Drawing Parameter (High byte)</td></tr>
<tr><td colspan="8">D1L</td><td>-- P8 D1 Drawing Parameter (Low byte)</td></tr>
<tr><td>0</td><td>0</td><td colspan="6">D1H</td><td>-- P9 D1 Drawing Parameter (High byte)</td></tr>
<tr><td colspan="8">DML</td><td>-- P10 DM Drawing Parameter (Low byte)</td></tr>
<tr><td>0</td><td>0</td><td colspan="6">DMH</td><td>-- P11 DM Drawing Parameter (High byte)</td></tr>
</table>

Fig. 18. FIGS Command and Associated Parameter Bytes

Fig. 19. Definition of FIGS Parameter Bytes

Byte	Function
1	The three least significant bits of this byte specify the direction that the figure is drawn, while the next 5 bits select the drawing type. The GDC uses the successful approximation technique to find the next pixel of the figure. This pixel is one of the eight nearest neighbors to the last pixel drawn. The GDC assigns each of these eight directions a number from 0 to 7 as shown in Figure 20. Figure 21 shows the result of the direction parameter on the various figure types.  SL R A GC L DIR -- P1 --Drawing Direction Line (Vector) Graphics Character Arc/Circle Rectangle Slanted Graphics Character
2 and 3	These two bytes specify the Drawing Count (DC) parameter. The value specified should be one less than the number fo RMW cycles to be executed. Byte 2 defines the least significant address; byte 3 defines the most significant address.
4 and 5	These two bytes specify the D' parameter which is the number of words to be accessed in the direction specified.
6 and 7	These two bytes specify the D2 parameter which takes on different meanings depending on the type of figures to be drawn. Note that the upper two bits of these bytes are zero. Figure 22 defines each of the parameters for a given drawing type. When drawing a line with negative slope, you must swap the X and Y coordinates so that the absolute value of delta X (dX) is equal to or greater than the absolute value of delta Y (dY). The direction that the line is drawn depends on the polarity (sign) of three values: delta X (dX), delta Y (dY) and the difference of absolute value of dX minus the absolute value of dY. Figure 23 shows the relationships.
8 and 9	These two parameter bytes specify the D1 drawing parameter and take on a different meaning depending on the figure to be drawn.
10 and 11	These two parameter bytes specify the DM drawing parameter which is the dots to be masked from the figure to be drawn.

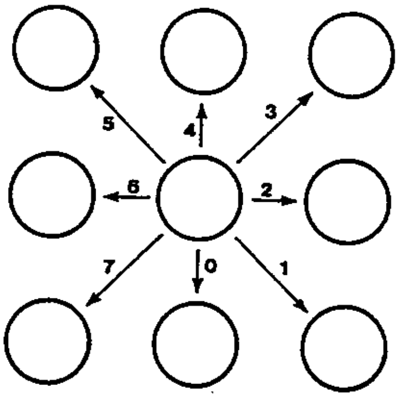


Fig. 20. Drawing Direction

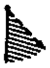
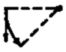
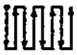
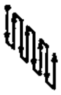
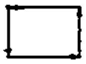
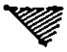
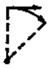
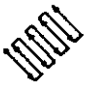
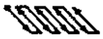
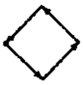
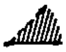
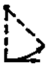
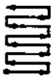
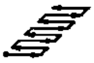
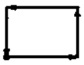
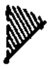
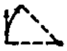
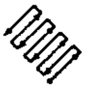
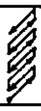
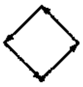
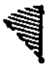
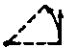
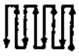
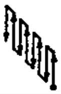
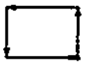
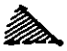
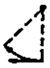
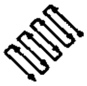
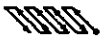
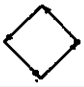
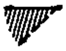
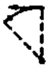
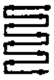
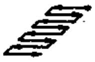
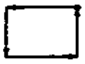
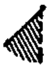
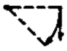
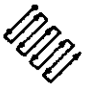
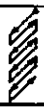
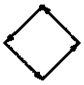
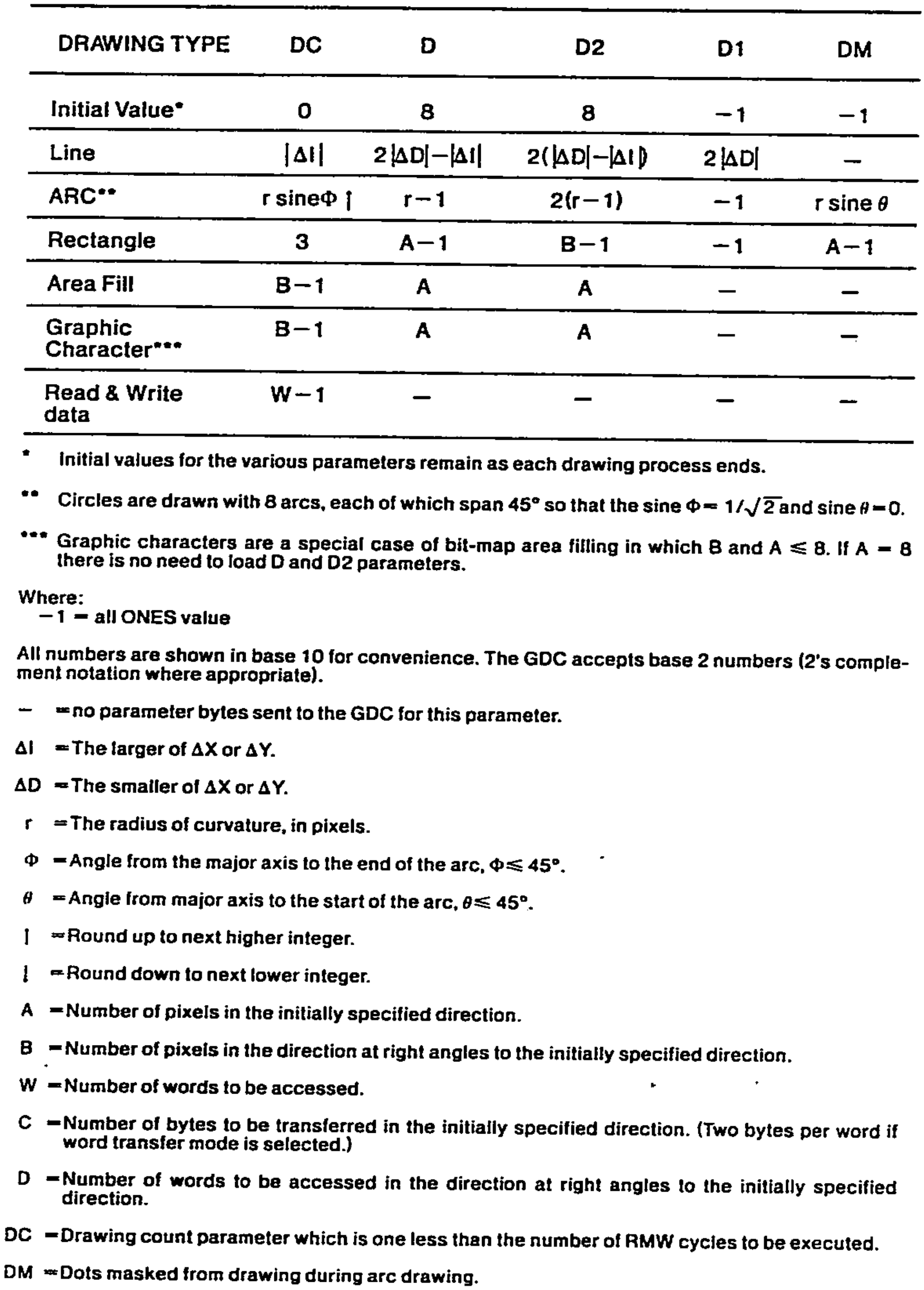
DIR	LINE	ARC	CHARACTER	SLANT CHAR	RECTANGLE
000					
001					
010					
011					
100					
101					
110					
111					

Fig. 21. Drawing Direction Resulting from Direction Parameter



DRAWING TYPE	DC	D	D2	D1	DM
Initial Value*	0	8	8	-1	-1
Line	$\lvert\Delta I\rvert$	$2\lvert\Delta D\rvert-\lvert\Delta I\rvert$	$2(\lvert\Delta D\rvert-\lvert\Delta I\rvert)$	$2\lvert\Delta D\rvert$	—
ARC**	r sineΦ ⌈	$r-1$	$2(r-1)$	-1	r sine θ
Rectangle	3	$A-1$	$B-1$	-1	$A-1$
Area Fill	$B-1$	A	A	—	—
Graphic Character***	$B-1$	A	A	—	—
Read & Write data	$W-1$	—	—	—	—

* Initial values for the various parameters remain as each drawing process ends.

** Circles are drawn with 8 arcs, each of which span 45° so that the sine $\Phi = 1/\sqrt{2}$ and sine $\theta = 0$.

*** Graphic characters are a special case of bit-map area filling in which B and $A \leq 8$. If $A = 8$ there is no need to load D and D2 parameters.

Where:
-1 = all ONES value

All numbers are shown in base 10 for convenience. The GDC accepts base 2 numbers (2's complement notation where appropriate).

— =no parameter bytes sent to the GDC for this parameter.

ΔI =The larger of ΔX or ΔY.

ΔD =The smaller of ΔX or ΔY.

r =The radius of curvature, in pixels.

Φ =Angle from the major axis to the end of the arc, $\Phi \leq 45°$.

θ =Angle from major axis to the start of the arc, $\theta \leq 45°$.

⌈ =Round up to next higher integer.

⌊ =Round down to next lower integer.

A =Number of pixels in the initially specified direction.

B =Number of pixels in the direction at right angles to the initially specified direction.

W =Number of words to be accessed.

C =Number of bytes to be transferred in the initially specified direction. (Two bytes per word if word transfer mode is selected.)

D =Number of words to be accessed in the direction at right angles to the initially specified direction.

DC =Drawing count parameter which is one less than the number of RMW cycles to be executed.

DM =Dots masked from drawing during arc drawing.

Fig. 22. Graphics Fiqure Drawing Paraneters

dX	dY	dX - dY	Direction	L	D
+	+	+	2	X	Y
+	+	-	3	Y	X
+	-	+	1	X	Y
+	-	-	0	Y	X
-	+	+	5	X	Y
-	+	-	4	Y	X
-	-	+	6	X	Y
-	-	-	7	Y	X

Fig. 23 Relationship of Line Drawing Direction

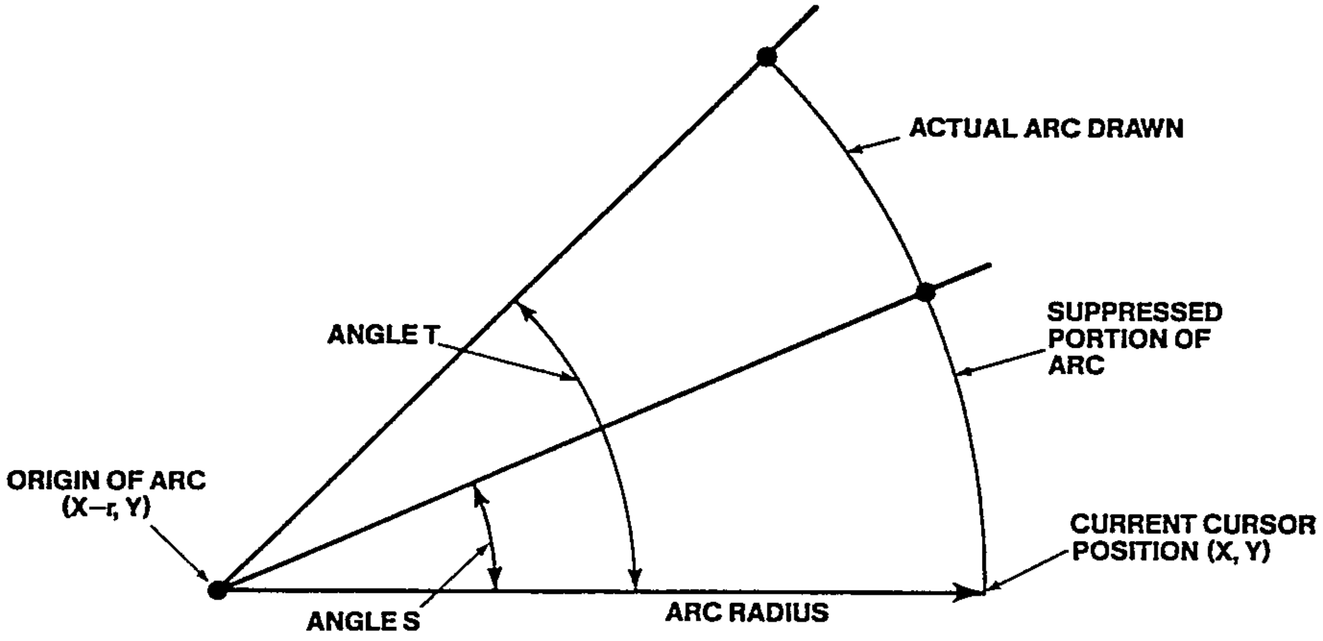
When drawing an arc, several points must be kept in mind. First, the maximum arc angle is 45 degree, starting on a coordinate axis intercept. The current cursor position is used as the coordinate intercept point, and the contents specified by the DIR parameter indicate the direction that the arc is drawn. Two angles are used to specify the arc. Figure 24 shows an example of drawing an arc with a DIR of 4. Angle "T" specifies the total angle from the the coordinate axis of the arc (up to 45 degree). Angle "S" specifies the angle from the coordinate axis of the suppressed (not displayed) portion of the arc. It also shows the drawing parameters definition.

A circle is drawn as eight 45 degree arcs, each starting at a coordinate axis intersection, the origin being at the center of the circle. If (X,Y) is the position at the center of the arc, the starting points are: X+r, Y (right edge), X-r, Y (left edge), X, Y + r (top edge), and X, Y-r (bottom edge). Two arcs are drawn in opposite directions beginning at the starting point.

When drawing a rectangle, the length and height parameters (D and D2) specify the number of pixels on each side of the box, not including the starting point. Figure 25 shows a box drawn with a starting point of (0,0), a lenght of 4 (D = lenght minus one or 3) and a height of 6 (D2 = height minus one or 5). The result would actually be a box 5 pixels by 7 pixels.

When filling a rectanglar area, you must change the parameter definition. Parameters D and D2 are set to length + 1, while the parameter DC is set to the height. For example, if the rectangle shown in Figure 25 were completely filled (including the borders), the cursor would be positioned at (0,0), parameters D and D2 would be set to 5 (length + 1), and parameter DC would be set to 6 (height).

If only the interior of the rectangle is to be filled (assuming the borders were previously drawn and are to be left intact), the cursor would be positioned at (1,1), parameters D and D2 would be set at 3 ((length +1)) -2). Two must be substracted to compensate for the left and right borders, and parameter DC wc ıld be set to 4 (height minus 2). Two is subtracted to compensate for the top and bottom borders.



PARAMETER	CONTENTS	
DC	r * sin (T)	ROUNDED UP
D	r – 1	
D2	2 * (r–1)	
D1	-1	
DM	r * sin (S)	ROUNDED DOWN

NOTE: ANGLE T MUST BE GREATER THAN 0 AND LESS THAN OR EQUAL TO 45°
ANGLE S MUST BE GREATER THAN OR EQUAL TO 0 AND LESS THAN 45°

Fig. 24. ARC Drawing

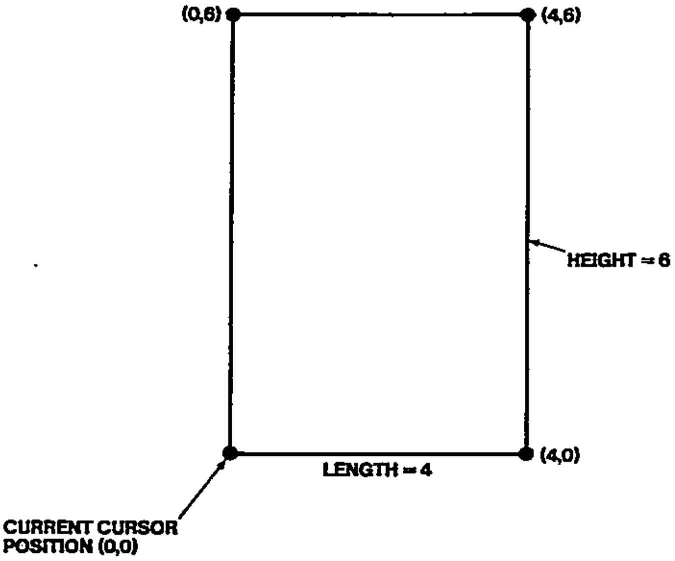


Fig. 25. Rectangle Drawing

All figures are drawn from the current cursor position. Typically, a CURS command is qiven prior to each figure drawing (FIGD) command to position the cursor at the starting point of the fiqure. The drawinq parameters loaded into the GDC are modified as drawinq progresses; therefore, they must be reloaded when another fiqure is to be drawn, even if it is the same fiqure.

2.20 FIGD (6CH) 2.20

The FIGD command loads the previously loaded drawinq parameters from the Parameter RAM into the drawinq processor and starts the drawing process at the pixel address pointed to by the cursor. Fiqure 26 shows the FIGD command.

```
---------------------------------
| 0 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |   FIGD (6CH)
---------------------------------
```

Fig. 26. FIGD Command

2.21 GCHRD (68H) 2.21

The GCHRD command is used to start drawing a graphics character or begin filling an area with the pattern stored in the Parameter RAM. The parameters are loaded with the FIGS command. Drawing begins at the address in display memory pointed to by the EAD and dAD values. Prior to executing this command, you must load the Drawing Parameter Registers with the appropriate values and load the parameter RAM with the character font or fill pattern. Note that the loadinq of the Parameter RAM changes the line drawing pattern for later drawings; hence, you must remember to reload the line pattern before doing another figure drawing. Figure 27 shows the GCHRD command.

```
---------------------------------
| 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |   --GCGRD (68H)
---------------------------------
```

Fig. 27. GCHRD Command

2.22 WDAT (base code = 20H) 2.22

The WDAT command initiates the RMW activity and writes data directly to the display memory. This command requires parameters to set the pattern register within the GDC while other commands use the stored value in the parameter RAM. The WDAT command must be preceeded by a FIGS command and its associated parameters. The first three parameters of the FIGS command are needed to set the type of drawing to take place, the direction of the drawing, and the DC value. The DC value of the DC parameter plus one is the number of RMW cycles performed by the GDC. The one RMW cycle is executed at the address pointed to be the cursor address. The EAD pointer is incremented to the next word as specified by the direction parameter. Additional parameters can then be accepted. For byte writes, the unspecified byte is treated as all zeros during the RMW memory cycle. Only the LSB of the parameter byte is used as the pattern during the RMW memory cycle. In addition, it may be used to set the type of drawing with other commands.

Three types of bit map modifications can take place; the lower two bits of this command specify the type. Once set, this modification mode is retained until another WDAT command is given. The WDAT command is given without parameters if setting the modification mode is all that is required.

If it is desired to write data directly to the bit map, the GDC must be placed in the mixed character/graphic mode. The cursor must then be positioned to the correct place and the mask register loaded with all ones (OFFFFH). The Direction and DC parameters must then be loaded with the desired direction code and a value equal to one less than the number of words to be written. A direction of 2 is typically used. The WDAT command is then given one set of parameters (either one or two bytes, depending on whether bytes or words are to be written). The GDC repeatedly writes the same data pattern to memory the number of times specified by the DC parameter.

While the GDC is writing to the display memory, the "Drawing-IN-Progress" bit is not set. In order to determine when the WDAT command is completed, send the GDC another command (such as Display Start) and check the FIFO Empty status bit. When the value for this bit goes to a 1, both commands are complete.

Remember, the modification mode set by the last WDAT command is used for all subsequent drawings. Finally, the GDC should be placed back into the graphics mode for normal operation. Figure 28 shows the WDAT command and associated parameter bytes. Figure 29 lists the modes and command codes.

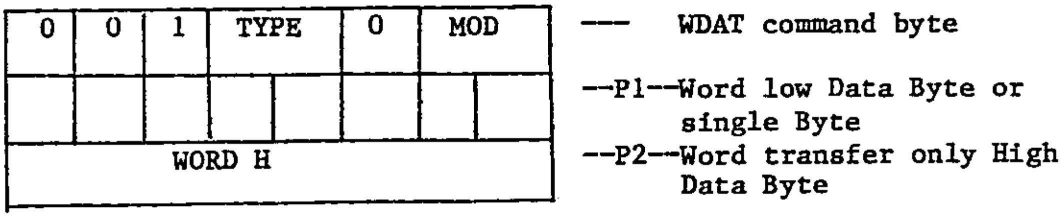

0
0
1
TYPE
0
MOD
WORD H
--- WDAT command byte
--P1--Word low Data Byte or single Byte
--P2--Word transfer only High Data Byte

Fig. 28. WDAT Command and Associated Parameter Bytes

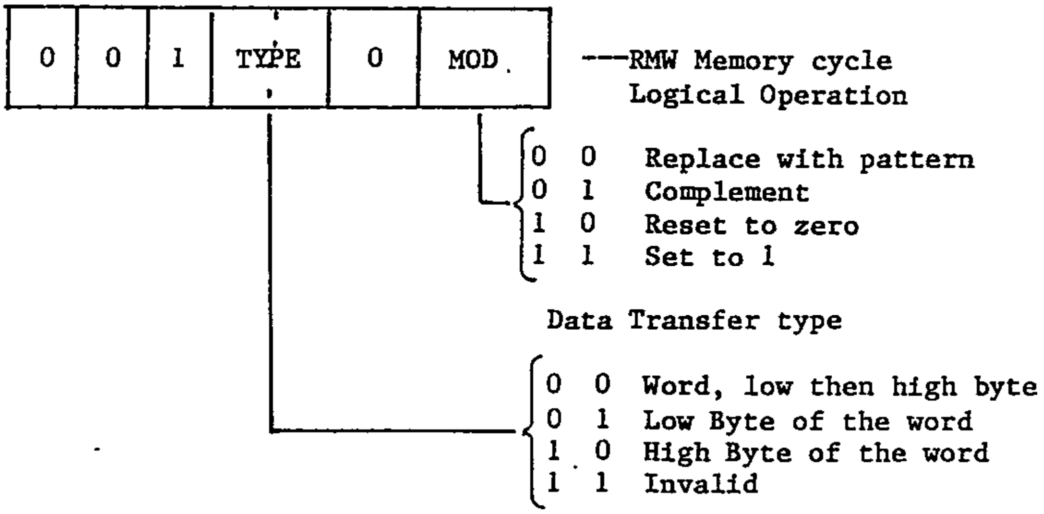
Byte	Function
	The three least significant bits of this command byte specify the 
1	This parameter byte is either the least significant byte of a word transfer or the single data byte.
2	This parameter byte is the most significant byte of a word transfer.

Fig. 29. Definition of WDAT Parameter Bytes

2.23 Data Read Commands 2.23

The Data Read commands allow the host processor to recover data from the GDC. This data is in addition to the status information always available via the status input port. After a Data Read command is given, the GDC˝s FIFO is cleared and is configured as an output buffer. Data is then passed back to the host processor with an input instruction. The presence of data is indicated by the setting of the Data Ready bit. A Data Read command may be aborted before all of the data is read by writing a new command to the GDC (ignore the FIFO Full Status bit). This terminates the Data Read command, clears the FIFO, and reverses the direction of data.

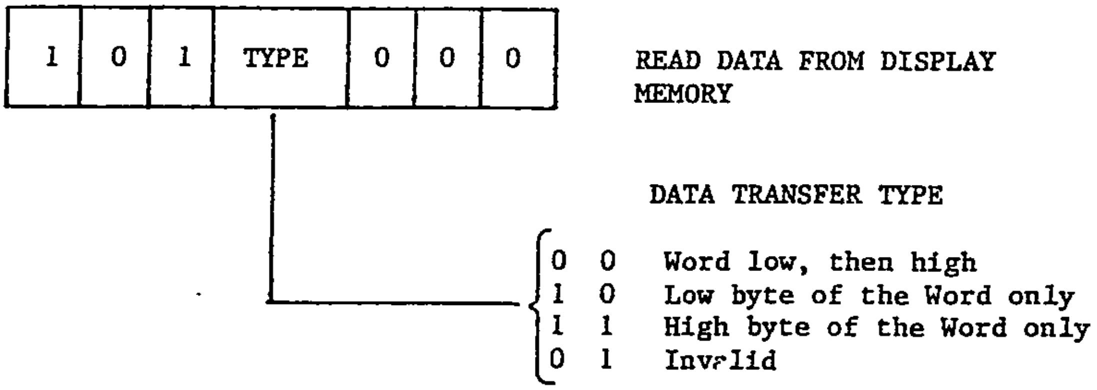
The Read Data Commands consists of RDAT, CURD, and LPRD.

2.24 RDAT (Base Code = 0A0H) 2.24

The RDAT command reads data directly from the bit map display memory. Either bytes or words can be read. The sequence to us is as follows:

First, the cursor must be positioned to the correct location. Next, the mask register should be loaded with all ones (FFFFH). The direction and DC parameters should then be loaded using the same format as that used with the WDAT command with one exception. The exception being that the DC value is the actual number of either bytes or words to be read (not one less). The RDAT command can then be loaded. The data is then read a byte at a time from the data port.

The two least significant bits of the RDAT command byte also set the modification mode in the same manner as that used with the WDAT command. If the code "OAOH is used, the mode is set for a replace operation. Make sure that the correct mode is reloaded before other operations are started. Figure 30 shows the RDAT command.



1	0	1	TYPE	0	0	0

READ DATA FROM DISPLAY MEMORY

DATA TRANSFER TYPE

0	0	Word low, then high
1	0	Low byte of the Word only
1	1	High byte of the Word only
0	1	Invalid

Fig. 30. RDAT Command

2.25 CURD (0E0H) 2.25

The CURD command reads the current cursor position. The first three bytes read are the 18 bitss of absolute memory address (only the least significant 14 bits are used). The next two bytes are the 16-bit contents of the Mask Register. If the mask register was not directly loaded, the returned contents of the Mask register will be the one of 16 codes for the dot address. A value of 1 in the least significant bit corresponds to a dot address of 0; a value of 1 in the most significant bit corresponds to a dot address of 0FH. Figure 31 shows the CURD command and the bytes returned.

224

1	1	1	0	0	0	0	0	--CURD (E0H)
EADL								--P1--Execute Address (EAD) Low Byte
EADH								--P2--Execute Address (EAD) High Byte
0	0	0	0	0	0	EADH		--P3--Execute Address (EAD) High bits
dADL								--P4--Dot Address (dAD); Low Byte
dADH								--P5--Dot Address (dAD); High Byte

Fig. 31. CURD Command and Returned Bytes

2.26 LPRD (0C0H) 2.26

The LPRD command reads the address of a light pen "hit" after indication is posted in the status register. For a light pen to register, two consecutive "hits" at the same address must be detected. The address returned reflects a word level (not a dot level); thus, the returned address is 18 bits (3 bytes) in length. There is a position error of several words due to internal and external delays. Figure 32 shows the LPRD command and returned bytes. VPB701 is not supplied with light pen connection.

1	1	1	0	0	0	0	0	—CURD (E0H)
LADL								—P1—Light Pen Address; Low Byte
LADH								—P2—Light Pen Address; High Byte
0	0	0	0	0	0	LADH		—P3—Dot Address (dAD); High Bits

Fig. 32. LPRD Command and Returned Parameter Bytes

2.27 Operation Sequence 2.27

The following sequences are recommended. Following these sequences ensures that all registers are in the proper state.

After power up, the first operation should be the initialization operation. Once initialized, the GDC may be given an indefinite number of drawing and display commands without further operator intervention.

2.28 Initialization 2.28

The following operation should be performed to prepare the VPB701 board for operation. Two key variables are used repeatedly and should be initialized here. These variables are the X resolution in dots (XRES) and the Y resolution in dots (YRES). The term "AR" refers to values selected "As Required" for the display monitor and for the particular system in use.

1. Issue Reset
   - a. Operation Mode = AR
   - b. Active Words/Line = (XRES/16) - 2
   - c. Horizontal Sync = AR
   - d. Vertical Sync = AR
   - e. Horizontal Front Porch = AR
   - f. Horizontal Back Porch = AR
   - g. Vertical Front Porch = AR
   - h. Active Lines/Frame = YRES or (YRES - 1)/2 for interlace operation)
   - i. Vertical Back Porch = AR

2. Set Pitch = XRES/16

3. Set Zoom = 0

4. Set Cursor Characteristics
   - a. Lines/Row = 1
   - b. Disable cursor

5. Set Starting Address in PRAM
   - a. Address = 0
   - b. Block Length = 3FFH

6. Set Vertical Sync Mode to Master

7. Clear, Screen with WDAT command
   - a. Set Mixed Character/Graphics Mode
   - b. Set cursor address to 0
   - c. Set Mask to 0FFFFH
   - d. Set Direction to 2, set DC to 3FFFH
   - e. Issue WDAT command with word operation, replace pattern (code 20H). Data =00,00
   - f. Set graphics mode

8. Begin Display

2.29 Symbol Drawing 2.29

The following steps should be followed when drawing characters or symbols to the screen:

1. Store character or symbol bit pattern into the parameter RAM.

2. Set modification mode.

3. Set Zoom as required.

4. Position cursor as required.
5. Set up drawing parameters.
   a. Select graphics character drawing.
   b. Set Direction as reqired.
   c. Set DC to 7.

6. Issue Draw Graphics Character command.

7. To repeat (as for multiple color planes), go back to step 4.

2.30 Figure Drawing 2.30

1. Set line pattern in bytes 8 and 9 of the parameter RAM.

2. Set modification mode.

3. Position cursor as required.

4. Set up drawing parameters.
   a. Select appropriate figure type and direction.
   b. Load parameters DC, D, D2, D1 and DM as required.

5. Issue Draw Figure command.

6. To repeat, go back to step 3.

2.31 Rectangle Filling 2.31

The following steps shold be followed when filling rectangular areas.

1. Store fill pattern into parameter RAM.

2. Set modification mode.

3. Set Zoom to 0.

4. Position cursor as required.

5. Set up Drawing parameters.
   a. Select Graphics Character Drawing.
   b. Set Direction as required.
   c. Set DC to height.
   d. Set D to D2 to length + 1

6. Issue Draw Graphics Character command.

7. To repeat, go back to step 4.

2.32 Example 2.32

In this example, the calculations are derived from a particular monitor. If you attempt to do this example, your calculations might be different depending on the type of monitor you are using. Using this as an example of how to derive the values for the different parameters, an arc will be created in display memory and displayed on the screen of your monitor.

Using the formulas given in paragraph 2-7, calculate the initialization parameters associated with the Reset command for your specific application. Send the Reset command and associated parameters required to initialize for calculation purposes in this example, lets use a horizontal resolution of 12H (18 words) and a vertical resolution of D8H (216 lines). We now have to calculate the EAD. Assume that we want to display a 45 degree arc starting at an X coordinate of 45H and at a Y coordinate of 50H on the screen with a radius of 225 bits from the center of the arc. Therefore,

EAD = (Y MAX - Y).(xres) words) + (x words)

EAD = (D8 -50)(12H) + (4)

EAD = (88)(12) + 4

EAD = 990 + 4

EAD = 994

dAD = 5

Now that the EAD and dAD are calculated, lets enter these values as the parameter bytes associated with the CURS command.

- Output the byte 49H (CURS command) to the command port of the GDC.

- Output the byte 94H as the value for the least significant byte of the cursor position (as calculated) to the data port of the GDC.

- Output the byte 09 as the value for the most significant byte of the cursor position (as calculated) to the data port of the GDC.

- Output the byte 50 as the value for the top address bits of the cursor position and as the dot address.

Now lets use the FIGS command to set up the drawing parameters for a 45 degree arc with a radius of 255 bits.

- Output the byte 4C (FIGS command) to the command port of the GDC.

- The code to specify Parameter 1 for drawing an arc with a DIR of 2 is 22H. Therefor, output the byte 22 to the data port of the GDC.

- The code to specify the DC parameter 2 for an arc specifies a value calculated by multiplying the radius times the sine of angle theta rounded up to the next even interger. The sine of a 45 degree angle is .707. Therefore, .707. x 255 = 180.28; rounded up gives 181. The hexadecimal value of 181 =B4H. Therefore, output the byte B4 to the data port of the GDC.

- Parameter 3 is the most significant byte of the DC parameter. Because B4 is the number of bits with which to draw the arc and because that value can be defined in the least significant byte, this byte should be set to 00.

- Parameter 4 is the D parameter. This parameter for an arc specifies r minus 1. The radius desired is 255; therefore, 255 minus 1 = 254. The hexadeciamal value for 254 = FEH. Therefore, output the byte FE to the data port of the GDC.

- Parameter 5 is the most significant byte of the D parameter. Because the value can be defined in the least significant byte, this byte should be set to 00. Therefore, output the byte 00 to the data port of the GDC.

- Parameter 6 is the D2 parameter. This parameter for an arc specifies 2(r - 1). The desired radius is 255. Thus, 2 (255 - 1) = 508. The hexadecimal value for 508 = 1FCH. Therefore, output the byte FC to the data port of the GDC.

- Parameter 7 is the most significant byte for the D2 parameter. Because the value for this byte exceeds FF, set this byte to 01.

- Parameter 8 is the D1 parameter byte and for an arc is equal to -1. Thus, as specified in figure 23, this byte is set to all ones or FFH. Therefore, output the byte FF to the data port of the GDC.

- Parameter 10 is the least significant byte of the DM parameter and for an arc is specified as the radius times the sine of angle theta rounded down. This parameter asks for the number of bits to be masked from the drawing. Because no bits are to be masked from the drawinq the value entered should be zero. Therefore, output the byte 00 to the data port of the GDC.

- The value for this byte is also zero for the same reason. Therefore, output the byte 00 to the data port of the GDC.

Now lets output the byte 6C (FIGD command) to the command port of the GDC to cause the GDC to draw the arc in memory and because of the initialization al-ready given, the drawing of the arc will appear on the screen at the coordinates given. Figure 33 shows the relative position of the arc displayed on the screen.

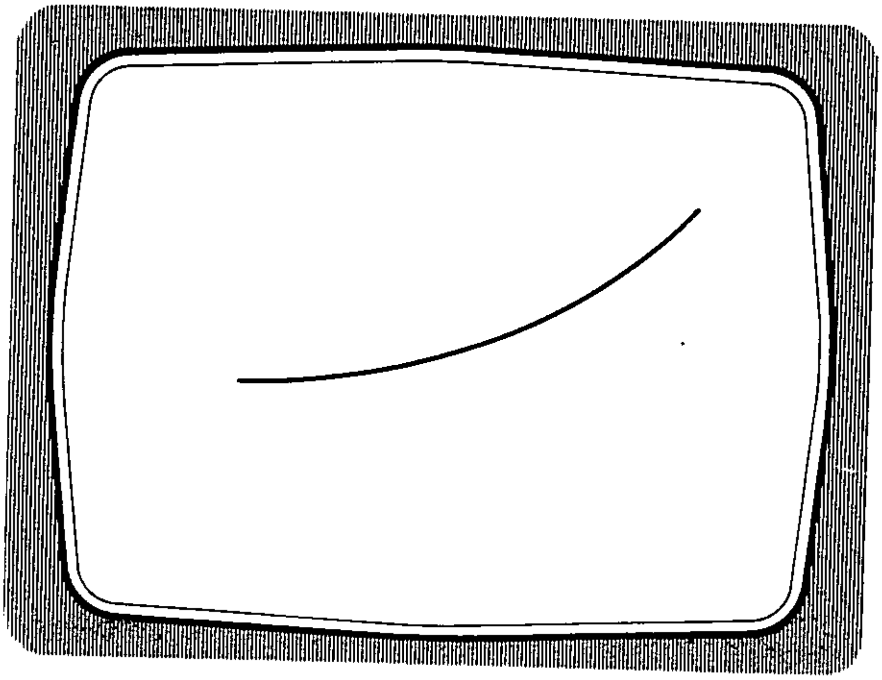


Fig. 33. Arc Displayed on the Monitor

A. INDICES

A.1 Survey of Figures

RETURN LETTER

Title: Programmers Reference Manual for VPB701 RCSL No.: 52-AA1224

A/S Regnecentralen af 1979/RC Computer A/S maintains a continual effort to improve the quality and usefulness of its publications. To do this effectively we need user feedback, your critical evaluation of this manual.

Please comment on this manual's completeness, accuracy, organization, usability, and readability:

____

____

____

____

Do you find errors in this manual? If so, specify by page.

____

____

____

____

How can this manual be improved?

____

____

____

____

Other comments?

____

____

____

____

____

Name: ____________________ Title: ____________________

Company: ____________________________________

Address: ____________________________________

Date: __________

Thank you