

Raster Technologies

Model One/80

PROGRAMMING GUIDE

Revision 2.0

October 11, 1985

Part # 552-00019-001

Raster Technologies, Inc.  
9 Executive Park Drive  
North Billerica, MA 01862  
(617) 667-8900



## MODEL ONE/80 PROGRAMMING GUIDE

The Model One/80 Programming Guide includes the following manuals:

### Model One/80 Modes of Operation

This manual describes ALPHA mode, GRAPHICS mode, and the Model One's special characters.

### Model One/80 Coordinates and Image Memory Addressing

This manual describes the coordinate origin, the clipping window, the screen origin, zooming, inhibiting screen refresh, the current point, the coordinate registers, and the crosshairs and hardware cursor.

### Model One/80 Pixel Values, Look-up Tables, and Image Memory

This manual describes the Model One/80 look-up tables, the value registers, the pixel processor, write and read masks, blinking colors, and the CLEAR and FLOOD commands.

### Model One/80 Graphics Primitives

This manual describes the Model One graphics primitives: points, lines, circles, arcs, rectangles, polygons, text, filled primitives, and area fills.

### Model One/80 Data Read-back and Image Transmission

This manual describes the read-back and image transmission commands.

### Model One/80 Macro Programming

This manual describes writing and execution of macro commands.

### Model One/80 Applications Development Features

This manual describes the REPLAY command, Command Stream Translator, and the integrated local debugger.

### Model One/80 Pixel Mover

This manual describes the Model One/80 pixel mover commands.

### Model One/80 Errors and Error Messages

This manual lists the errors and error messages.

### Model One/80 Command Reference

This manual provides a complete reference to the Model One/80 command set.



RASTER TECHNOLOGIES  
MODEL ONE/80  
MODES OF OPERATION

Revision 1.0 February 27, 1984



MODEL ONE MODES OF OPERATION  
February 27, 1984

Copyright 1984 by Raster Technologies, Inc. All rights reserved. No part of this work covered by the copyrights herein may be reproduced or copied in any form or by any means—electronic, graphic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without written permission.

NOTICE:

The information contained in this document is subject to change without notice.

RASTER TECHNOLOGIES DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS MATERIAL (INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESS OR IMPLIED. RASTER TECHNOLOGIES SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS MATERIAL.

This document contains proprietary information which is protected by copyright.

WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual may cause interference with regard to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.



Table of Contents

1.0	ALPHA Mode .....	4
2.0	GRAPHICS Mode .....	4
2.1	GRAPHICS Mode From the Local Terminal .....	5
2.2	GRAPHICS Mode From the Host Computer .....	6
2.3	The Model One's Input and Output Buffers .....	8
3.0	The Model One's Special Characters .....	9

Tables

Table 1	Default I/O Buffer Sizes .....	9
Table 2	The Model One's Special Characters .....	9



MODES OF OPERATION

The Model One functions in two modes: GRAPHICS mode, in which the Model One command interpreter decodes and executes graphics commands, and ALPHA mode, in which ASCII characters are passed through the Model One to connect the local alphanumeric terminal and the host computer.

1.0 ALPHA Mode

When the Model One is powered up, or if the RESET button mounted on the rear panel is pressed, a COLDstart command is executed. After COLDstart, the Model One is in ALPHA mode. You can then use the local alphanumeric terminal or keyboard to communicate directly with the host computer.

The Model One's COLDstart command, executed by pressing the RESET button or by entering the COLD command directly, performs a complete COLDstart on the Model One. The COLDstart includes clearing defined macros, coordinate and value registers, resetting the clipping window, and setting the Look-Up-Tables to the default. The Command Reference gives complete details of the COLDstart command. You should execute a COLDstart command after each example in this manual to reset the Model One.

The Model One supports three different host transmission formats: 8-bit binary, ASCII hexadecimal, and pure ASCII. The choice between 8-bit binary and ASCII hexadecimal is made with the SYSCFG command, as described in the Model One Installation Guide. ASCII hexadecimal format is used when the host computer cannot be programmed to transmit 8 bits of binary data over each character sent to the terminal.

Pure ASCII format allows the host computer to issue graphics commands in exactly the same format as commands typed at the local terminal. The ASCII flag command is used to set the host interface for pure ASCII format; all subsequent commands must be sent from the host to the Model One exactly as they would be typed in locally.

Pure ASCII format requires many more characters to be sent from the host to execute a series of commands and should be used only when the command stream must be directly interpreted by the programmer or user rather than the Model One.

2.0 GRAPHICS Mode

In GRAPHICS mode, the Model One command interpreter decodes and executes graphics commands coming from the local terminal or from the host computer. To enter GRAPHICS mode, a [CTRL-D] (04H or 84H) is sent from the host computer or from the local terminal. The [CTRL-D] must be sent as a 7-bit ASCII character, independent of whether the the Model One has been set to accept data in 8-bit binary or ASCII hex. The SPCHAR (see section 3) command can be used to change the ENTERGRAPHICS control character from a CTRL-D to any desired ASCII code. In fact, the SPCHAR command can be used to change any Model One special control character to any user-defined ASCII code. (The SAVCFG command will saves any changes made to special characters.)

Only one I/O port, either the port to the local terminal (ALPHASIO port) or one of the ports to the host computer (HOSTSIO, HOSTGPIB, or HOSTPIO), can be in GRAPHICS mode at any given time. When the ENTERGRAPHICS control character appears at one of these ports, the Model One enters GRAPHICS mode. Once the Model One is in GRAPHICS mode, the Model One expects graphics commands from the port which received the ENTERGRAPHICS control character, whether that is the host computer or the local terminal.

## 2.1 GRAPHICS Mode From the Local Terminal

When the ENTERGRAPHICS control character appears at the Model One's ALPHASIO port, the local terminal is put into GRAPHICS mode; the GRAPHICS prompt character ! is displayed at the terminal. Now, commands that you type are processed directly by the Model One and not sent to the host computer.

The format for graphics commands typed locally is:

```
COMMAND_MNEMONIC parameter1 ... parameterN
```

For example, the command to draw a circle of a given radius around the current point is:

```
CIRCLE 40
```

where CIRCLE is the command mnemonic, and 40 is the desired radius. Parameter values may be entered as signed, base-10 numbers (such as 117 or 45), or as unsigned hexadecimal numbers when preceded by a # sign (such as #FF, #9E, or #0A0F).

The parameter values must be separated by a comma or space, as desired. This can be used to set off sections of the commands, or to clarify what the command is doing:

```
ARC 10 45,135
```

This command draws an arc of radius 10, with starting angle 45 degrees and ending angle 135. You can also use commas to set off x,y pairs or groups of values, as you will see in the examples in this manual. Some further examples of locally-typed commands are:

```
MOVABS 0,0  
DRWREL 30 150  
VECPAT #F0F0
```

The Model One command interpreter includes a HELP subsystem. To receive a list of all the Model One commands, type

```
HELP
```

HELP may also be used to obtain parameter information, by typing

HELP COMMAND

For example, typing

HELP MOVABS

displays this information:

```
MOVABS: OPCODE = 001
1. <16 BIT SIGNED NUMBER>
2. <16 BIT SIGNED NUMBER>
```

and typing

HELP PRMFIL

displays

```
PRMFIL: OPCODE = 031
1. OFF ON
```

Abbreviations may be used. The abbreviation for each command is the shortest unique identifier, such as MOVA for MOVABS, CIRCL for CIRCLE, and so on.

The QUIT command is used to leave GRAPHICS mode and return to ALPHA mode. Type the ENTERGRAPHICS control character [CTRL-D] and then a QUIT command, to verify that the Model One returns to ALPHA mode. In ALPHA mode, you can once again use your terminal to communicate with the host computer.

## 2.2 GRAPHICS Mode From the Host Computer

When the ENTERGRAPHICS character is sent from the host computer to the Model One, the host port which sent the ENTERGRAPHICS character, whether it was the HOSTSIO, HOSTGPIB, or HOSTPIO port, is put into GRAPHICS mode.

A host computer application program, using the Model One's Host FORTRAN library, sends the ENTERGRAPHICS control character by calling the subroutine ENTGRA. All graphics commands from the host computer to the Model One must be sent between a call to the ENTGRA subroutine and a call to the QUIT subroutine, as shown in the example below. (The host FORTRAN library is described in more detail in The FORTRAN Library.)

```
INTEGER I,J,K,L
INTEGER IRAD,IANG,IJANG      Model One in ALPHA Mode
:                             No graphics commands may be
:                             issued until a CALL ENTGRA
:                             command is given.
:                             Alphanumeric I/O is allowed.
:
CALL ENTGRA                  Enter GRAPHICS Mode.
:
:
CALL MOVABS(I,J)             Graphics subroutine calls are
CALL CIRCLE(M)               allowed: no alphanumeric I/O
```

Model One/80 Modes of Operation

```

CALL DRWABS(K,L)           to Model One.
:
:
CALL QUIT                  Quit GRAPHICS Mode; return to
:                          ALPHA Mode. Alphanumeric data
:                          to local terminal allowed.
:
CALL ENTGRA                Reenter GRAPHICS Mode.
:
:
CALL CIRCLE(IRAD)
CALL ARC(IRAD,IANG,JANG)
CALL DRAWBS(J,K)
:
:
CALL QUIT                  Exit GRAPHICS Mode.
STOP
END

```

If you are not using the Model One's host FORTRAN library, your program will have to send the ENTERGRAPHICS control code to the Model One in some other way. You should verify that your host programs can send the ENTERGRAPHICS control code and properly issue graphics commands before continuing. Graphics commands from the host computer are composed of a stream of opcodes and parameters. Each opcode is one byte, ranging from 00H to FFH. The opcode for a MOVABS command is 01H. The opcode for a CIRCLE command is 0EH. The MOVABS command has four bytes of parameter data which must immediately follow its opcode. Thus, the byte stream for a complete MOVABS command would be:

```

01H    03H FFH 00H 0FH
opcode parameters

```

This host command stream is equivalent to typing:

```
MOVABS #03FF,#000F
```

or

```
MOVABS 1023,15
```

at the local alphanumeric keyboard.

The CIRCLE command has one two-byte parameter which gives the radius of the circle to be drawn. Thus, the CIRCLE command would be:

```

0EH    01H 03H
opcode parameters

```

when sent from the host computer. This command is equivalent to typing:

```
CIRCLE #0103
```

or

CIRCLE 259

at the local alphanumeric terminal.

The QUIT command, which returns the Model One to ALPHA mode, has an opcode of FFH (with no parameters). Each command stream to the Model One which is to be followed by normal terminal I/O must end with a QUIT command. Thus, a complete command stream to the Model One would be:

```
04H (or 84H) 01H    03H FFH 00H 00H 0EH    01H 03H    FFH
ENTERGRAPHICS opcode parameters          opcode parameters QUIT
```

Note that the ENTERGRAPHICS control code may or may not have its high bit set; it is interpreted as a 7-bit ASCII code.

If the Model One has been configured to accept 8-bit binary host transmission (see the Model One Installation Guide), each byte in the command stream is sent in a single character from the host computer. For 8-bit binary to work properly, the host computer may not use the eighth bit of each character for parity or force it high or low. It also must allow every control code to pass to the terminal. In addition, the host computer must not insert carriage control characters unpredictably into the output stream.

ASCII hex format removes all of these restrictions at the cost of doubling the transmission time. In ASCII hex format, each byte is expanded into two hexadecimal characters, one for the high nibble (four bits) of the byte, followed by a second character for the low nibble of the byte. The byte stream

```
01H 03H FFH 00H 0EH
```

requires only five characters when using binary transmission. In ASCII hex, it would require ten characters:

```
"0" "1" "0" "3" "F" "F" "0" "0" "0" "F"
```

In ASCII hex format, all carriage control characters are ignored, so that a FORTRAN WRITE statement or a BASIC PRINT statement will work properly.

### 2.3 The Model One's Input and Output Buffers

While the local terminal is in GRAPHICS mode, any characters sent by the host to any of the Model One's host ports (HOSTSIO, HOSTPIO, or HOSTGPIB) are stored in the Model One's input queue until you type QUIT to reenter ALPHA mode. The HOSTSIO input queue defaults to 4096 bytes (characters). The HOSTPIO input buffer is 2 characters.

Similarly, while the host port is in GRAPHICS mode, any characters typed at the local alphanumeric terminal are stored in the ALPHASIO input queue until the host computer issues a QUIT command to exit GRAPHICS mode. The ALPHASIO input queue holds 64 characters.



Table 1 shows the default I/O buffer sizes for the Model One.

Port	Input	Output
HOSTSIO	2048	64
ALPHASIO	64	64

Table 1 Default I/O Buffer Sizes

### 3.0 The Model One's Special Characters

The Model One responds to a set of special control characters, such as the ENTERGRAPHICS [CTRL-D] described above, to perform certain functions. These are:

Function	Default Special Character
Enter GRAPHICS Mode	[CTRL-D]
Send BREAK to host	[CTRL-P]
Execute WARMstart	[CTRL-[] or [ESC]
Kill current line	@
Backspace	[CTRL-H]
ACKnowledge	[CTRL-F]
Negative ACKnowledge	[CTRL-U]
Enter local debugger	[CTRL-X]
Suspend communications	[CTRL-S]
Restart communications	[CTRL-Q]

Table 2 The Model One's Special Characters

Note that [CTRL-S] and [CTRL-Q] are not sent to the host if you are not using the HOSTSIO port. They do, however, stop and start communications to the local terminal.

The SPCHAR char,flag,code command can be used to change the default special characters for the Model One. The most commonly modified default special character is the WARMstart character, which defaults to an ESCAPE. The SPCHAR command to change the WARMstart character to a [CTRL-G] (the bell) is:

```
SPCHAR WARM,1,7
```

The command

```
SPCHAR 2,0,0
```

disables the WARMstart character entirely. (While this is dangerous during program development, it may be desirable after debugging has been completed.) The SPCHAR command is described in detail in the Model One Command Reference. The SAVCFG command can be used to save changed special characters, if desired.



RASTER TECHNOLOGIES  
MODEL ONE/80  
COORDINATES AND  
IMAGE MEMORY ADDRESSING

Revision 1.0 February 27, 1984



MODEL ONE COORDINATES AND IMAGE MEMORY ADDRESSING  
February 27, 1984

Copyright 1984 by Raster Technologies, Inc. All rights reserved. No part of this work covered by the copyrights herein may be reproduced or copied in any form or by any means--electronic, graphic, or mechanical, including photocopying, recording, taping, or information and retrieval systems--without written permission.

NOTICE:

The information contained in this document is subject to change without notice.

RASTER TECHNOLOGIES DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS MATERIAL (INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESS OR IMPLIED. RASTER TECHNOLOGIES SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS MATERIAL.

This document contains proprietary information which is protected by copyright.

WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual may cause interference with regard to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.



Table of Contents

1.0	The Coordinate Origin .....	4
2.0	The Clipping Window .....	5
3.0	The Screen Origin .....	6
4.0	Display Scale Factor .....	6
5.0	Zooming the Display .....	6
6.0	Inhibiting Screen Refresh .....	7
7.0	The Current Point .....	7
8.0	The Coordinate Registers .....	8
9.0	The Crosshairs .....	11

Figures

Figure 1	The Default Addressing Space for the Model One/80 .....	4
Figure 2	Resetting the Coordinate Origin .....	5

Tables

Table 1	Model One Coordinate Register Assignments .....	8-10
---------	---	------

COORDINATES AND IMAGE MEMORY ADDRESSING

The Model One uses a two-dimensional coordinate system to describe the graphic entities that are drawn into image memory. Each coordinate is stored as an X component and a Y component; these components are stored within the Model One as two's complement 16-bit integers. The Model One's graphics commands use this 16-bit "address space" to specify the position of points, lines, circles, arcs, polygons, rectangles, and so on.

Because the physical image memory of the Model One is not large enough to allow a full 16-bits of addressing in both the X and Y dimensions, the physical image memory covers only a patch of the 16-bit address space.

On the Model One/80, the physical image memory ranges from (-640,-512) to (639,511). On some systems, the display may be only 768x576; in this case, the off-screen memory may be displayed by panning the screen with the SCRORG command. Alternatively, the off-screen memory can be used to store fonts, menus, and so on.

1.0 The Coordinate Origin

The CORORG x,y command can be used, immediately after a COLDstart, to reposition the physical image memory in the 16-bit address space. Figure 1 shows the default coordinate system for the Model One/80. Figure 2 shows the use of the CORORG command to set up the screen so that (0,0) is in the lower-left hand of the screen and all points can be addressed as positive numbers.

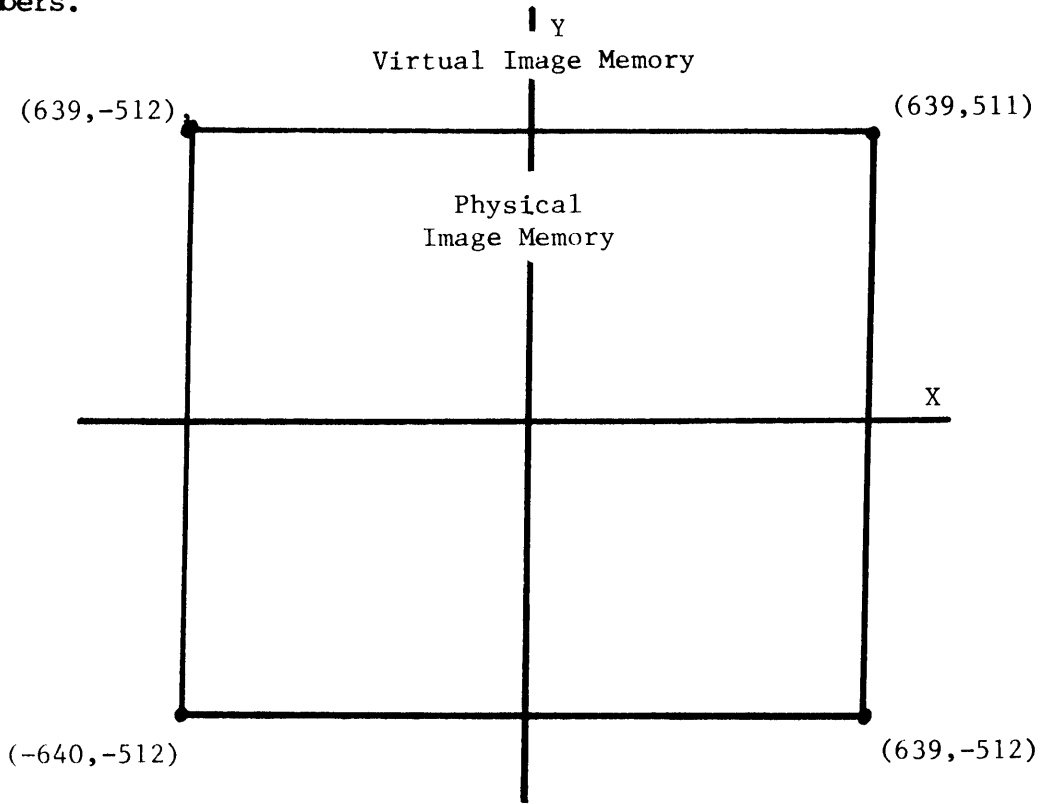


Figure 1 The Default Addressing Space for the Model One/80

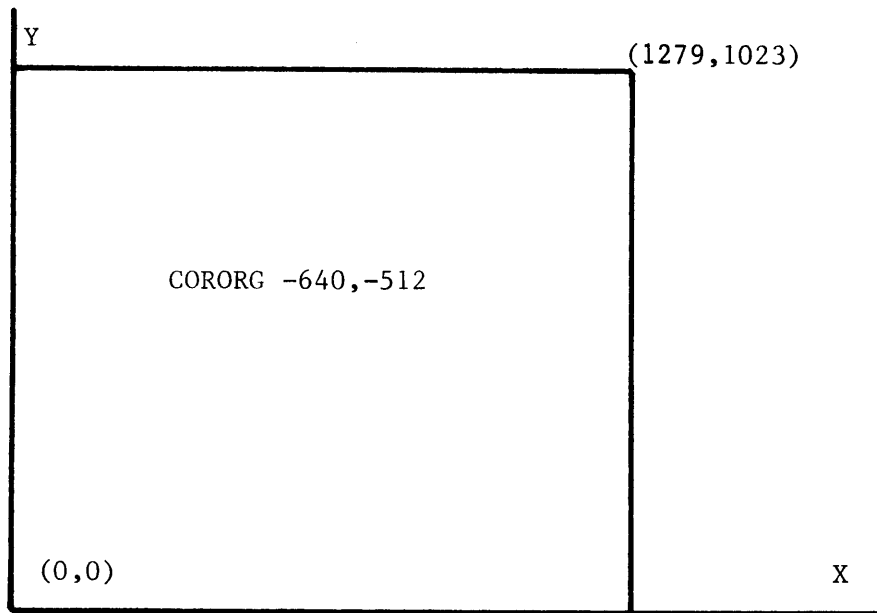


Figure 2 Resetting the Coordinate Origin

The CORORG command should be issued only immediately following a COLDstart, because all coordinate registers are modified by the CORORG command.

## 2.0 The Clipping Window

When a graphic primitive, such as a line or circle, is to be drawn, its coordinates are given as 16-bit addresses. When the primitive is then drawn by the Model One, it is clipped so that it is drawn only into the physical image memory.

The Model One automatically clips all graphics primitives to a preset clipping window. If no clipping window was specified, it draws only that portion of all graphics primitives which lie in physical image memory. To support this clipping, the Model One maintains a clipping window; the clipping window defines a rectangular area of the virtual address space outside of which nothing is drawn. The default clipping window is defined by the physical image memory. In the Model One/80, the default clipping window has the corners (-640,-512), (-640,511), (639,511), (639,-512).

The WINDOW x1,y1 x2,y2 command changes the position of the clipping window by respecifying the lower-left and upper-right corners of the window. For example, the command WINDOW 0,0 255,255 defines the clipping window with a lower-left corner of (0,0) and an upper-right corner of (255,255). Positioning all or part of the clipping window outside of the physical image memory clips the window itself, as clipping occurs automatically beyond image memory bounds.

### 3.0 The Screen Origin

Ordinarily, the video monitor displays the entire contents of image memory. The displayed image is essentially a window into image memory, however, and it can be modified in both size and position.

The screen origin specifies the physical image memory location that will appear at the center of the screen. The screen origin may be placed only at specific intervals. For the Model One/80, the vertical interval is two pixels; the horizontal interval is 20 pixels for an image memory size of 1280x1024. The default screen origin is (0,0).

Note that, as a result of these defined horizontal intervals, the image may shift slightly when zooming by an odd number, such as 3 or 9.

The SCRORG x,y command is used to change the position of the screen origin within image memory. For example, SCRORG -40,-40 puts the point (-40,-40) at the center of the screen. If the image is large enough, wrapping around of the image can be seen when the command is executed. The Model One automatically wraps the image around (side-to-side and top-to-bottom panning) when any part of the screen falls off the edge of physical image memory, as the screen refresh addresses are generated modulus 1280 horizontally and modulus 1024 vertically. This wraparound cannot be disabled.

### 4.0 Display Scale Factor

The display scale factor determines the number of pixels that are displayed on the screen:

For the Model One/80:

<u>Scale Factor</u>	<u>Number of Pixels</u>
1	1280x1024
2	640x512
4	320x256
8	160x128
16	80x64

### 5.0 Zooming the Display

The size of the window of image memory that is displayed is controlled by the ZOOM and ZOOMIN commands, which modify the display scale factor.

The ZOOM factor command allows explicit definition of the display scale; the display scale is from 1 to 16, inclusive. For example, ZOOM 4 sets the display scale to 4.

The ZOOMIN command zooms in the display by a factor of two; ZOOMIN sets the scale to 4 if the current scale is 2, or 8 if the current scale is 4. Finally, if the current scale is 8, the display is restored to a scale factor of 1 by a ZOOMIN command, as follows:

<u>Current Scale</u>	<u>After ZOOMIN</u>
1	2
2	4
3	6
4	8
5	10
6	12
7	14
8-16	1

ZOOM and ZOOMIN do not change the current screen origin, clipping window, or coordinate origin.

When you zoom in the display, the number of pixels displayed on the screen is reduced, so that each pixel takes up a larger display area. At a zoom scale factor of eight, for example, you can see each pixel quite clearly.

To zoom in a specific portion of the image, you should move that portion of the image to the center of the screen, using the SCRORG command, then execute the zoom command.

#### 6.0 Inhibiting Screen Refresh

The BLANK flag command totally inhibits screen refresh, leaving image memory available all of the time for updates by the vector generator, pixel processor, or optional host DMA. BLANK 1 inhibits screen refresh: BLANK 0 restores normal screen refresh. When screen refresh is inhibited, the displayed image is forced to black.

The BLANK command can be used to increase the pixel writing rate, since more time is available for vector writing when screen refresh is inhibited.

#### 7.0 The Current Point

All Model One commands which draw graphics primitives use the current point as a reference. For example, the CIRCLE command draws a circle of given radius around the current point.

To draw a line in image memory from a given starting point to a specified ending point, the current point must first be set to the starting point of the line. Then, the line is drawn from the current point to the specified ending point. The line-drawing commands leave the end point of the line as the new current point after the line is drawn.

Five commands move, and therefore modify, the current point: MOVABS, MOVREL, MOV3R, MOV2R, and MOVI.

The MOVABS x,y command specifies a new current point: MOVABS -10,-10 sets the current point to (-10,-10).

The MOVREL dx,dy command moves the current point a relative distance from the previous current point. For example, the command sequence

```
MOVABS 143,271  
MOVREL -10,-10
```

would place the current point at (133,261).

The MOV3R dx,dy and MOV2R dx,dy commands are special forms of the MOVREL command for use when the displacement to the new current point is small. The MOV2R command requires only two bytes to be sent from the host computer; the MOV3R command requires three bytes. The MOVABS and MOVREL commands require five bytes when sent from the host. Details of the MOV2R and MOV3R commands are given in the Command Reference.

The MOVI creg command moves the current point to the point specified by the indicated coordinate register. For example, MOVI 2 moves the current point to the point specified by coordinate register 2 (coordinate register 2 gives the location of the cursor on the digitizing tablet). Details of the coordinate registers are given in the next section.

### 8.0 The Coordinate Registers

The Model One stores 64 coordinate registers internally. The coordinate registers store coordinate values within the Model One: some have a predefined function within the Model One, others are available for programmer use.

Each coordinate register (CREG) stores a 16-bit X coordinate and a 16-bit Y coordinate. Table 1 shows the coordinate register assignments for the Model One.

Coordinate Register	Function
0 CURPNT	Current point: used as a reference point by graphics commands. The current point is modified by MOVE and DRAW commands.
1 JOYSTK	Joystick or trackball location, updated automatically by the Model One every 1/30th second.
2 DIGTZR	Digitizing tablet cursor location, updated automatically by the Model One every 1/30th second.
3 CORORG	Coordinate origin: used to position physical image memory within the virtual address space. The coordinate origin is modified by the CORORG command.
4 SCRORG	Screen origin: specifies the point which appears at the center of the screen. The screen origin is changed by the SCRORG command. CREG 4 is used for horizontal and vertical panning.
5 XHAIRO	Crosshair 0 current location: changes made to this register move crosshair 0. The crosshair is enabled/disabled using the XHAIR command.

Table 1 Coordinate Register Assignments  
(continued on next page)

Table 1 Coordinate Register Assignments (continued)

Coordinate Register	Function
6 XHAIR1	Crosshair 1 current location: changes made to this register move crosshair 1. The crosshair is enabled/disabled using the XHAIR command.
7-8	Reserved.
9 LWNORG	Clipping window, lower-left corner.
10 UWNORG	Clipping window, upper-right corner.
11,12	Diagonal corners: source window for PIXMOV.
13	PIXMOV destination window
14	PIXMOV pixel writing direction
15-16	Reserved.
17	Hardware cursor location (see CURSOR command).
18-19	Reserved.
20-63	Available for use by applications programmer.

Table 1 Coordinate Register Assignments

Four commands load and alter the coordinate registers: CLOAD, CMOVE, CADD, and CSUB. The command READCR reads back or displays the contents of a specified coordinate register.

The CLOAD creg x,y command loads a given 16-bit X coordinate and a 16-bit Y coordinate into the specified coordinate register. For example, CLOAD 25 -75,75 loads the point (-75,75) into coordinate register 25.

The CMOVE cdst, csrc command copies data from one coordinate register into another: CMOVE 0 2 or CMOVE CURPNT DIGTZR moves the contents of coordinate register 2 (the cursor location) into coordinate register 0 (the current point). The command CMOVE 0 2 thus specifies that the new current point is to be taken from the cursor location on the digitizing tablet.

CADD csum, creg and CSUB cdif, creg add and subtract coordinates between two specified coordinate registers: CADD 0 21 adds the contents of CREG 21 to the contents of CREG 0.

Note that in the coordinate register pairs specified as parameters for `CMOVE`, `CADD`, and `CSUB`, the register which is to be modified is specified first.

Several graphics primitive commands include an indirect addressing form. In this form, coordinates which are needed to execute the command are given by specifying a coordinate register instead of being supplied directly: `MOVI` moves to the point given by a coordinate register, `RECTI` uses a coordinate register to specify the diagonal corner of a rectangle, and so on.

Finally, the command `READCR creg` reads or displays the contents of a specified coordinate register. For example, `READCR 0` displays the contents of `CREG 0` (the current point).

### 9.0 The Crosshairs

The `XHAIR num,flag` command controls the crosshairs. For the Model One/80, two firmware-controlled crosshairs are available: crosshairs 0 and 1. `num` gives the crosshair number; if `flag=1`, the crosshair is displayed. If `flag=0`, the crosshair returns to its default "invisible" state.

In addition, there is a hardware full-screen cursor, controlled with the `CURSOR` command. The cursor should be used whenever possible, as it requires less processing overhead than the `XHAIR` crosshairs; the cursor does not reduce system performance, while the crosshairs may.

Crosshairs, when displayed, take their location from the coordinate registers. Crosshair 0 uses `CREG 5`. Crosshair 1 uses `CREG 6`. The cursor uses `CREG 17`. It is enabled with the command `CURSOR number flag`; `number` must be 0, and `flag=0` disables the cursor. `flag=1` enables the cursor.

For the crosshair to track the cursor on the digitizing tablet, it is necessary to write a small macro. (Macro programming is described in detail in Macro Programming.)

```
MACDEF 10
CMOVE 5 2    Load crosshair 0 location with cursor location
MACEND
BUTTBL 0 10  Execute macro 10 every 1/30th second
```

This macro will execute every 1/30th of a second, writing the cursor location into the crosshair location. For the crosshair to be displayed, it is still necessary to execute the command `XHAIR 0 1`. The cursor is enabled with `CURSOR 0 1` and disabled with `CURSOR 0 0`. To turn off the crosshair, use the command `XHAIR 0 0`.

The crosshair colors are determined by value registers 1 (crosshair 0) and 2 (crosshair 1). The crosshair color is then XORed with the color in image memory to display the crosshair; the default crosshair value is 255,255,255 for the 24-bit Model One/80. For the 8-bit Model One/80, the default crosshair value is 255,0,0.



The hardware cursor is always forced to white when it is displayed, regardless of the pixel value it overlays.

Using the crosshairs increases the processing time for some commands slightly. If maximum performance is important, turn off the crosshairs or use the hardware cursor whenever possible.



RASTER TECHNOLOGIES  
MODEL ONE/80  
PIXEL VALUES,  
LOOK-UP TABLES,  
AND IMAGE MEMORY

Revision 1.0 February 27, 1984



MODEL ONE/80 PIXEL VALUES, LOOK-UP TABLES, AND IMAGE MEMORY  
February 27, 1984

Copyright 1984 by Raster Technologies, Inc. All rights reserved. No part of this work covered by the copyrights herein may be reproduced or copied in any form or by any means—electronic, graphic, or mechanical, including photocopying, recording, taping, or information and retrieval systems—without written permission.

NOTICE:

The information contained in this document is subject to change without notice.

RASTER TECHNOLOGIES DISCLAIMS ALL WARRANTIES WITH RESPECT TO THIS MATERIAL (INCLUDING WITHOUT LIMITATION WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE), EITHER EXPRESS OR IMPLIED. RASTER TECHNOLOGIES SHALL NOT BE LIABLE FOR DAMAGES RESULTING FROM ANY ERROR CONTAINED HEREIN, INCLUDING, BUT NOT LIMITED TO, FOR ANY SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR IN CONNECTION WITH THE USE OF THIS MATERIAL.

This document contains proprietary information which is protected by copyright.

WARNING: This equipment generates, uses, and can radiate radio frequency energy and if not installed and used in accordance with the instructions manual may cause interference with regard to radio communications. It has been tested and found to comply with the limits for a Class A computing device pursuant to Subpart J of Part 15 of FCC Rules, which are designed to provide reasonable protection against such interference when operated in a commercial environment. Operation of this equipment in a residential area is likely to cause interference in which case the user at his own expense will be required to take whatever measures may be required to correct the interference.



Table of Contents

1.0	The Model One/80 and Its Look-Up-Tables .....	4
1.1	8-Bit Model One/80 .....	5
1.2	24-Bit Model One/80 .....	6
2.0	The Look-Up-Tables .....	7
3.0	The Value Registers .....	8
4.0	The Pixel Processor .....	10
5.0	Write-Enable and Read-Enable Masks .....	11
6.0	Blinking Colors and the Blink Table .....	12
7.0	The CLEAR and FLOOD Commands .....	12

Examples

Example 1	A Crawling Circle Demonstrates the Look-Up-Table	4
-----------	--	---

Tables

Table 1	LUT Values Corresponding to VALK Commands .....	6
Table 2	Value Register Assignments .....	9
Table 3	PIXFUN Modes and Functions .....	11

MODEL ONE/80 PIXEL VALUES, LOOK-UP TABLES, AND IMAGE MEMORY

The eight-bit Model One/80 use a single 8-bit-in, 24-bit-out look-up-table. Each pixel in image memory may be assigned a value from 0 to 256, corresponding to a location in this look-up-table. In the 24-bit Model One/80, there are three 8-bit-in, 8-bit-out look-up-tables, each corresponding to a primary color: red, green, and blue. Each pixel is assigned a red value from 0 to 256, a green value from 0 to 256, and a blue value from 0 to 255.

In the eight-bit system, the 64-position look-up-table is initialized to the default values assigned for the Model One/25's 1K addressing mode, ensuring compatibility with other members of the Model One/family. The easiest way to examine the default look-up-table is with the following macro program (see Macro Programming for details of macro programming):

```

MACDEF 10          Define a macro program
VADD 0 10         Add the contents of value register 10
                  to the current value
CADD 0 21         Add the contents of coordinate register
                  21 to the current point
CIRCLE 50         Draw a circle of radius 50, using the
                  current value
MACEND           End the macro
VALK 0           Select black as the starting color
VLOAD 10 1,0,0   Load value register 10 with 1
CLOAD 21 1,1     Load coordinate register 21 with (1,1)
PRMFIL ON       Select filled primitives
MOVABS -512,-512 Move to lower-left corner of the screen
BUTTBL 0 10     Execute macro 10 every 1/30th second
    
```

Example 1 A Crawling Circle Demonstrates the Look-Up Table

You can enter these commands by typing the ENTERGRAPHICS character at the terminal, then typing the commands exactly as they are written. You can stop execution by typing the command BUTTBL 0 0 at the terminal.

1.0 The Model One/80 and Its Look-Up-Tables

There are two versions of the Model One/80. The first version is an eight-bit version with a 256-position look-up-table addressed by the VALK command. (It can also be addressed with the other value commands, but the VALK command is most straightforward.)

The second version is a 24-bit full-color Model One/80. It uses three 8-bit-in, 24-bit-out look-up-tables, one each for red, green, and blue. The various value commands set the pixel values differently. The RGBTRU command, which enables and disables true-color mode (also called RGBTRU mode), also influences how the value commands work.