3. Coordinate Register Assignments

REG 0	Current point. Starting point of graphics primitives. Updated by a MOVE or DRAW.
REG 1	Joystick/trackball cursor location. Current coordinate from joystick or trackball. Updated automatically.
REG 2	Digitizing tablet cursor location. Current coordinate from the digitizing tablet. Updated automatically.
REG 3	Coordinate origin. Coordinates of the center of image memory.
REG 4	Screen origin. Coordinates of the pixel in the center of the screen.
REG 5	Crosshair location in image memory.
REG 6-8	Reserved.
REG 9	Clipping window origin. Lower-left corner of current clipping window. All vectors are clipped to this window.
REG 10	Clipping window origin. Upper-right corner of current clipping window. All vectors are clipped to this window.
REG 11, 12	Diagonal corners for PIXMOV source window definition.
REG 13	Start corner for PIXMOV destination window.
REG 14	Direction of pixel writing for PIXMOV destination window.
REG 15**	Overlay plane origin.
REG 16	Reserved for future use.
REG 17	Cursor origin for Model One/80.
REG 18	Reserved for future use.
REG 19***	Center of pick aperture.
REG 20-63	Unassigned. Available for temporary coordinate storage.

**Model One/10 only.
***With Display List Firmware only.

4. Value Register Assignments

REG 0 Current Value	The value used in all graphics primitives commands.
REG 1	Value used for Crosshair 0.
REG 2	Value used for Crosshair 1.
REG 3	Fill mask used for area fills.
REG 4**	Color assignment for Overlay Plane 0.
REG 5*	Color assignment for Overlay Plane 1.
REG 6	Reserved for future use.
REG 7-15	Available for temporary value storage.
REG 16, 19...40*	Foreground color, Alphanumeric Windows 0-8.
REG 17, 20...41*	Background color, Alphanumeric Windows 0-8.

14. Value Register Assignments (Continued)

VREG 18, 21...42*	Cursor color, Alphanumeric Windows 0-8.
VREG 43, 44	Reserved for future use.
VREG 45***	Hilite color.
VREG 46-50	Reserved for future use.
VREG 51-63	Available for temporary value storage.

*Model One/80 only.
**Model One/10 only.
***With Display List Firmware only.

15. System Configuration

DFTCFG	Restore all ports to default configurations.
DISCFG	Display current system configurations.
SAVCFG	Save configuration set with SYSCFG.
SYSCFG HOST	Mode [ASCII/BINARY]
SYSCFG SERIAL	[Port-mnemonic] [RTS on/off] [CTS on/off] [STOP 1/2] [BITS 7/8] [PARITY e/o/1/h/n] [BAUD rate] [XIN on/off] [XOUT on/off] [CTRL on/off]
SYSCFG SERIAL	TABLET [GTCO/SUMMA].

16. Default Port Configurations

Port Mnemonic	RTS	CTS	Baud	Parity	XIN	XOUT	CTRL	STOP	NBITS
KEYBSIO*	off	off	300	none	on	off	on	1	8
TABLETSIO	off	off	9600	none	on	off	off	2	8
HOSTSIO	off	off	9600	none	off	on	off	2	8
ALPHASIO*	off	off	9600	none	on	off	on	2	8

*Model One/80 only.

17. Alphanumeric Terminal Emulation†

ALPHEM flag	Enable (flag = 1 or ON) or disable (flag = 0 or OFF) the alphanumeric terminal emulator. Route text to selected window. [$C2_H$]
BOLD flag	Enable (flag = 1 or ON) or disable (flag = 0 or OFF) drawing to bold text. [CC_H]
DEFWIN window, x1, y1, x2, y2, x-size, y-size, bitm, bankm	Define size and positions of indicated window number. (x1,y1) define first corner; (x2,y2) define diagonal corner. x-size, y-size define text size; bitm, bankm define write mask for window (see WRMASK command). [CO_H]
DELWIN window	Delete window. [$C3_H$]
DIRCUR x, y	Move cursor to character position x, y within window. [$C4_H$]
GETCUR	Return Model One coordinates of cursor in currently-selected window. [$C9_H$]
GETPOS	Return character position of cursor in currently-selected window. [$C5_H$]

17. Alphanumeric Terminal Emulation† (Continued)

GETWIN	Return number of active window (−1 for no active window). [CE_H]
HOME	Move cursor to character position (0, 0), the upper-left corner of the window. [CF_H]
MOVCUR x, y	Move cursor to Model One coordinate x, y within window limits. [$C8_H$]
OVRSTK flag	Enable (flag = 1 or ON) or disable (flag = 0 or OFF) overstriking of text. [CD_H]
SCROLL flag	Enable (flag = 1 or ON) or disable (flag = 0 or OFF) scrolling of text. [CA_H]
SELWIN window	Select window as defined by DEFWIN. Set routing for ALPHEM command. [$C1_H$]
SETCUR flag	Enable (flag = 1 or ON) or disable (flag = 0 or OFF) cursor. [$C7_H$]
SETSIZ xscale, yscale	Set x, y scaling (multiples of 16 pixels). Default is (1, 1). [$C6_H$]
WRAP flag	Enable (flag = 1 or ON) or disable flag = 0 or OFF) wraparound of text. [CB_H]

†Model One/80 only. The Model One/10 emulates the VT100/ANSI X3.64.

18. Display List Firmware†

DELPID	Delete primitives with pick ID = PIDREG and with segment ID = SEGREG. [EC_H]
EXMODE mode	Set display controller to specified execution mode. Mode 0 = NORMAL DRAW, Mode 1 = PICK, Mode 2 = HILITE, Mode 3 = UNHILITE. [$7B_H$]
IGNORE flag	Set display controller to ignore subsequent incoming commands. Use only in systems without local display lists. Flag = 0, OFF, Flag = 1, ON. [$7C_H$]
PICKID pickid	Assign a pick ID to all commands immediately following the PICKID command until the next PICKID command or until end of segment. [$D9_H$]
POP state, arg	Restore specified state by popping it off current stack. [76_H]
PUSH state, arg	Save specified state or item by placing it on top of the current stack. [75_H]
RDPICK type, startent, totalent	Read back information. [EF_H]
RDREG reserved	Read back current SEGREG and PIDREG. [EO_H]
RDXFORM type	Read back matrix elements of current 2-D transformation. [$D6_H$]
SEGAPP segment	Reopen specified segment for appending commands to it. SEGEND ends the append. [DB_H]

18. Display List Firmware† (Continued)

SEGCOP segment2, segment1	Copy segment1 into segment2. [$E7_H$]
SEGDEF segment	Begin segment definition. [DC_H]
SEGDEL segment	Delete specified segment. [DE_H]
SEGEND	End segment definition. [DD_H]
SEGINIT reserved, blocksize	Initialize Display List Firmware. [$E1_H$]
SEGINQ segment	Read back attributes of specified segment. [$E5_H$]
SEGREF segment	(1) Execute a segment according to current execution mode. (2) Nest specified segment within current open segment. [$D8_H$]
SEGREN segment2, segment1	Rename segment1 to segment2. [DA_H]
SETATR segment,attribute, flag	Set VISibility (= 0) or PICKability (= 1) of specified segment. Flag 1 = ON, Flag 0 = OFF. [$E6_H$]
SETGL global, value	Set: pick aperture; PIDREG; SEGREG; number of picks to ignore during picking; or type of information to store in pick buffer during picking. [46_H]
SYSTAT infotype	Return information on system memory usage and availability. [$E4_H$]
XFORM2D, type,arg	Map 2-D coordinates from WCS into DCS; translate; scale; rotate. [$7D_H$]
XMOVE type,reg2,reg1	Map 2-D coordinates in reg1 from WCS into DCS or from DCS into WCS, placing transformed coordinates in reg2. [$7F_H$]

†Optional with Model One/80 only. Standard on Model One/10.

9 Executive Park Drive,
North Billerica, MA 01862
August, 1984

10272091

Raster Technologies

MODEL ONE/80 and MODEL ONE/10 Command Reference Card

This card summarizes the Model One/80 and Model One/10 firmware commands. It is for quick reference only.
Brackets [ ] indicate the hexadecimal opcode of each command.

1. Help

HELP	List all commands.
HELP mnemonic	Give information on specified command.

2. Graphics Primitives

ARC rad, a1, a2	Draw arc of radius rad. Starting angle is a1; ending angle is a2. [11_H]
AREA1	Area fill. Boundary is any pixel different in value from the current point. The area is filled with current value. [13_H]
AREA2 vreg	Area fill. Boundary pixel value given in vreg. [14_H]
CIRCI creg	Draw circle. Location given by creg lies on the circumference. [10_H]
CIRCLE rad	Draw circle of radius rad. [$0E_H$]
CIRCXY x, y	Draw circle. Point x, y lies on the circumference. [$0F_H$]
CLEAR	Flood current window to current pixel value. [87_H]
DRW2R dx, dy	Draw vector relative by dx, dy. [84_H]
DRW3R dx, dy	Draw vector relative by dx, dy. [83_H]
DRWABS x, y	Draw vector from current point to the point x, y. [81_H]
DRWI creg	Draw vector to location given by creg. [85_H]
DRWREL dx, dy	Draw vector relative by dx, dy. [82_H]
FILMSK rmsk, gmsk, bmsk	Image data is ANDed with mask before checking value in AREA fill commands. [$9F_H$]
FLOOD	Flood displayed image memory to current pixel value. [07_H]
MOV2R dx, dy	Move relative by dx, dy. [04_H]
MOV3R dx, dy	Move relative by dx, dy. [03_H]
MOVABS x, y	Move absolute location of current point to x, y. [01_H]
MOVI creg	Move to location given by coordinate register creg. [05_H]
MOVREL dx, dy	Move relative by dx, dy. [02_H]
POINT	Set current point to current pixel value. [88_H]

2. Graphics Primitives (Continued)

POLYGN npoly, verts	Draw polygons. Npoly gives number of polygons; for each polygon, verts gives number of vertices and the vertices. [12_H]
PRMFIL flag	Primitive fill. Filled primitives are drawn if flag = 1. If flag = 0, the perimeter of graphics primitives is drawn. [$1F_H$]
RECREL dx, dy	Draw rectangle. Diagonal corner is dx, dy away from current point. [89_H]
RECTAN x, y	Draw rectangle. Point x, y specifies diagonal corner. [$8E_H$]
RECTI creg	Draw rectangle. Location given by creg is diagonal corner. [$8F_H$]
TEXT1 string	Draw text string with Font 1. [90_H]
TEXT2 string	Draw text string with Font 2. [91_H]
TEXTC size, ang	Specify size of text and draw at angle ang. [92_H]
TEXTN xsize, ysize, xscale, yscale	Specify xsize, ysize, xscale, and yscale of text. [$A9_H$]
TEXTDN char, veclst	Define downloaded character in Font 2. [26_H]
TEXTRE	Restore default character set. [$B1_H$]
VAL1K val	Set current pixel value (for 1K mode). [$B0_H$]
VAL8 val	Set current pixel value to val. [86_H]
VAL16 val	Set current pixel value to specified 16-bit value. [45_H]
VALUE r, g, b	Set current pixel value to r, g, b. [06_H]
VTEXT1 string	Draw vertical text string with Font 1. [93_H]
VTEXT2 string	Draw vertical text string with Font 2. [94_H]

3. Look-up Table

LUT8 index, r, g, b	Make entry r, g, b at location given by index in Red, Green, and Blue LUTs. [$1C_H$]
LUT16** index, r, g, b	Make entry r, g, b at location given by index in Red, Green, and Blue LUTs. Access is to all 1024 LUT entries. [16_H]
LUTA index, entry	Make entry in all LUTs. Place entry at location given in index. [$1B_H$]
LUTB index, entry	Make entry in Blue LUT. Place entry at location given in index. [$1A_H$]
LUTG index, entry	Make entry in Green LUT. Place entry at location given in index. [19_H]
LUTR index, entry	Make entry in Red LUT. Place entry at location given in index. [18_H]
LUTRMP code, sind, eind, sent, ent	Load LUTs with ramp function. [$1D_H$]

**Model One/10 only.

4. Image Transmissions

PIXEL8 nrow, ncoln, val	Define image pixel-by-pixel (8-bit pixels). [29_H]
PIXELS nrows, ncols, r, g, b	Define image pixel-by-pixel (full 24-bit pixels). [28_H]
PIXEL16** nrows, ncols, val	Define image pixel-by-pixel (16-bit pixels). [41_H]
RNLN16** nrows, ncols, val, cnt	Transmit a run-length encoded image. Pixel value is val. (16-bit value). Horizontal count is cnt. [47_H]
RUNLEN nrows, ncols, r, g, b, cnt	Transmit a run-length encoded image. Pixel value is r, g, b (24-bit pixels). Horizontal count is cnt. (8-bit count) [$2A_H$]
RUNLN8 nrows, cols, val, cnt**	Transmit a run-length encoded image. Pixel value is val. (8-bit value). Horizontal count is cnt. (8-bit count) [$2B_H$]

**Model One/10 only.

5. Display Control

ASCII flag	Set host port input as free format ASCII if flag = 1, if flag = 0 binary. [$9B_H$]
BLANK flag	Blank screen when flag = 1, normal video is restored when flag = 0. [31_H]
COLD	Coldstart. Reset the Model One. [FD_H]
CORORG x, y	Load coordinate origin register with x, y. [37_H]
CURSOR 0 flag	Set full-screen hardware cursor. If flag = 0, OFF; flag = 1, ON. [$4A_H$]
FIRSTP flag	Inhibit writing of first pixel of vectors when flag = 1 or ON. [$2F_H$]
MEMSEL* memunit	Specify memory unit to be address by future graphics primitive commands. [48_H]
MODDIS flag	Select display mode. [$2C_H$]
PIXCLP* flag	Pixel processor clipping status. Clip on over/underflow flag = 1. [$3C_H$]
PIXFUN* mode	Set pixel processor mode. All vectors and DMA writes are affected. [$3B_H$]
PIXMOV	Initiate pixel mover transfer. Move window specified by CREG 11 and 12 as controlled by CREG 13 and 14. [BB_H]
PMCTL** flag, startsrcpl, startdespl, num, 0, 0, 0	Set bit planes to be moved with with PIXMOV. [BF_H]
QUIT	Exit graphics mode. [FF_H]
RDCNFG*	Return to current graphics port hardware and firmware configuration of attached Model One. [$D0_H$]
RDMASK mask	Set read mask. All pixel values read from image are ANDed with mask. [$9E_H$]
RGBTRU* flag	Set full-color display mode. flag = 1, ON; flag = 0, OFF. [$4E_H$]

5. Display Control (Continued)

RMSK16 mask	Set read mask for up to 16 bits of image memory. [43_H]
SCRORG x, y	Set screen origin register to x, y. [36_H]
SPCHAR char, flag, code	Redefine special characters (see Special Characters below). If flag = 0, disable, flag = 1, enable. [$B2_H$]
TEKEM flag	Invoke Tektronix emulator. [39_H]
VECPAT mask	Vector generator pattern register is set to mask. [$2E_H$]
VGWAIT frames	Inihibit transfer of vectors from vector queue for frames frame times. [30_H]
WAIT frames	Wait for given number of frame times before continuing command execution. [$3D_H$]
WARM	Warmstart. Reinitialize Model One. [FE_H]
WINDOW x1, y1, x2, y2	Set current window. Defined by diagonal x1, y1 and x2, y2. [$3A_H$]
WMSK16 mask	Set 16-bit write-enable mask. [44_H]
WRMASK bitm, bankm	Set write-enable mask. Bit planes indicated by bitm and banks indicated by bankm are write-enabled. [$9D_H$]
XHAIR num, flag	Enable/Disable Crosshair number num. If flag = 1 enable, if flag = 0 disable. [$9C_H$]
ZOOM* fact	Zoom by factor of fact = 1 to 16. [34_H]
ZOOMIN*	Zoom in by factor of 2. [45_H]

*Model One/80 only.
**Model One/10 only.

6. Shading and Depth-buffering*

FOLD flag, quad	Configure depth-buffer memory. [58_H]
SHMODE flag	Set shading mode. If flag = OFF or 0 = flat; if flag = ON or 1 = smooth. [56_H]
ZCLEAR zdepth	Clear depth-buffer and set to 16-bit specified depth. [50_H]
ZCLIP flag, negclip, posclip	Clip z values if flag = 1. If flag = 0 no clipping. negclip = z value which new z value must be greater than. posclip = z value which new z value must be less than [57_H].
ZFUNCT function	Update depth-buffer memory according to specified function. [53_H]
ZGRID type, numu, numv, grid1x, grid1y, grid1z, grid1val	Send a grid of z-patches. [55_H]
ZPATCH type, nvert, xvert1, yvert1, zvert1, vert1val	Interpolate, shade, and depth-buffer a polygonal patch. If type = 1 8-bit color value used. If type = 0 24-bit used. For each polygon, nvert gives number of vertices, xvert1, yvert1, zvert1 gives xyz values of first vertex, and vert1val is color of first vertex. If vert1val = 0 24 bits, if vert1val = 1 8 bits. [52_H]

6. Shading and Depth-buffering* (Continued)

ZRANGE dposz, dnegz	Set tolerance for ZFUNCT modes. dposz = 16-bit positive change (32,768 to 32,767); dnegz = 16-bit negative change (−32,768 to −32,767). [54_H]

*Model One/80 only.

7. Special Characters (Default Values)

ASCII Code	Char	Purpose
CTRL D	0	Enter Graphics
CTRL P	1	Break
ESC or CTRL[	2	Warmstart
@	3	Line Kill
CTRL H	4	Backspace
CTRL F	5	ACK
CTRL U	6	NACK
CTRL X	7	Invoke Debug
CTRL S	8	Suspend Communications
CTRL Q	9	Resume Communications

8. Readback

All serial readback commands require a 7-bit ASCII ACK.

RDKEYB* nchar, tchar, flag	Read keyboard [$F1_H$]
RDMODE flag	Set readback mode to ASCII decimal (flag = OFF or 0) or binary (flag = ON or 1). [$D3_H$]
READBU flag, cflag	Read button number. If flag = 1 wait for next button. If flag = 0 send number of last button pushed. If cflg = 1 send current digitizing tablet coordinate. If clfg = 0 send current joystick/trackball coordinate. [$9A_H$]
READCR creg	Read coordinate register. Send x,y to port in graphics mode. [98_H]
READER	Return byte with number of first error. [38_H]
READF func	Set pixel readback format. func specifies format. [27_H]
READP	Read pixel. Send value of pixel to port in graphics mode. [95_H]
READVR vreg	Read value register. Send pixel value to port in graphics mode. [99_H]
READW nrows, ncols, bf	Read window. Send value of pixels in window to port in graphics mode. [96_H]
READWE nrows, ncols, bf	Read window run-length encoded. Send values of pixels in window in run-length encoded form to port in graphics mode. [97_H]

*Model One/80 only.

9. Register Operations

CADD csum, creg	Place result of csum + creg in csum. [$A2_H$]

9. Register Operations (Continued)

CLOAD creg x, y	Load coordinate register creg with x, y. [$A0_H$]
CMOVE cdst, csrc	Move contents of csrc into cdst. [$A1_H$]
CSUB cdif, creg	Place result of cdif − creg in cdif. [$A3_H$]
VADD vsum, vreg	Place result of vsum + vreg into vsum. [$A6_H$]
VLOAD vreg, r, g, b	Load contents of value register vreg with r, g, b. [$A4_H$]
VMOVE vdst, csrc	Move contents of vsrc into vdst. [$A5_H$]
VSUB vdif, vreg	Place result of vdif − vreg into vdif. [$A7_H$]

10. Software Development

*	Program comment. [80_H]
ALPHAO strlen, string	Send text to local alpha-numeric display. [$B4_H$]
DEBUG flag	Enter/Exit Command Stream Translator. Exit when flag = 0, else enter. [$A8_H$]
DELAY factor	Delay transmission of characters. [$B6_H$]
HOSTO strlen, string	Send a text string to the host. [$B5_H$]
IN** address	Return a byte value from specified Model One/10 I/O space address. [71_H]
NULL	No operation. [00_H]
OUT** address value	Output specified byte value to specified Model One/10 I/O space address. [72_H]
PEEK addr	Display contents of CPU memory. [BD_H]
POKE addr, data	Change contents of addr in CPU memory. [BE_H]
REPLAY	Dump last 32 characters of HOSTSIO input buffer to ALPHASIO port. [BC_H]

**Model One/10 only.

11. Macro Programming

MACDEF num	Define Macro number num. Macro is terminated by MACEND command. [$8B_H$]
MACEND	End of Macro definition. [$0C_H$]
MACERA num	Erase Macro num. [$8C_H$]
MACRO num	Execute Macro num. [$0B_H$]

12. Interactive Device Support

BLINKC	Clear blink table. [23_H]
BLINKD lut, index	Disable Blink of specified lut, index. [21_H]
BLINKE lut, index entry1, entry2	Enable Blink specified lut, index. Use Entry 1 and Entry 2 as alternate values. [20_H]
BLINKR frames	Blink rate is frame times. [22_H]
BUTTBL index, nmac	Place Macro nmac in Button Table at location index. [AA_H]
BUTTON index	Execute macro indicated by Button Table at location index. [AB_H]
FLUSH	Empty function button event queue. [15_H]
RDPIXR vreg	Place value of pixel at current point in specified value register vreg. [AF_H]