VECTOR GENERAL

Reference Manual

Graphics Display Unit

Interactive Graphics Products

VG 101056

Reference Manual

Graphics Display Unit

1 JULY, 1976

VECTOR GENERAL INC. • 21300 Oxnard Street, Woodland Hills, California 91364

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Cont'd.)

LIST OF FIGURES

LIST OF TABLES

SECTION I
GENERAL INFORMATION

1.1 INTRODUCTION

The Vector General Graphics Display System is an interactive graphics cathode ray tube (CRT) display that may be connected to any computer system with standard input/output capability. The display interacts with an on-line user by displaying pictorial information on the surface of the cathode-ray tube and by accepting inputs from external control devices. The inputs are requested and processed by computer programs that alter and maintain the output picture being presented to the user. This manual contains information needed by the programmer to write programs that use the capabilities of the display to the best advantage. The topics included are a system description, an explanation of display principles, a discussion of the functional organization of the system, a description of optional control devices, a description of display interrupt operation, a description of display instructions with directions for their use, and a sample program.

1.2 SYSTEM DESCRIPTION

The display system contains the necessary features for interactive displays plus several optional features. The standard features are an interface unit, a display controller (DC), a dual digital-to-analog converter (DD), a vector generator (VG), and a display monitor(DM). The optional features are a character generator (CG), circle/arc generator (CAG), and three coordinate transformation gererators: two-dimensional (2D), two-dimensional with rotation (2DR), and three-dimensional (3D). Any of six interactive control devices may be connected to the system. The devices are an alphanumeric keyboard (KB), data tablet (DT), joystick (JS), 16 (or 32 optional) lighted function switches (FS) with manual interrupt, control dials (CD), and a light pen (LP). A simplified block diagram of the system is shown in Figure 1-1.

1.3 STANDARD FEATURES

The computer communicates with the display controller by way of the interface through three types of channels.

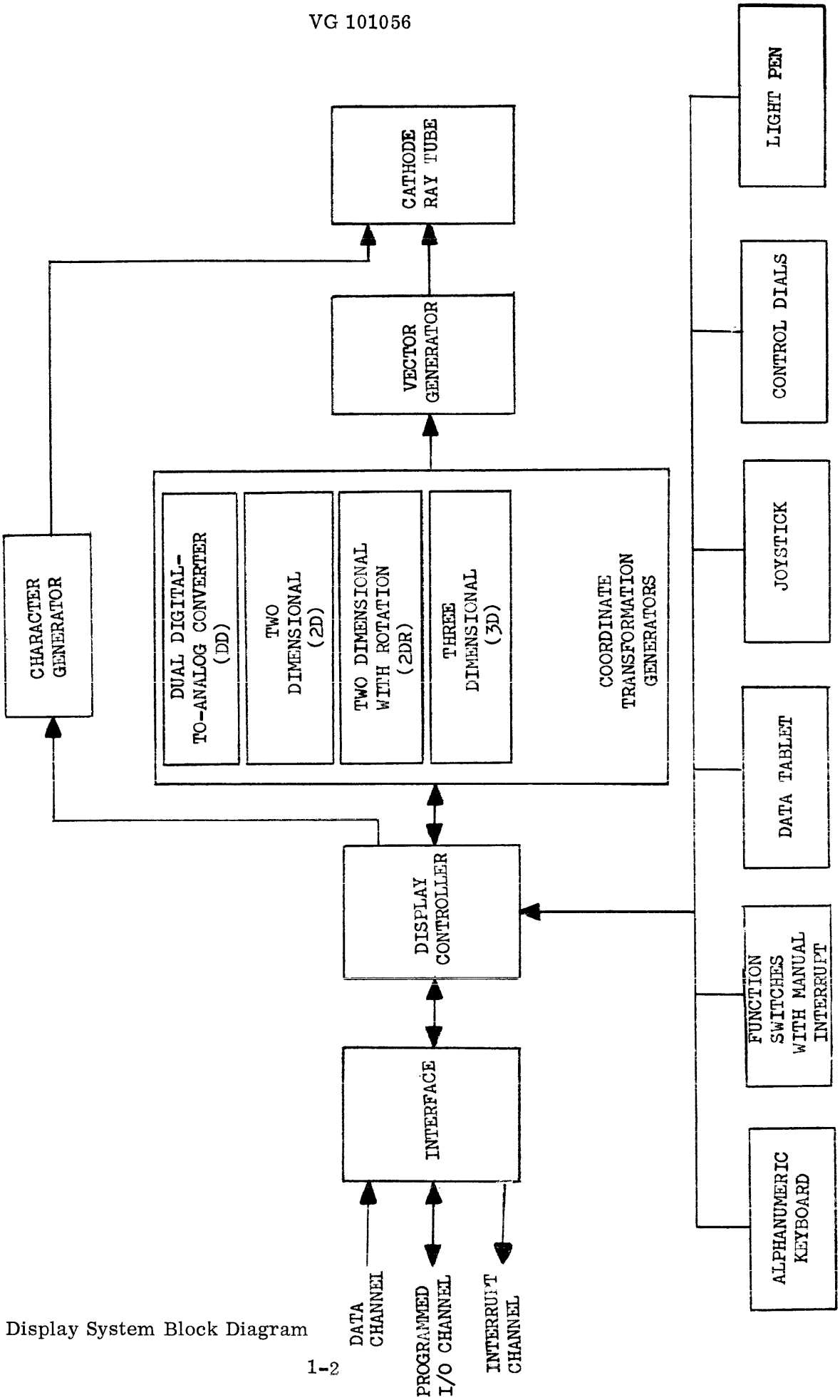


Figure 1-1. Display System Block Diagram

These channels are as follows:

- Data Channel - Direct memory access channel used to output the picture being presented on the CRT screen
- Programmed Input/Output Channel - Used to start the controller, acknowledge interrupts, and provide access to the display controller and device registers
- Interrupt Channel - Used by display and device response interrupt to activate computer programs

The display controller processes all display functions, running asynchronously with the computer central processor. The controller also receives inputs from the external control devices.

The dual digital-to-analog converter (DD) is the standard version of the coordinate transformation generators. It converts the digital values from the display controller into analog signals for use in the vector generator.

The vector generator accepts input from the coordinate transformation generators and uses it to present solid, dashed, or dotted lines between two positions on the display screen or to place a point at any given position.

The cathode-ray tube generates an electron beam that shows as a spot of light on the face of the tube. An electromagnetic deflection system causes the spot to move in any direction on the tube face in response to signals from the vector generator. An input from the vector generator causes the brightness of the spot to vary and turns the spot off completely when desired.

1.4 OPTIONAL FEATURES

The character generator processes a data stream of ASCII* characters and generates the characters as text for the display. Any one of four sizes may be selected by the program.

The circle/arc generator provides the facilities to draw arcs and circles in response to a set of programmed instructions.

---

* American National Standard Code for Information Interchange.

The following coordinate transformation generators are available:

- Two-dimensional for Scale and Translation (2D) - Scales and translates two-dimensional constructs and displays them. Scaling changes the size of image portions, and translation moves an image portion along one or both of its axes.

- Two-dimensional for Scale, Translation, and Single-axis rotation (2DR) - Scales and translates two-dimensional constructs and displays them with rotation in a single plane.

- Three-dimensional (3D) - Generates three-dimensional constructs and displays them with scaling, translation, and rotation about any axis.

The external control devices provide the display controller with inputs that can be used by the computer programs.

1.5 SYSTEM SPECIFICATIONS

Table 1-1 lists the general specifications for the display system.

1.6 DISPLAY PRINCIPLES

A cathode-ray tube display is a visible pattern on the face of a cathode-ray tube formed by a fluorescent spot moving on a screen inside the tube. To present a clear image, the pattern traced on the tube is repeated at 30 to 60 times a second. Any such repetition is called a "frame" and the frequency at which it is generated is called the "refresh rate".

The Vector General display uses the random scan method of controlling the movement of the spot. Random scan control involves steering the spot in a straight line between two points on the display screen. A series of these straight lines constitutes an image portion. All these directed lines are defined between the previous position of the spot on the screen (the starting point) and the position currently specified by the program (the end point).

Table 1-1. Display System Specifications

Feature	Characteristic	Specification
Interface and Controller	High-Speed I/O Channel	
	Access	Direct memory from CPU
	Word	16 bit
	Arithmetic	Parallel two's complement
	Addressable registers	50 destination; 72 source
	Vector formatting	Absolute, relative, short incremental, long incremental, autoincrement
	Register operations	Load, add, AND, OR
	Controls	Frame clock, vector generator, coordinate transformation generator, character generator, interactive devices
	Channels	1 per controller
	Programmed I/O Channel	
	Operations	Register read Interrupt handling Interactive device input
	Channels	1 per controller

Table 1-1. Display System Specifications (Cont.)

Feature	Characteristic	Specification
Interface and Controller (Cont.)	Interrupt Multiplexing	Priority interrupts multiplexed in controller
	Levels	1 CPU level per controller
Coordinate Transformation Generators	Dual Digital-to-Analog Converter (DD) Speed	1.5 μs per coordinate pair
	Two Dimensional (2D) Speed	1.5 μs per coordinate pair 5 μs coefficient settling
	Two Dimensional with Rotation (2DR) Speed	2.5 μs per coordinate pair 5 μs coefficient settling
	Three Dimensional (3D) Speed	2.5 μs per coordinate triple 5 μs coefficient settling
CRT - Vector Generator†	Tube shape	21 inches rectangular
	Display area	21-inch tube: 13 inches high, 14 inches wide
	Deflection type	Dual electromagnetic
	Spot size	0.020 inch - optional .010 inch
	Phosphor protection (PP1)	Hardware
	Brightness	50 foot-Lamberts††

†These specifications apply to a 10-inch x 10-inch precision area within the display area on the screen for the 21-inch CRT.

††Based on a 50-kHz signal applied to produce a 10-inch x 10-inch flat face raster with P40 phosphor.

Table 1-1. Display System Specifications (Cont.)

Feature	Characteristic	Specification
CRT - Vector Generator (Cont.)	Contrast	4:1
	Intensity levels	32
	Intensity modulation	Optional on 3D models
	Dynamic range	30 inches x 30 inches on 21" CRT
	Repeatability	20 Mils
	Drift	max= .05" over 8 hr. period after 30 min warmup
	Linearity	1%
	Addressable locations	4096 x 4096
	Positioning accuracy	2%
	Spot jitter	0.05% peak to peak
	Drawing speed *	
	21" high speed tube: (21DM3)	For "move":
	Vectors longer than 0.25 inch	[(L - 0.25) (0.7) + 2] μs * For "draw": [(L - 0.25) (1.5) + 2] μs * For "point": [(L - 0.25) (0.7) + 2.5] μs * where L = longest component in inches
	Vectors shorter than 0.25 inch	2 μs *
	21" medium speed tube: (21DM2)	For "move":
	Vectors longer than 0.25 inch	[(L - 0.25) (1.6) + 5] μs * For "draw": [(L - 0.25) (3.0) + 5] μs * For "point": [(L - 0.25) (1.6) + 6] μs * where L = longest component in inches
	Vectors shorter than 0.25 inch	5 μs *
	End matching	0.020 inch
	End closure	0.020 inch

* Refer to Coordinate Transformation Generator Specifications (Pg. 1-6) for array settling times. Timing is for units shipped after 4-1-73.

Table 1-1. Display System Specifications (Cont.)

Feature	Characteristic	Specification
CRT - Vector Generator (cont.)	Scissoring	Hardware
	Modes	Dot, dash, point, solid, dot/dash
Control Devices	Alphanumeric Keyboard (KB1)	70 keys including cursor function
	Function Switches (FS1)	16 momentary 1 interrupt
	Function Switches (FS2)	32 momentary 1 interrupt
	Data Tablet (DT1)	
	Size	11 inches x 11 inches
	Resolution	0.1%
	Control Dials (CD1)	
	Number	10
	Type	Single turn
	Function	Programmatic
	Light Pen (LP2) (LP3)	
	Type	Solid state
	Response time	3μs (1μs optional)
	Joystick (JS1)	3-axis electromechanical
Character Generator (CG1)	Type	Draw
	Character set	96 extended ASCII
		96 specials
		32 optional specials
	Aspect ratio	3:2
	Writing time	
	21" High Speed	7.5 μs, average
	21" Medium Speed	10. μs, average
	Cursor	Hardware
	Sizes	4
Circular Arc Generator		Drawing time*
(CAG1 on DD2)	21" High Speed tube	(r = radius)
	0-1 inch radius	[(0.028)(deg of arc)+8]μs
	1-12.8 inch radius	[(0.006 r)(3.0)(deg of arc)+8]μs
(CAG1 on DD3)	21" High Speed tube	
	0 - 1 inch radius	[(0.028)(deg of arc)+5]μs
	1 - 12.8 inch radius	[(0.006 r)(1.5)(deg of arc)+5]

* Based on high speed vector generator.

1.7 VISIBLE SPACE

That rectangular portion of the CRT which can be viewed by a user will be called the "Visible Space". The "Visible Space" is limited by an opaque mask with a rectangular cutout. See Figure 1-2.

The picture being generated is adjusted in size (scaled) to present the desired output by means of two controls:

a. The program controlled "Picture Scale" (PS) register in the transformation hardware (not available on standard DD system).
b. The manually adjustable "gain-controls" on the CRT deflection hardware.

The picture can be generated on a "Picture Space" coordinate system and scaled for viewing through the "Visible Space".

The maximum size "Picture Space" is larger than the "Visible Space". This permits limited "zooming" but primarily allows fully visible objects to be rotated and positioned to the extreme limits of the "Visible Space" and yet draw any remaining visible portions without distortion.

For the 21" CRT with the gain knobs at standard midrange calibrated settings, the maximum "Picture Space" (over which the vector generator accurately reproduces images) is a 30" x 30" plane of which the "Visible Space" (CRT screen visible throughmask) is a 13" by 14" rectangle in the center. (See Figure 1-3).

1.8 PICTURE SPACE

The hardware transformation options permit the coordinates defining an image portion to be transformed prior to use for display generation. The transformed coordinates used for display will describe a rotated and translated instance of the image portion.

For the input coordinates (X, Y, Z) the output transformed X and Y are used to generate the image portions' horizontal and vertical "Picture Space" position respectively. Thus, the "Picture Space" is the X-Y projection of the transformed image definition space (or (X, Y, Z) "Image Space").

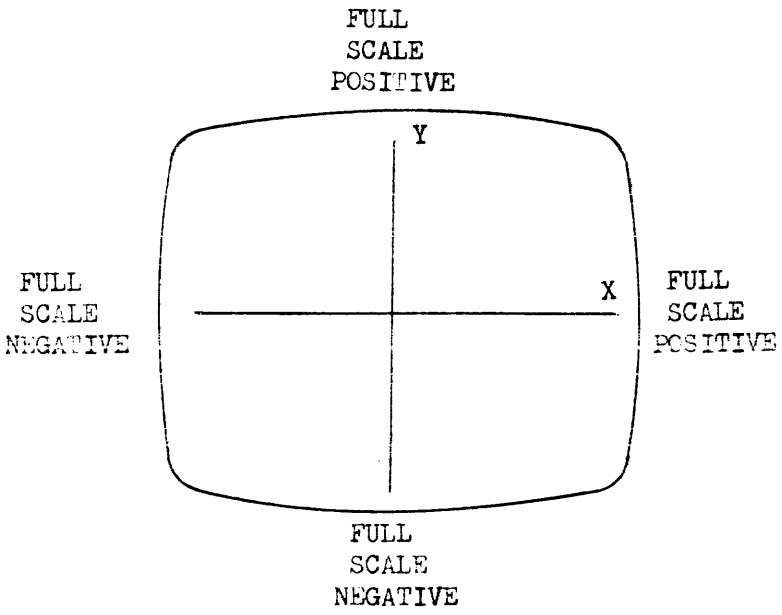


Figure 1-2. Visible Space

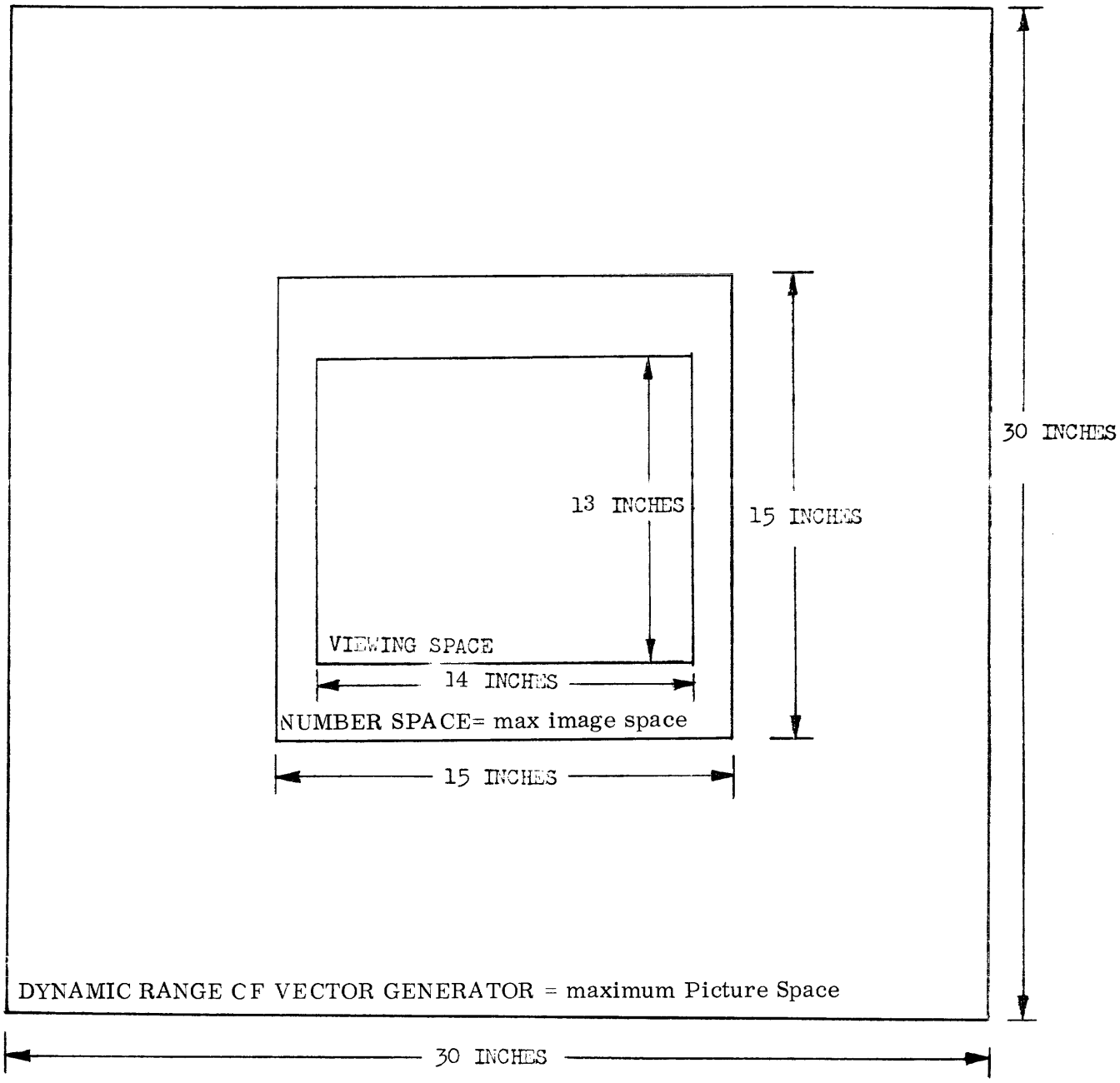


Figure 1-3. Image Areas, 21-Inch Display

If no transformation is performed, or for zero rotations, zero displacements, and full scale size transformation, an image coordinate (X, Y, Z) will correspond directly to the "Picture Space" (X, Y), with positive X being horizontal towards the right of a viewer and positive Y being vertical. With the gain knobs at the calibrated settings, and the Picture Scale register (PS) set to maximum, a plus full scale X image coordinate value transforms into an X Picture Space coordinate value which corresponds to a horizontal displacement 7.5" to the right of center or 1/2" to the right of the Visible Space. Similarly, for no transformation and maximum Picture Scale (PS), a full scale Y image coordinate value corresponds to a Picture Space position 7.5" up from the center.

To view a centered two-dimensional object defined over the entire X-Y coordinate range (such as a page of text), the Picture Scale register can be loaded with .92 or the gain knobs turned down (as required on a DD system). To view an entire centered rotated two-dimensional object, an additional factor of $1/\sqrt{2}$ picture scale is needed (not required on DD system since it does not implement rotation). To view an entire centered three-dimensional object which is defined over the entire (X, Y, Z) Image Space, an $1/\sqrt{3}$ factor is needed to view the maximum length of the projected diagonals of the Image Space.

Due to the larger range of the Picture Space over the Visible Space, each of these views may be positioned out of the viewing area in any direction without distorting any remaining visible portions. This capability is termed the "Hardware Scissoring Facility".

1.9 IMAGE SPACE

Prior to transformation and projection onto the Picture Space for viewing through the Visible Space, an object is defined in a coordinate system which we will refer to as the Image Space. All separately transformed objects of a displayed picture are defined in their respective untransformed image spaces.

To exploit maximum use of transformation ranges and coordinate resolution, all objects should be defined as large as possible in their defining Image Space. Objects are defined primarily in terms of generated visual elements; Vectors and Characters.

In cases where efficient interactive modification, dynamic model presentation, or motion is desired, an object definition may contain as elements "subimage calls" to generate transformed instances of other objects. In these cases, a composite Transformation of the existing transform with that of the called instance must be loaded into the hardware prior to processing elements of the called object definition for display generation. This permits nesting of transformable object definitions which can be directly processed for display.

In addition to the programmable linear vectors, the display system produces sets of ASCII characters generated independently of the computer program by a character generator.

1.10 ABSOLUTE VECTORS

The coordinates of absolute vectors are specified with respect to the zero position in the center of the Image Space (or screen for no transformations). Each new input data value is located directly on the Image Space grid as shown for a two-dimensional vector in Figure 1-4.

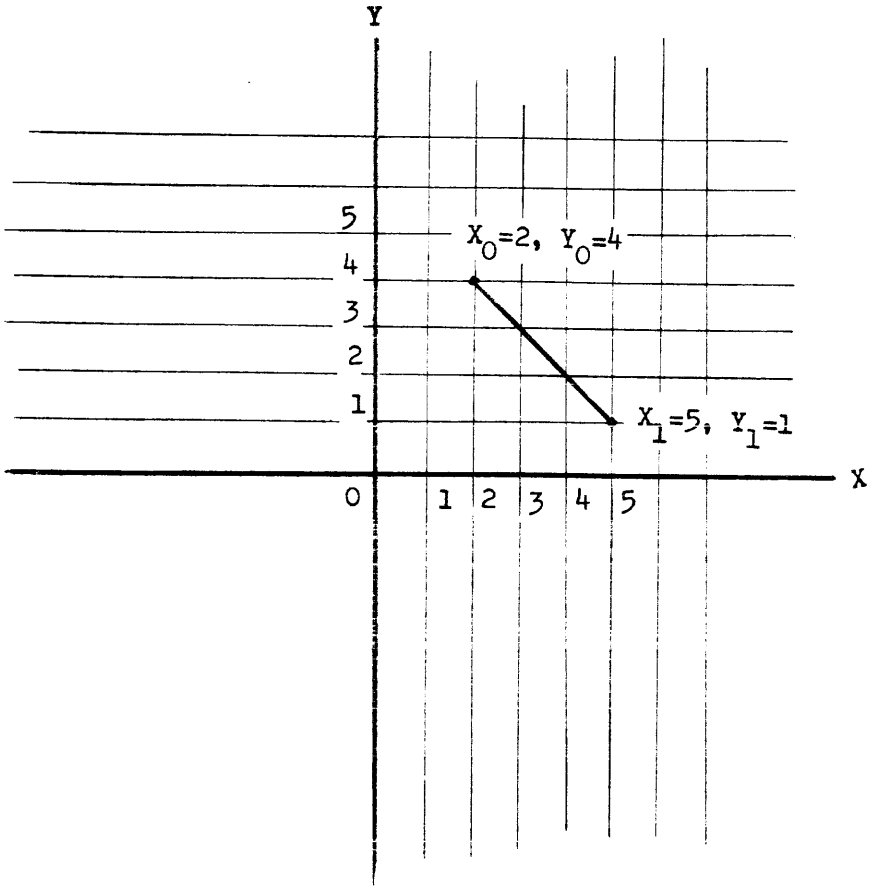


Figure 1-4. Absolute Vector

1.11 RELATIVE VECTORS

The end-point coordinates of a relative vector are located with respect to the starting point coordinates. In other words, relative vector data is specified in the form of increments that are added to or subtracted from the previous coordinate values as shown in Figure 1-5. An entire image construction can be positioned by drawing an initial absolute vector and defining the rest of the image with relative vectors without computing new end-point coordinates. This is an effective means of (unscaled and unrotated) subimage calling when no transformation hardware is available.

1.12 INCREMENTAL VECTORS

Incremental vectors are used when data storage is limited. Data increments can be shorter than relative vector increments, with a resultant reduction in the amount of data needed. Incremental vector display, therefore, requires less data storage and improves performance by increasing the rate of output and presentation. For coarse resolution, increments are added to the high-order end of the previous coordinate values; for fine resolution the increments are added to the low-order end.

1.13 AUTOINCREMENTING

The autoincrementing feature is used to step one coordinate at regular intervals while the other coordinate is open to program change, as shown in Figure 1-6. This feature, used for graphs and similar presentations, decreases memory requirements by 1/2.

1.14 THREE-DIMENSIONAL DISPLAY

Three-dimensional presentation involves the addition of a third, or Z, axis that is perpendicular to the face of the screen and intersects the X and Y Picture Space axes at the zero point as shown in Figure 1-7. The Z axis represents depth into and out of the display screen. The illusion of depth may be achieved by varying the light intensity of the fluorescent spot in proportion to the value of the Z coordinate. The intensity increases exponentially with the value from minus full-scale to one-half full-scale intensity, with maximum intensity at the face of the screen. For Z values much less than zero or greater than one-half

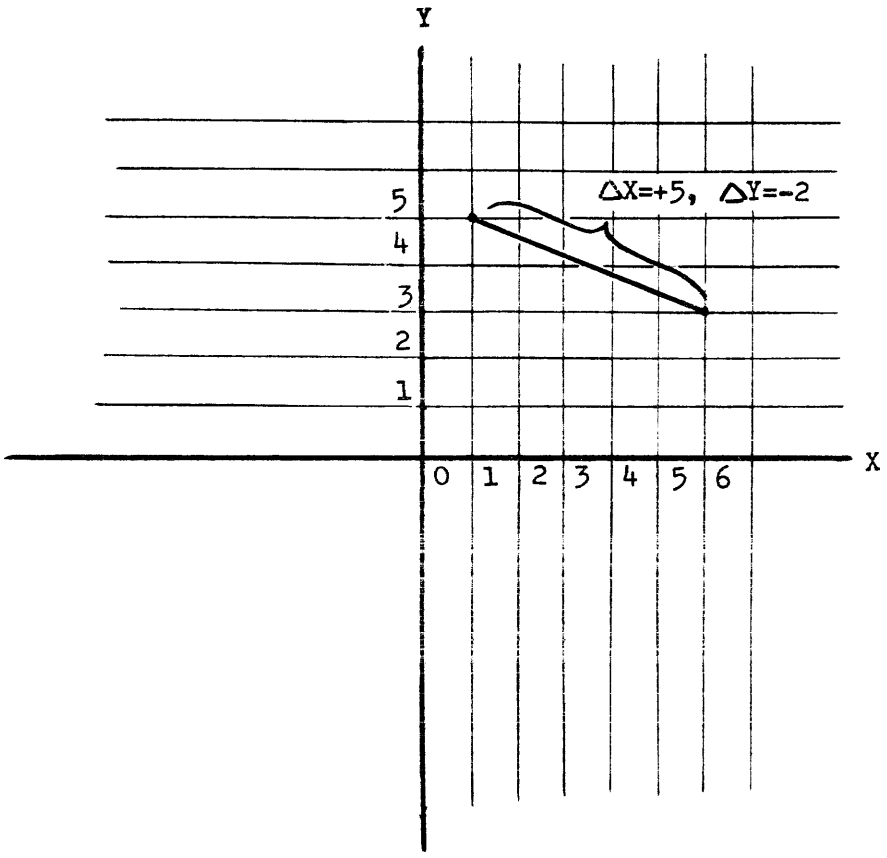


Figure 1-5. Relative Vector

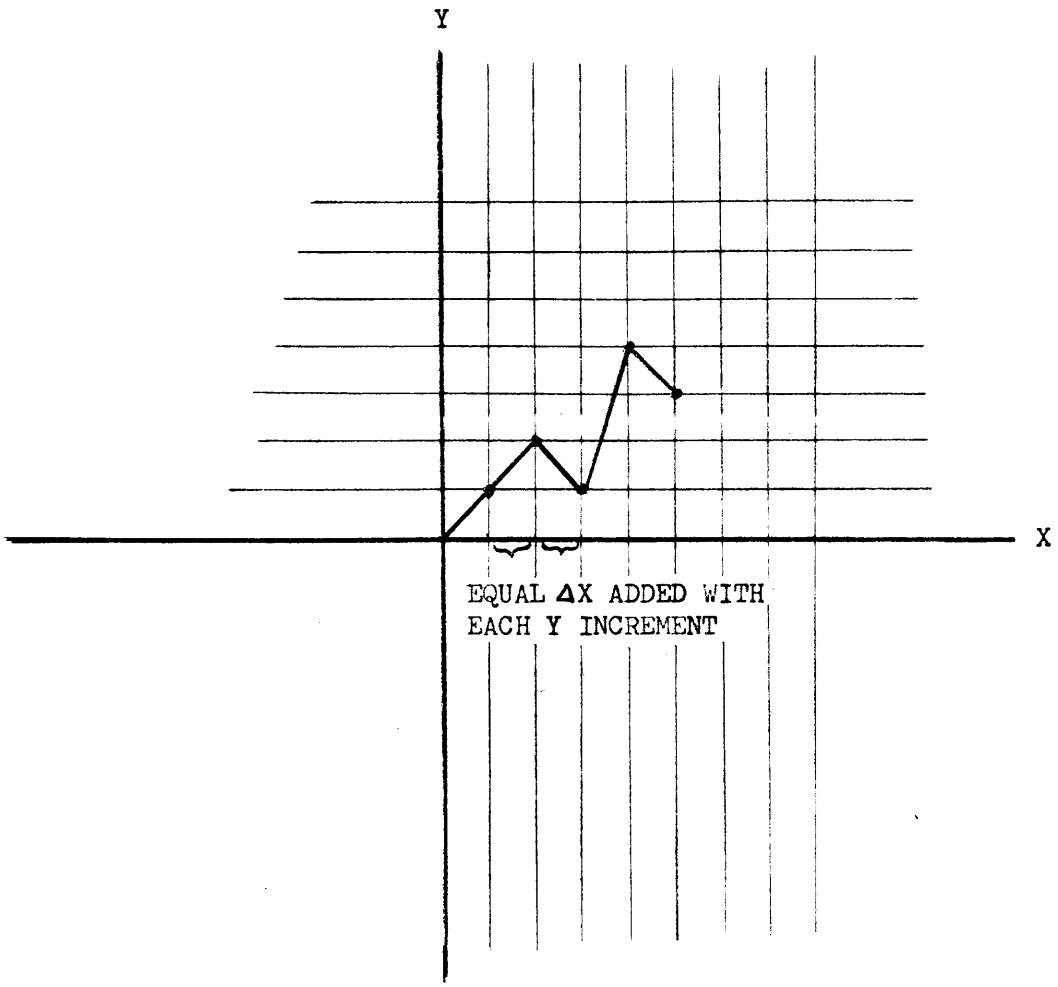


Figure 1-6. Auto-incrementing

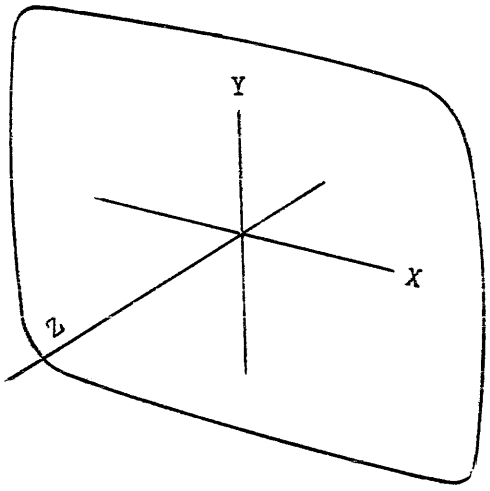


Figure 1-7. X, Y, and Z Axes

full scale, the intensity is zero; that is, the spot is turned off or blanked.

1.15 CHARACTER GENERATION

The character generator accepts coded inputs from the display controller and produces text strings composed of ASCII characters and special characters. Characters are drawn on the screen as a series of short vectors and curves. Unlike the vector generator, however, the character draws are generated automatically by the character generator each time a character code is received.

The program can select one of four character sizes and one of 32 intensity levels. A character scaling option is available for continuous character sizes. This option allows Picture Scale and Coordinate Scale to scale the image and characters proportionately. The program also can specify whether the text lines are to be displayed horizontally on the screen or are to be positioned as if on a page that has been rotated 90^{o} counterclockwise. One of the characters is a cursor, which differs from other displayed characters in that the character following the cursor is drawn in the same place, without a column feed. This feature permits the cursor to be moved over the screen as desired with manual inputs. A hardware feature causes the cursor to blink twice per second.

1.16 CHARACTER FORMATION

The character generator uses the function method of drawing characters, rather than the raster or scanning method sometimes used in display systems. The functional approach involves steering the fluorescent spot through a sequence of strokes to create character shapes. The characters are composed from a set of basic image elements, or draw figures, as shown in Figure 1-8. Any ASCII character can be produced in three draws or fewer, a draw being defined as all or a subset of one of the four shapes illustrated. The spot is blanked while moving through

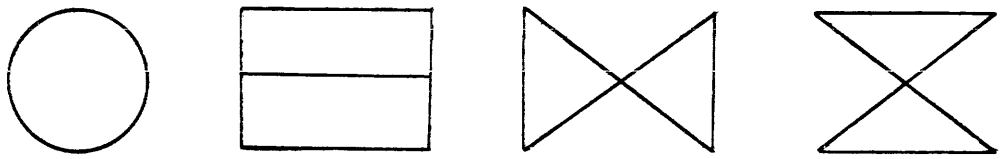


Figure 1-8. Draw Figure Definition

undisplayed sections of a character draw or from one character to another.

1.17 CONTROL CHARACTERS

Twelve codes in the character set are used for control purposes only and do not cause a display on the screen. The control characters and their functions are as follows:

- Null - Displays a blank in the corresponding character position. The spot is not stepped to the next character position
- Delete - Same as Null
- Backspace - Causes the spot to revert to the previous character position
- Line Feed - Causes the spot to move down to the corresponding character position in the line below
- Form Feed - Causes the spot to move to the position of the first character on the page; that is, Line 1, Column 1
- Carriage Return - Causes the spot to move to Column Position 1 on the line below
- DC1 - Causes the spot to move up to the corresponding character position in the line above. Equivalent to backline operation
- DC2 - Decreases the current character size by 1. This permits sub- and superscript sizes to be embedded in the text.
- DC3 - Increases the current character size by 1.
- DC4 - Terminates the data associated with a character generation display instruction.
- Horizontal Tab - Resets the current column position to "horizontal center" and increases the current line position by one line.
- Vertical Tab - Instates current character positioning to "horizontal center" of Line 1.

The first character in a string always starts at the location defined by the current X and Y coordinates.

1.18 INTENSITY LEVELS

For two-dimensional display, 32 constant intensity levels can be selected by the program. These intensities can be applied to vectors and characters.

The spot can be blanked as desired under program control during vector display.

Automatic blanking is an effect in the following operations:

- Presenting dashed or dotted lines between two positions on the display screen - The spot is alternately blanked and unblanked at appropriate intervals while a vector is drawn. The start and end of a vector are always unblanked.

- Placing a point at any given position. The spot is blanked while moving from one location on the screen to another and briefly unblanked at the end of the vector to form a point.

1.19 IMAGE TRANSFORMATION

Image transformation is an optional hardware feature that involves scaling, rotation, and position change (translation). The DD system has no transformation. The 2D system implements scaling and translation only. The 2DR system implements scaling and translation with rotation around the Z axis. The 3D system contains all the image transformation features, including rotation around any axis.

1.20 SCALING

The scaling operation consists of changing the size of an image portion by multiplying each end-point coordinate by the desired scale factor before processing. The scale factor is specified by the program, and the current scale factor is maintained in a hardware register to be multiplied by the X, Y, and Z coordinate values. An example of scaling is shown in Figure 1-9.

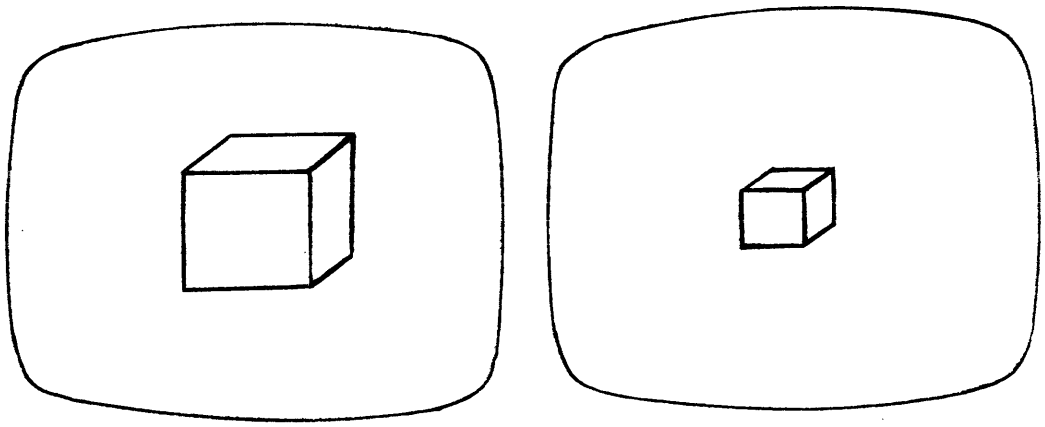


Figure 1-9. Scaling

1.21 ROTATION

An image portion can be rotated around any of its axes by using the optional hardware rotation matrix. The desired rotation is specified by loading direction cosines, or the sums of triple products of trigonometric functions in the more elaborate cases, into the rotation matrix, which has registers for each coordinate axis. The rotated image instance is automatically defined by a linear transformation of the coordinates of the unrotated master, using the direction cosines or the triple products which represent the angles between the coordinate axes of the two images. The 2D system rotation matrix contains only the registers necessary to rotate the X and Y coordinates around the Z axis. An example of rotation is shown in Figure 1-10.

1.22 TRANSLATION

An arbitrary image may be positioned anywhere in 3-dimensional space by adding a value to each of the scaled and rotated coordinate values every time an end point is specified. The value added must be constant for each coordinate to maintain the original image configuration. An example of translation is shown in Figure 1-11.

1.23 PICTURE TRANSFORMATION

When a 3-dimensional image made up of characters and vectors has been transformed to obtain the desired scale, rotation, and translation, a 2-dimensional view can be extracted and presented as a picture on the display screen. The two operations involved in this final presentation are picture scaling, to change the size of the transformed image, and intensity modulation (optional) to give a 3-dimensional depth cueing effect.

1.24 PICTURE SCALING

A hardware register is provided to hold a value that scales all the final transformed X and Y coordinate values. This scaling is used primarily to reduce full-scale, rotated, 3-dimensional images so that they fit into the display screen while permitting untransformed images, such as text pages and graphs, to fill the same display area.

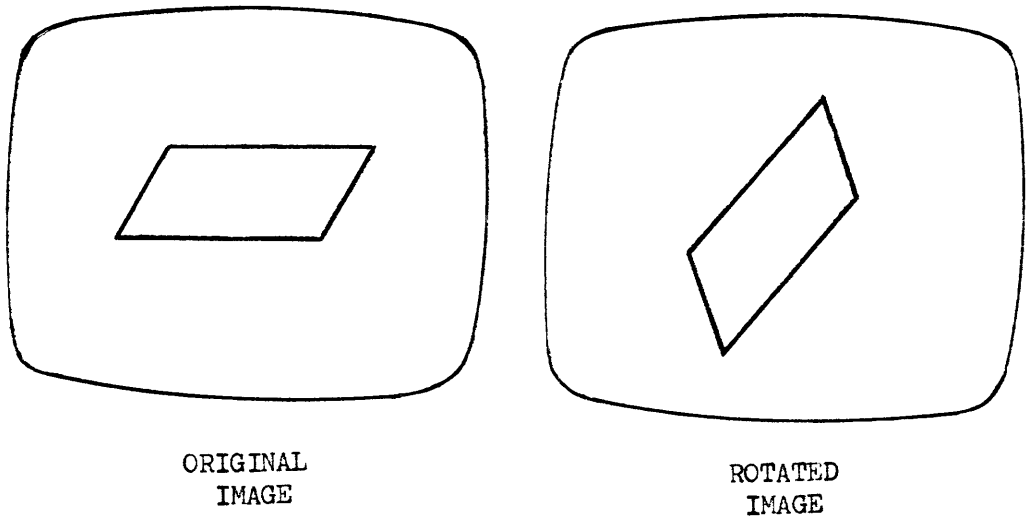


Figure 1-10. Rotation

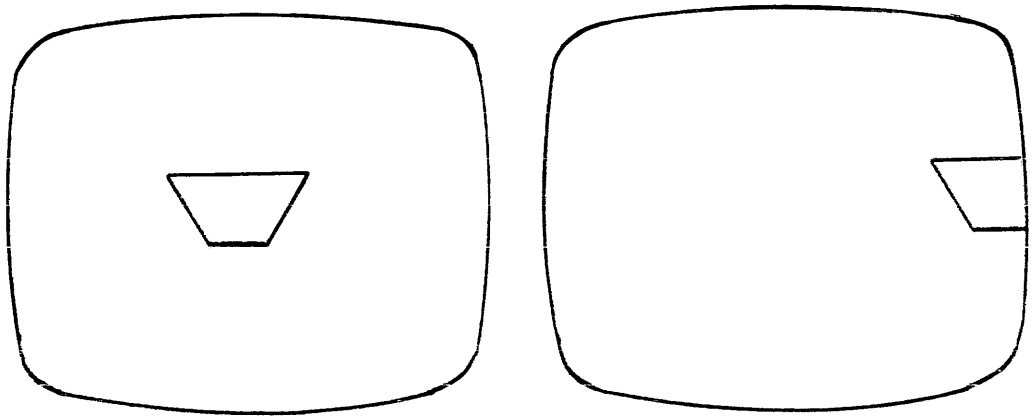


Figure 1-11. Translation

1.25 INTENSITY MODULATION

Intensity modulation is the name given to the depth cueing transformation used in all 3D systems, that shades the intensity of the displayed picture to give a 3-dimensional effect. The value of the transformed Z coordinate is used to represent depth into and out of the display screen, and therefore, controls spot intensity.

The picture transformation hardware includes the facility to blank any part of the picture that falls out of the screen towards the viewer. The cutoff plane can be moved toward or away from the viewer by the program so that sectional views may be obtained. This feature is an advantage when it is desirable to remove parts of cluttered images for clearer visibility.

SECTION II

SYSTEM ORGANIZATION

2.1 INTRODUCTION

This section contains a functional description of the system components, including the hardware registers. The optional control devices that may be used with the system are also described.

2.2 FUNCTIONAL DESCRIPTION

A functional block diagram of the CRT Display System showing the basic operational elements and data flow through these elements is shown in Figure 2-1. The basic elements can be grouped into the following functional sections: vector coordinate registers; coordinate scaling and displacement option; rotation option; picture control option; character generator; vector generator; cathode ray tube; and input/output facility.

2.3 VECTOR COORDINATES

The display system maintains the coordinates of the current position of the fluorescent spot in the 12-bit X- and Y- registers, with the inclusion of a 12-bit Z- register for the 3D option. These registers hold the X, Y and Z coordinates respectively. The values in these registers are updated as new coordinate values are received on the data channel.

When relative vectors are specified, a coordinate increment is received on the data channel and added to the current coordinate value. The sum is then loaded into the proper register. If incremental vectors are specified, the increment is added to either the high-order or low-order end of the current coordinate value, depending on the scale specified by the program.

2.4 COORDINATE SCALE OPTION

To change the overall size of the image without changing its shape, a

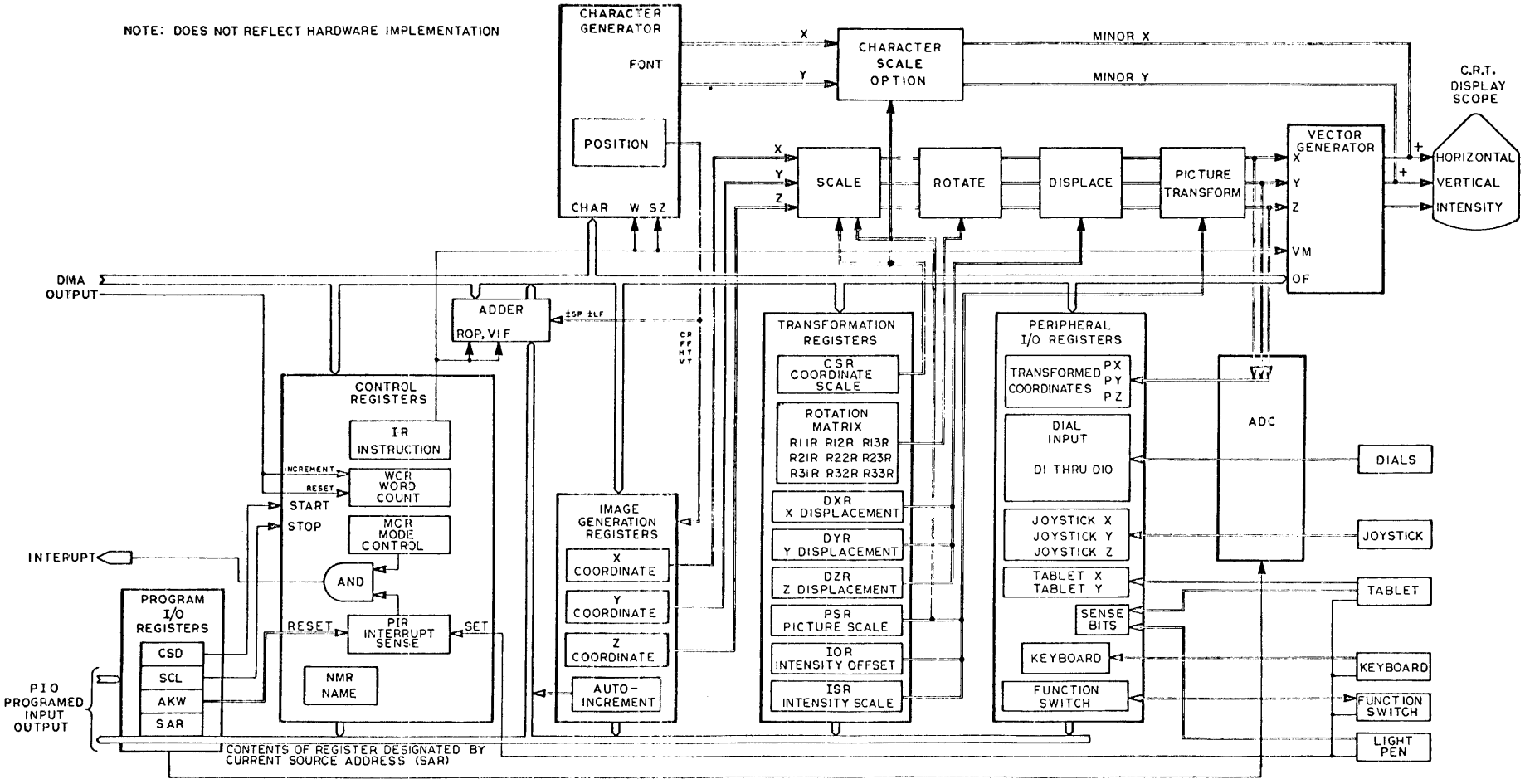


FIGURE
DISPLAY FUNCTIONAL BLOCK DIAGRAM

Figure 2-1

scale factor is loaded by the program into a 12-bit coordinate scale register. This number is multiplied by the current coordinate values from the X, Y, and Z coordinate registers. Characters also are scaled in proportion to the rest of the image with the character scale option.

2.5 ROTATION OPTION

To rotate the image around any of the three axes, trigonometric values are loaded into rotation matrix registers R11R through R33R. Registers R13R, R31R, R32R, R23R, and R33R are used only for 3D rotation. If the three scaled input coordinates are defined as X_0, Y_0, and Z_0 and the three computed outputs are X_1, Y_1, and Z_1, the following computation is performed:

$$X_1 = R11R \cdot X_0 + R12R \cdot Y_0 + R13R \cdot Z_0$$

$$Y_1 = R21R \cdot X_0 + R22R \cdot Y_0 + R23R \cdot Z_0$$

$$Z_1 = R31R \cdot X_0 + R32R \cdot Y_0 + R33R \cdot Z_0$$

The coefficients of an object after rotation may be continuously computed from the coordinates of the unrotated master by loading coefficients defining the desired rotation into registers R11R through R33R.

Figure 2-2 illustrates the effect on a point of two-dimensional rotation about the Z axis by the angle θ. The coordinates of the rotated point in terms of its original unrotated coordinates are as follows:

$$X' = X \cos \theta + Y \sin \theta$$

$$Y' = X(-\sin \theta + Y \cos \theta$$

$$Z' = Z$$

The values of R11 through R33 that would perform the illustrated rotation are as follows:

$$R_{11} = R_{22} = \cos \theta$$

$$R_{12} = \sin \theta$$

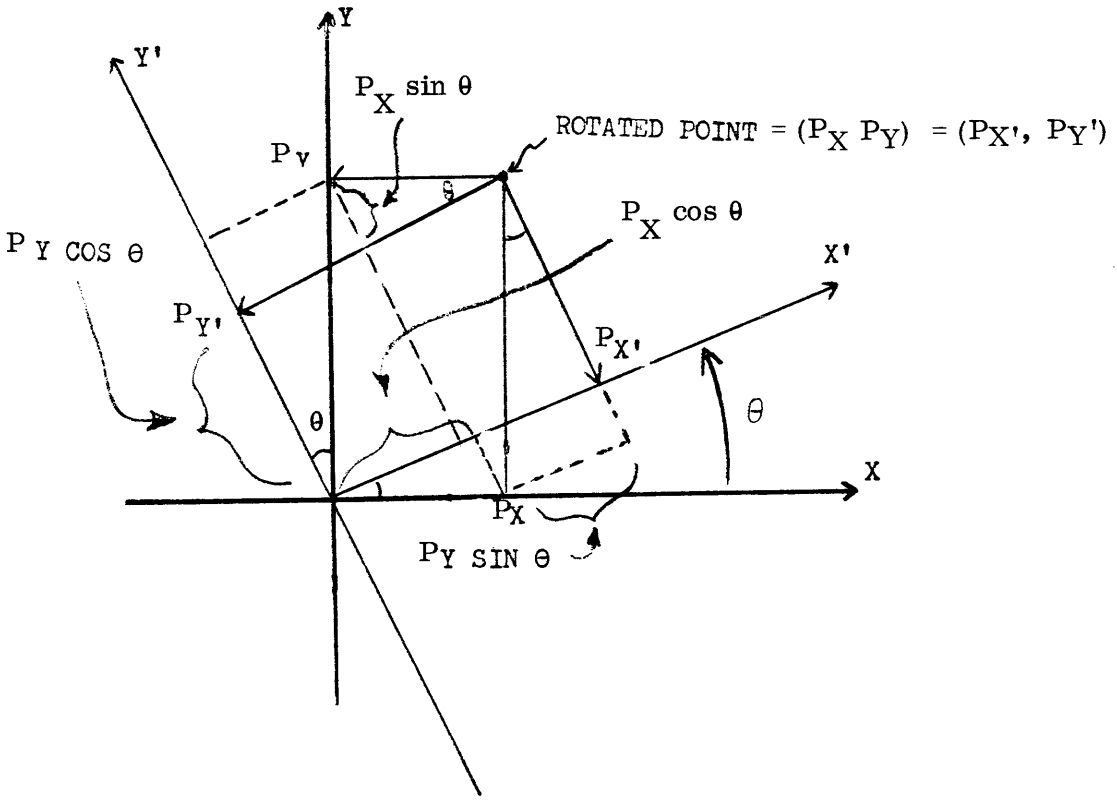
$$R_{21} = -\sin \theta$$

$$R_{33} = 1$$

$$R_{13} = R_{23} = R_{31} = R_{32} = 0$$

2.6 DISPLACEMENT VECTOR OPTION

The displacement vector option performs the translation function in the image transformation feature by moving the image intact along any of the three axes. The X-displacement, Y-displacement, and Z-displacement registers are used to implement this feature. A displacement constant loaded by the program into any one of these registers is added to the associated rotated coordinate values being maintained by the rotation, scale, and coordinate registers. The result is a displacement of the entire image along any axis whose displacement register is



$$P_{X'} = P_X \text{ COS } \theta + P_Y \text{ SIN } \theta$$

$$P_{Y'} = -P_X \text{ SIN } \theta + P_Y \text{ COS } \theta$$

Figure 2-2. Two-Dimensional Rotation

loaded. An example of the displacement operation in an X, Y plane is shown in Figure 2-3. The X displacement register contains a 2 and the Y displacement register contains a 3. The value 2 is added to each X coordinate and the value 3 is added to each Y coordinate as follows:

register contains a 3. The value 2 is added to each X coordinate and the value 3 is added to each Y coordinate as follows:

Original Position	New Position
$X_0 = 2$	$X_0 = 2 + 2 = 4$
$Y_0 = 2$	$Y_0 = 2 + 3 = 5$
$X_1 = 3$	$X_1 = 3 + 2 = 5$
$Y_1 = 1$	$Y_1 = 1 + 3 = 4$
$X_2 = 5$	$X_2 = 5 + 2 = 7$
$Y_2 = 2$	$Y_2 = 2 + 3 = 5$
$X_3 = 6$	$X_3 = 6 + 2 = 8$
$Y_3 = 1$	$Y_3 = 1 + 3 = 4$

2.7 PICTURE CONTROL OPTION

The picture control option is used for picture transformation after the transformation of individual images on the screen has been completed. The registers used for this feature are the 12-bit intensity offset register, the 12-bit intensity scale register, and the 12-bit picture scale register. The value in the picture scale register is multiplied by each of the transformed

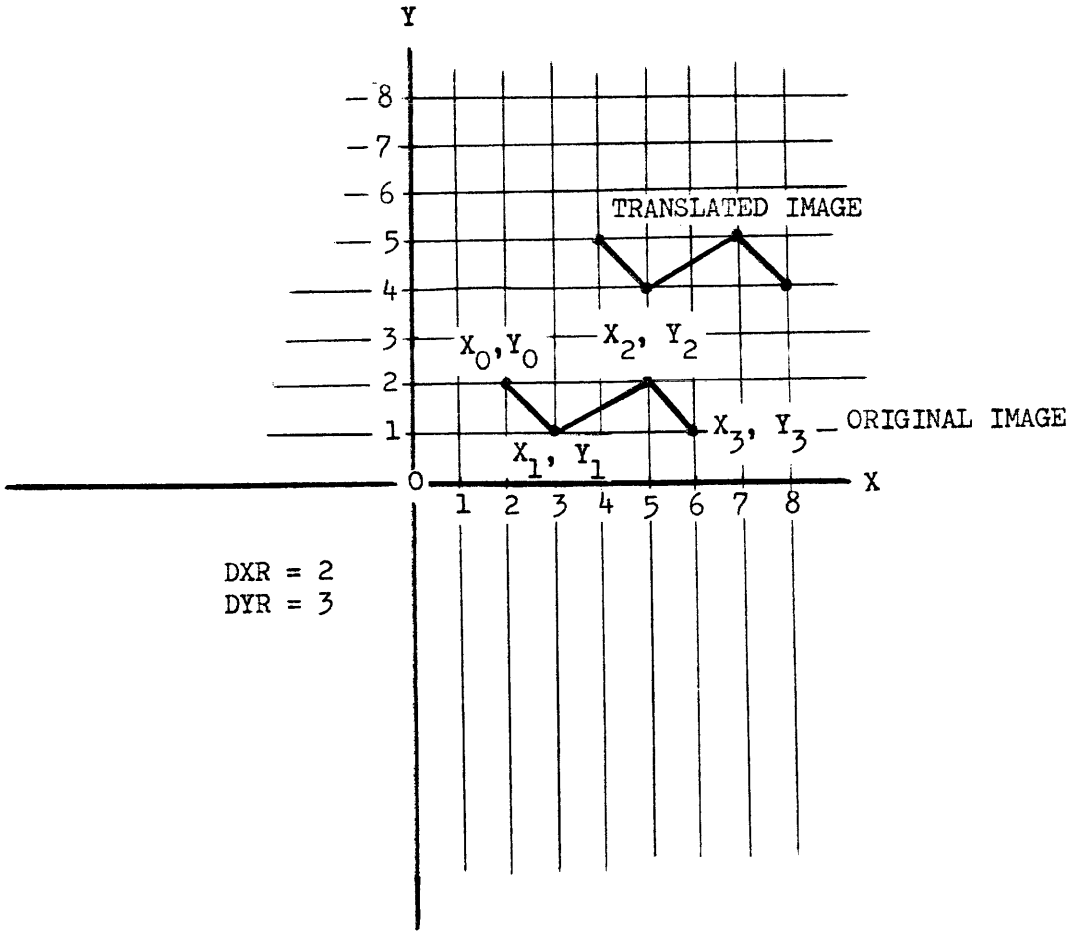


Figure 2-3. Two-Dimensional Translation

X, Y. and Z coordinates to establish the final picture size. This scaling applies also to characters in the picture, with the char-scale option.

In a two-dimensional system, the 12-bit intensity offset register (IOR) is loaded by the program to specify 1 of 32 intensity levels. Only the high-order five bits of the register are used for this purpose. The intensity levels apply to characters as well as to two-dimensional vectors. Full scale positive in the intensity offset register designates maximum intensity, and the intensity decreases exponentially as the value decreases to minus full scale.

In a three-dimensional system with the Intensity Modulation option, the intensity scale register is used in conjunction with the intensity offset register to provide depth cueing, or shading of the intensity of the picture according to the value of the Z coordinate. The intensity of the spot at any instant is represented by the following equation:

if IS_{sign}	= 0:	$I = I_{max} \cdot e^{k(Z')}$	when $Z' \leq .02$	
			when $Z' > .02$	saturation occurs
else if IS_{sign}	= 1:	$I = I_{max} \cdot e^{k Z'}$	when $Z' \leq .02$	
		$I = 0$	when $Z' > .02$	

where:

$$Z' = (1/2) * IS_{mag} * Z_{transformed} + IO$$

This equation provides for exponential shading of the intensity along the length of vectors drawn between coordinates of different intensity values. A "screen-cutoff" can be imposed by setting the sign bit of ISR; then if Z' is greater than .02, the intensity is 0 and the spot is blanked.

The intensity cutoff plane is established by the value in the intensity offset register. Within the depth range of an image, the intensity is blanked between the viewer and the screen. The intensity is at its maximum at the face of the screen and decreases exponentially with decreasing values of Z toward the back of the image. Figure 2-4 shows a simplified cross section of a CRT with a three-dimensional image in two different positions with respect to the intensity cutoff plane. As the value in the intensity offset register is changed, the image moves forward or backward through the intensity range, to vary the section that is intensified and the part that is blanked out.

The intensity range, or apparent depth of the image, is determined by the value in the intensity scale register. If the value is 1, the maximum intensity range is achieved. If the value is 0, the intensity is constant and the image has no depth-cueing. Figure 2-5 shows how a variation in intensity scale changes the depth of the image.

2.8 VECTOR GENERATOR

The vector generator accepts as inputs the transformed coordinate values and the display controller instruction. Two outputs from the vector generator move the floures-ent spot in a horizontal and vertical direction on the screen. A third output varies the intensity of the display. Programmed vector mode information is stored in the vector generator and used to provide blank and unblank inputs to the CRT to

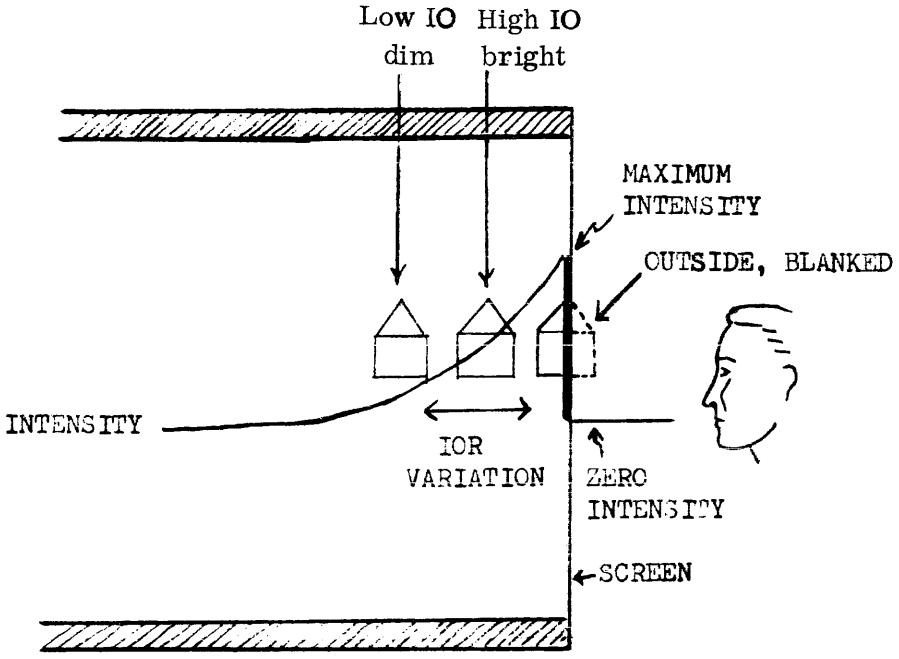


Figure 2-4. Effect of Intensity Offset Variation

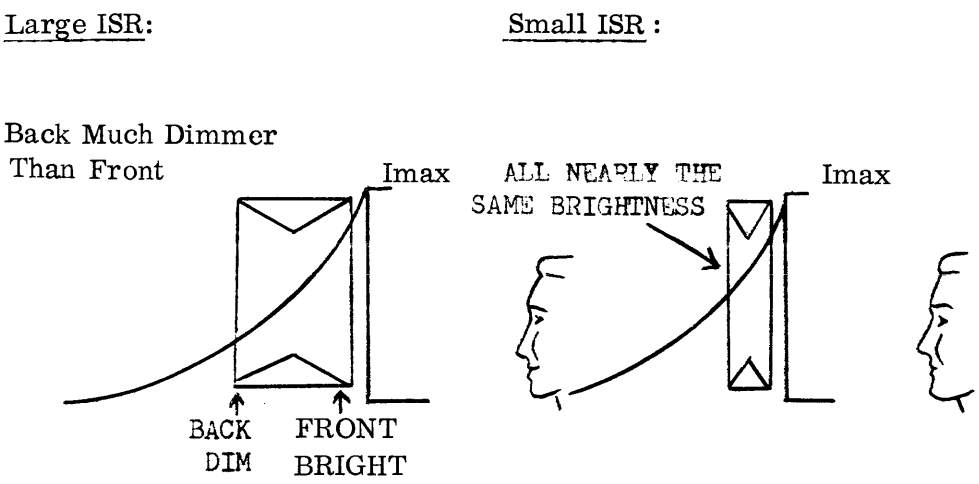


Figure 2-5. Effect of Intensity Scale Variation

specify lines, dashes, dots, or points. Vector operation information, also stored in the vector generator, determines whether the spot on the CRT will draw a vector, move from one location to another without drawing a vector, or remain stationary while new current coordinates are being received from the computer.

2.9 CHARACTER GENERATOR

The character generator interprets character codes received from the display controller and provides small X- and Y-axis deflection inputs to the cathode ray tube.

Inputs to the character generator are in the form of a stream of ASCII codes and information specifying size and the character fonts.

Character positioning signals from the character generator are sent to the adder for combination with the current X and Y coordinates to locate the starting point for each new character. Size information is decoded to control the minor deflection signals in four different ways to produce the four character sizes. Two-dimensional scaling inputs from the coordinate scale option and from the picture scale control option are used in the character generator so that character strings may be scaled and translated with their associated picture structures; that is, images and their labels may be transformed as a single construct.

The dimensions for character generator outputs in Number Space units are given in Figure 2-7.

The following picture shows the standard character set font. The codes for each character can be found in Appendix A.

2.10 CIRCULAR ARC GENERATOR

The arc generator accepts as inputs the transformed coordinate values and the display controller instruction. Two outputs from the arc generator move the fluorescent spot in a horizontal and vertical direction on the screen. A third output varies the intensification of the display.

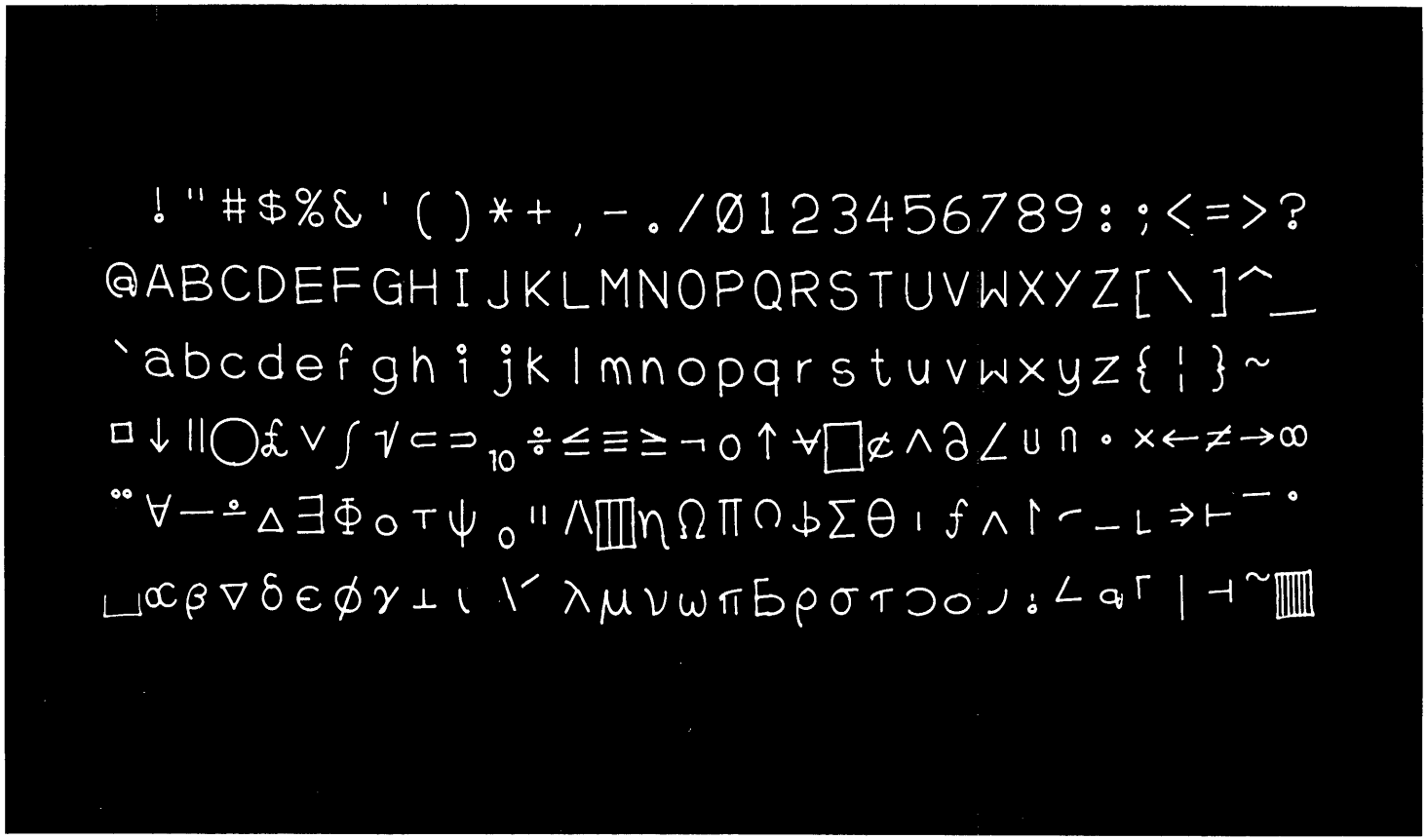
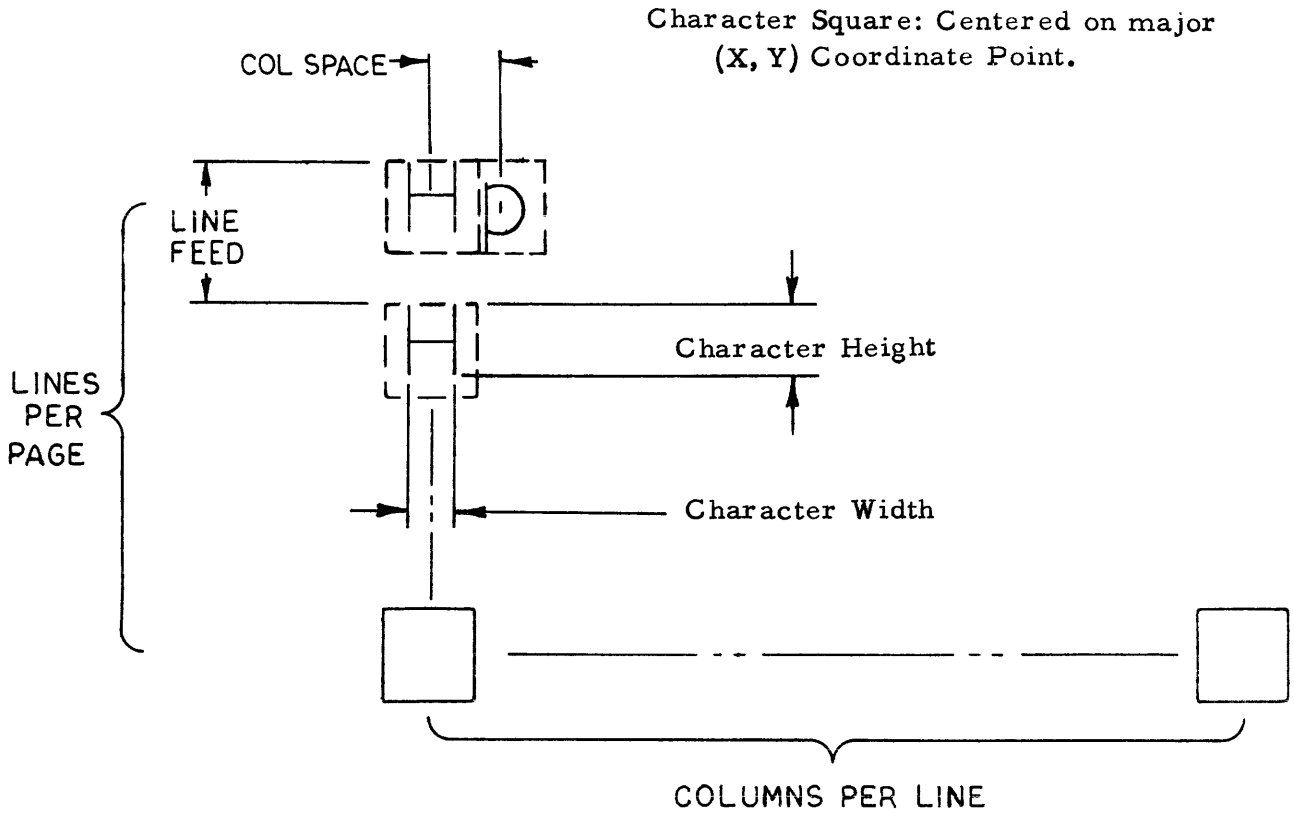


Figure 2-6. Character Generator Display



Char. Size	Cols./Line	Lines/Page	Col. Space* Size	Linefeed* Size	Character* Height	Character* Width
0 = (1X)	120	60	34	68	34	23
1 = (1.5X)	81	41	50	100	50	33
2 = (2X)	60	30	68	136	68	45
3 = (4X)	32	16	128	256	128	85

Character Size	Character Square	Col. Space Size **	Linefeed Size **	Character Height **	Character Width **
0 = (1X)		.125"	.250"	.125"	.083"
1 = (1.5X)		.183"	.366"	.183"	.121"
2 = (2X)		.249"	.498"	.249"	.165"
3 = (4X)		.469"	.938"	.469"	.311"

* in addressable points (base 10)

** with no transformation, full scale positive picture and coordinate scaling (not applicable to standard DD systems), and gains calibrated to a 7.5" center to side deflection for full scale X and Y image coordinate value.

Figure 2-7. Character Generator Outputs

The generated arcs are coded anywhere within any of the following types of vector lists:

Vector Relative
Vector Relative Auto-X
Vector Relative Auto-Y
Vector Relative Auto-Z
Vector Absolute
Vector Absolute Auto-X
Vector Absolute Auto-Y
Vector Absolute Auto-Z

Thus, arcs can have line texture (solid, dotted, dashed, or dash-dot-dashed) and can be mixed with vectors.

The arc generator draws arcs from the initial beam position to the given end-point (omitted for 360^{o} circle) about the following center-point.

The center and endpoints of the arcs are properly transformed in both two and three space, but the arcs are drawn in a plane parallel to the screen (as are characters).

Thus, all arcs are properly transformed by DD, 2D, and 2DR systems, and only rotatable about Z in 3D systems.

2.11 CATHODE RAY TUBE

The three inputs to the CRT are horizontal and vertical deflection, to control the movement of the fluorescent spot, and intensity, to control the brightness. The intensity input is received as two signals. One is an intensity level signal, and the other is an on/off blanking signal. The major deflection signals are received from the vector generator, and minor deflection inputs from the character generator are superimposed.

2.12 OPTIONAL CONTROL DEVICES

The functions of the interactive control devices that may be used with the CRT display are described below.

2.13 ALPHANUMERIC KEYBOARD

The alphanumeric keyboard is used as an entry device for manual input to the display system. Pressing a key on the keyboard enters an eight-bit character code into the keyboard register in the display controller and sets bit 12 (PIK) of the priority interrupt request register to indicate a keyboard interrupt condition. The PIK bit must be cleared by acknowledging the keyboard interrupt before another character can be entered. The character entered in the keyboard register does not directly affect the display on the screen. The program can read the keyboard register contents and use the information in its operation. One function of the program may be to place the character into a display list being presented on the screen. Holding any key down will maintain the correct code in the keyboard register and, after an initial delay, will repetitively raise PIK (keyboard interrupt request) to repeat any character. Appendix A lists the codes generated by the keyboard for shifted and unshifted key combinations. The following diagram illustrates the keyboard switch layout.

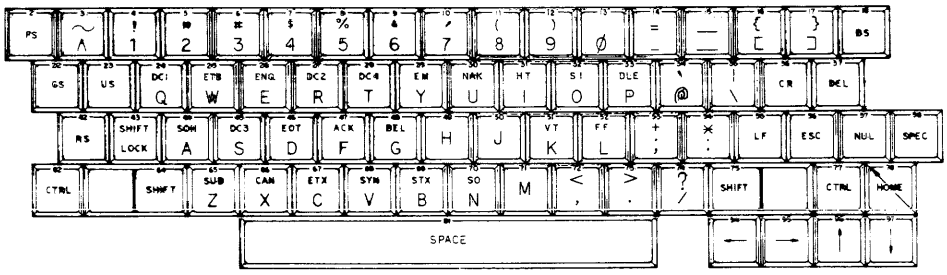


Figure 2-8. Keyboard Layout

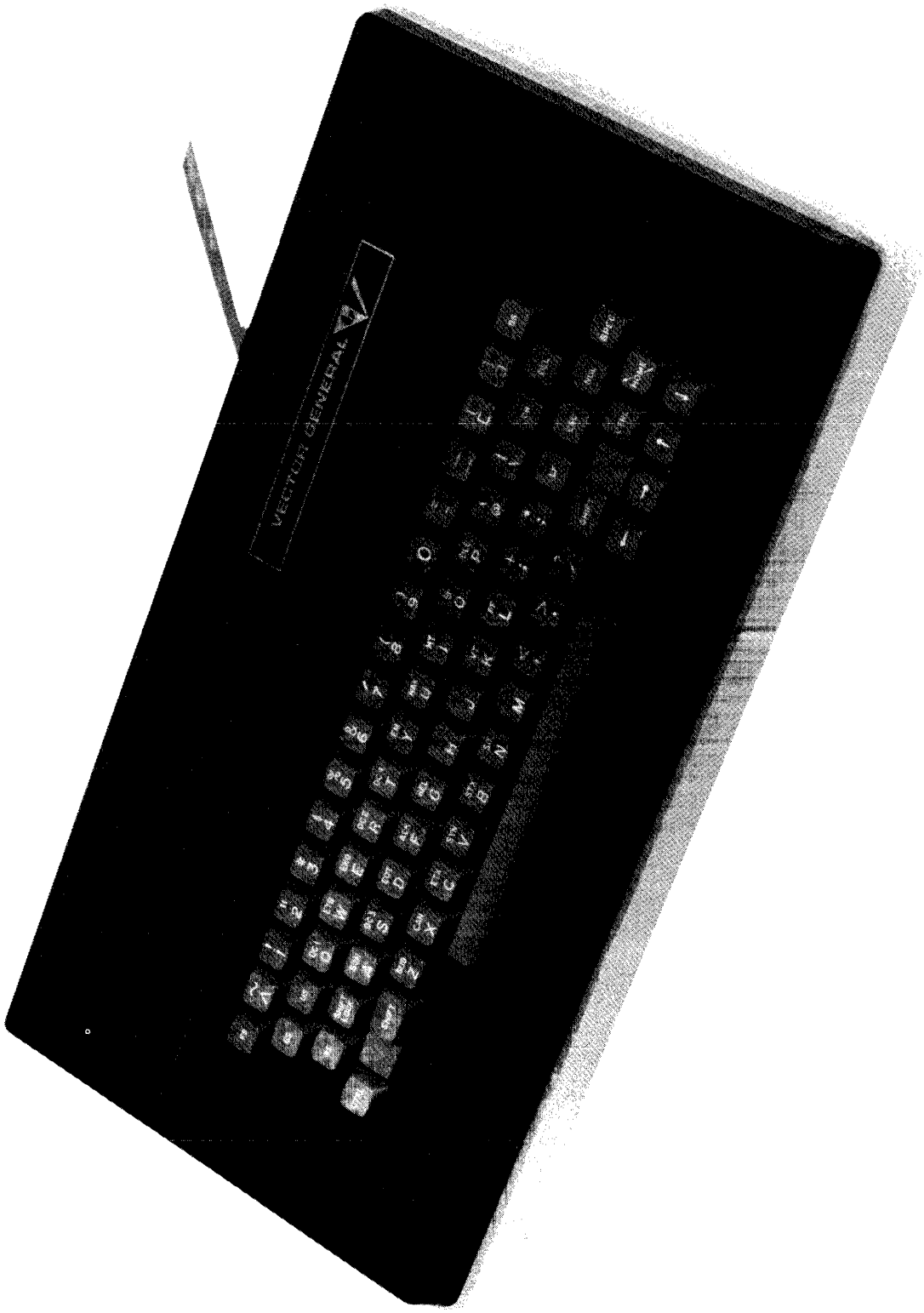


Figure 2-9. Keyboard - Top View

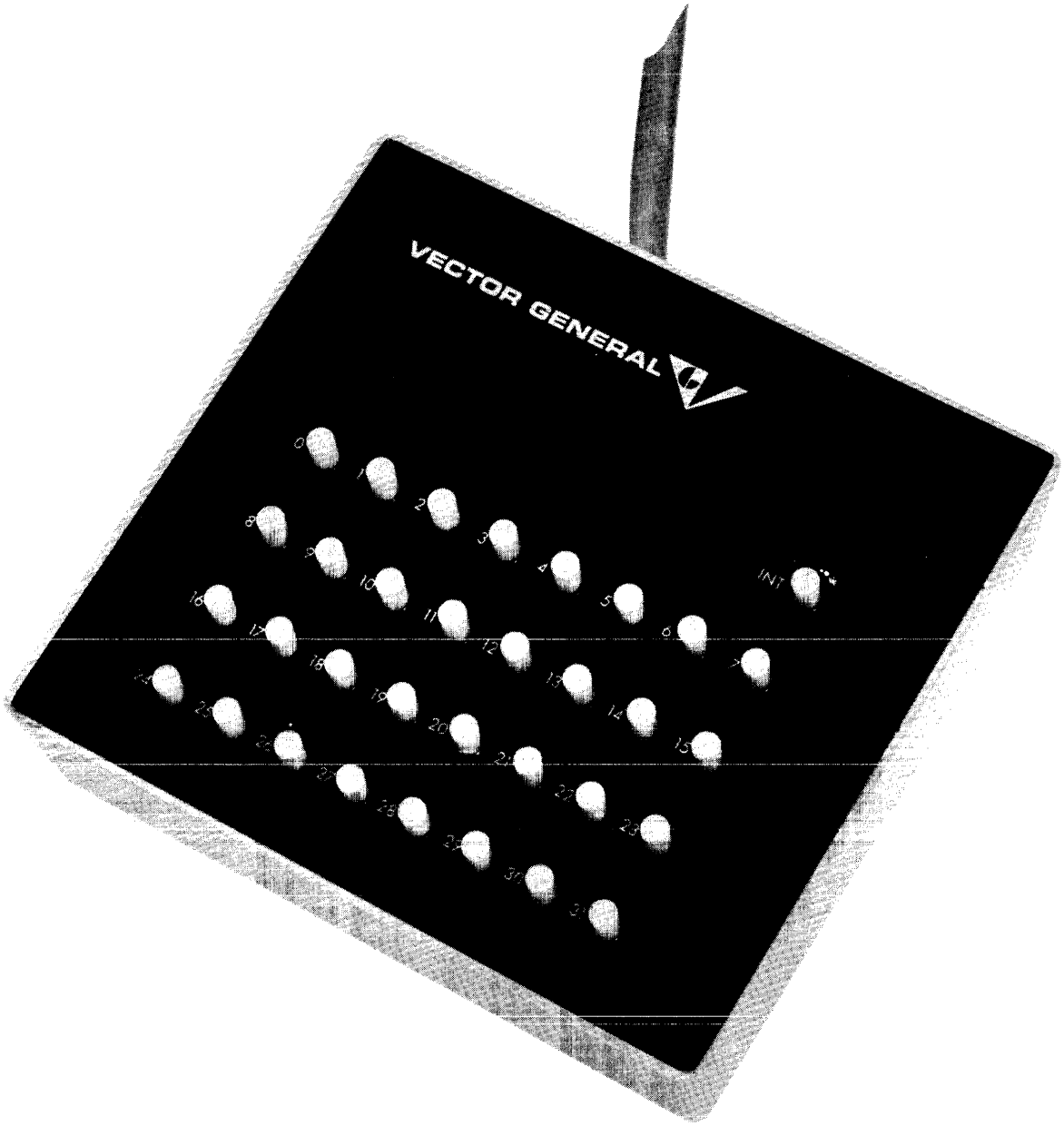


Figure 2-10. Function Switch Option - Top View

2.14 LIGHTED FUNCTION SWITCHES WITH MANUAL INTERRUPT

This device contains 16 or 32 function switches plus a manual interrupt switch. The function switch register in the display controller has one bit corresponding to each function switch; that is, bit 0 for function switch 1, bit 1 for function switch 2, and so on through bit 15 for function switch 16. While any function switch is depressed, the corresponding bit in the function switch register is set. The computer can then read the contents of the register and use them.

The manual interrupt switch can be used to cause an interrupt. This feature allows the operator to intercept the program at any desired point. When the manual interrupt switch is pressed, bit 13 of the priority interrupt request register (PIR) is set to indicate a manual interrupt condition.

The first 8 bits of the first two output-register addresses control the 16 function switch lights. Sending ones will light the corresponding light, and zeroes turn them off. Note: as with all display registers, ANDing and ORing operations permit independent manipulation of fields.

2.15 JOYSTICK

The joystick is a mechanical device used to enter coordinate values in the 12-bit joystick X, Y, and Z input registers. A forward or backward motion of the joystick increases or reduces the value for the joystick Y-input register. A motion from side to side changes the joystick X-input value. The joystick Z-input value are decreased or increased when the joystick is twisted in a clockwise or counterclockwise direction. All three motions have a spring return to an adjustable null center position. These input registers may be read by the computer, and, if desired, the joystick values may be added into the X, Y, and Z displacement registers to move the display accordingly. Note input values range at least over $\pm$ 1/2 F.S.

Figure 2-11. Joystick - Top View

2.16 LIGHT PEN

The light pen, a wand containing a photocell, is held near the face of the CRT by the viewer and used to point at and identify an existing element in the display or to create new information by "drawing" on the CRT. The position of the pen in the pattern can be continuously computed from the pen's response to the pattern and the coordinates can be maintained by the computer program.

When light from a line or point in the display is detected by the photocell, bit 10 (PIP) of the priority interrupt register (PIR) sets to indicate a light pen interrupt condition. In this manner, the program may identify the word which caused the interrupt by reading the display list word count register. The word count may be further resolved to the halfword field of packed-data and character words via the pen-byte-resolution (PB) field of the mode register (MCR). The option also contains a delay feature which inhibits proceeding to a new instruction when the pen generates an interrupt condition. The pen also contains a manually operated switch which sets bit 15 (SP1) of the PIR when activated by the operator.

2.17 DATA TABLET

The data tablet is a graphic input device containing an X-Y coordinate grid which corresponds to the grid on the CRT screen. Information is entered through the tablet with a pen-like stylus. The tablet circuitry senses the coordinate position of the stylus on the grid and loads these data into the tablet X and Y registers (TIX and TIY). Each register also contains 2 status bits which specify the following conditions:

TIX bit 14 (STN) = 1 when the stylus is less than 1/2 inch above the tablet;
0 when the stylus is higher than 1/2 inch above the tablet.

TIX bit 15 (STP) = 1 when the stylus pressure switch is activated;
0 when the stylus pressure switch is deactivated.

TIY bit 14 (XOS) = 1 when the stylus is moved into an X margin area;
0 when the stylus is moved out of the X margin area.

TIY bit 15 (YOS) = 1 when the stylus is moved into a Y margin area;
0 when the stylus is moved out of the Y margin area.

When any of the above 4 bits changes state, an interrupt is generated (PIT).

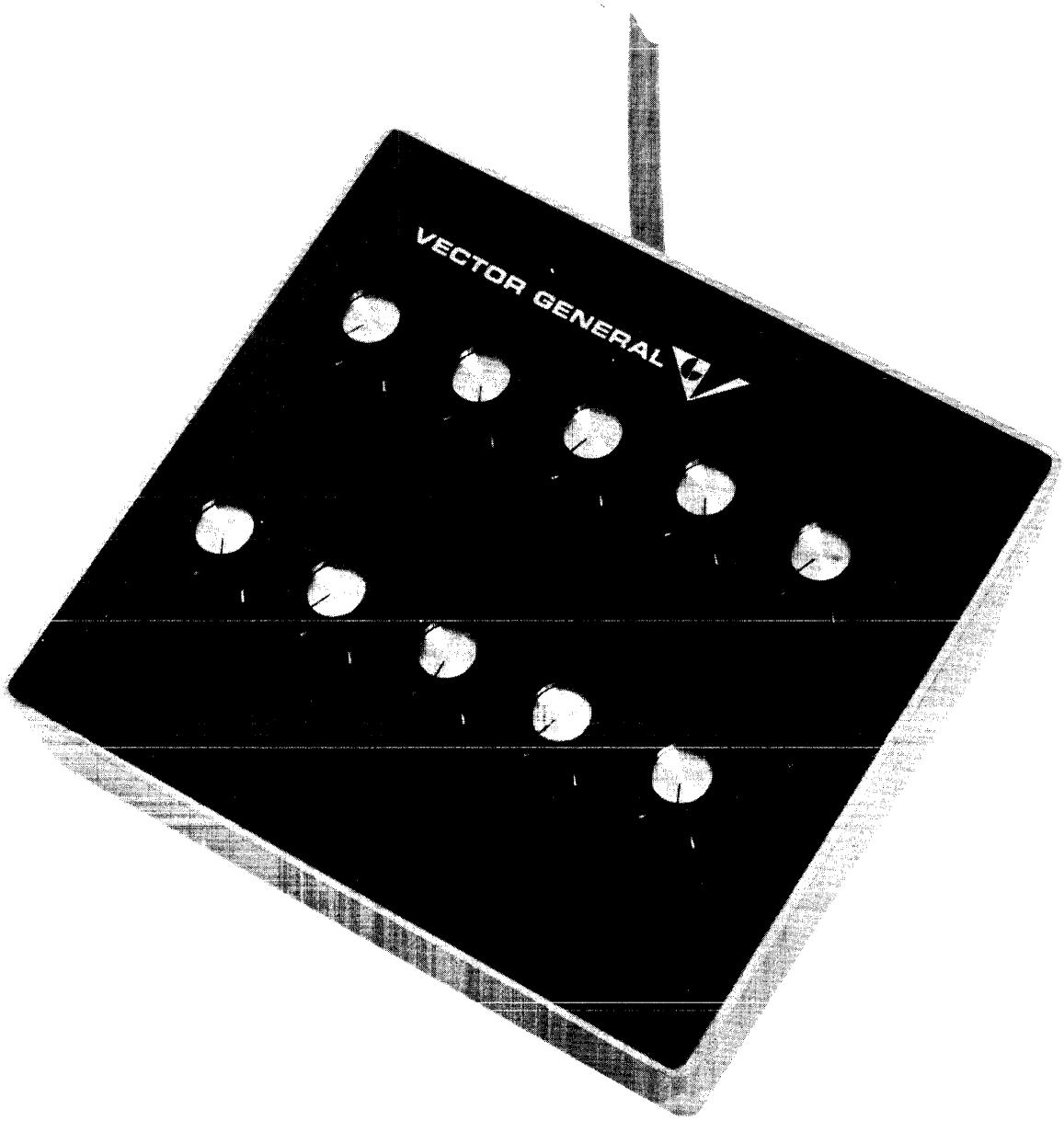


Figure 2-12. Control Dials - Top View

2.18 CONTROL DIALS

Ten optional control dials may be used to send digital numerical information to the computer for any purpose specified by the program. Each dial is associated with a 12-bit dial input register in the display controller. As the dial is turned, the corresponding register will read back a succession of numbers. These numbers can be read by the computer at any time.

2.19 PROGRAMMED INPUT/OUTPUT CHANNEL

The display is stopped or started and interrupts may be acknowledged by the computer over the programmed input/output channel. This channel also is used to read the contents of the display registers. A source address is sent to the controller to specify which register is to be read first. If further reading is programmed, the contents of other registers are read in numerical order by adding one to the source address each time a register is read.

2.20 INTERRUPT CHANNEL

A bit in the priority interrupt register is set when an interrupt condition is detected. If the corresponding enable bit is set in the mode control register, an interrupt is sent to the computer.

2.21 SELF TESTER

A hardware option designed to test the display controller and monitor without using the host computer. In addition to testing the primary controller functions, the tester provides display patterns to facilitate analog circuit adjustments.

SECTION III

DISPLAY SYSTEM PROGRAMMING

3.1 INTRODUCTION

This section contains a discussion of the priority interrupt system as well as a functional description of each display instruction with its applicable data lists. The display system registers available to the programmer are described, and descriptions of the various word formats used in programming the Display System are given.

Operation of the display system consists of processing data words in accordance with their associated instructions. Instructions that draw lines or text strings process data words giving the end point coordinates of the lines or character codes of the text. Register destination instructions are followed by data words containing the information to be acted upon and written into the addressed register.

3.2 PROGRAMMATIC INTERFACE

The interface between the display system and the computer consists of:

a. A single programmed I/O channel

b. A single priority interrupt level

c. A single direct memory access channel

The display presented to the viewer is sent by means of a direct memory access (DMA) block transfer data channel. A computer program must service the DMA to output the lists of display instructions.

Programmatic I/O operations are used by computer programs to control the display system, read its status, and communicate with any peripheral I/O devices.

The interrupt is used to support the peripheral I/O devices and to execute programs required by the display lists being output. The use of the interrupt system is further elaborated in Paragraph 3.9.

3.3 DISPLAY SYSTEM REGISTERS

The display system contains registers directly addressable by the program. Registers with DAR addresses (Figure 3-1) may be changed by display instructions and are therefore referred to as destination registers. All registers with SAR addresses (Figure 3-1) may be input by a program with a programmed I/O read operation and are therefore referred to as source registers. The address of a register to be changed by a display instruction is held in a nonaccessible destination address register (DAR), and the address of the next register to be read via programmed input is held in the source address register (SAR).

3.4 DESTINATION REGISTERS

Registers with listed DAR addresses are directly addressable as destination registers, and their contents can be changed by register setting display instructions. Figure 3-1 illustrates the registers and gives the register names and their mnemonics.

Register 6 is the instruction register (IR) which holds the current display instruction of the list being processed from DMA output. If the IR register is used as a DAR, the register remains unmodified. Register 7 is the word count register (WCR) and is reset to zero each time the data channel is restarted. As each display list word is transmitted for display processing, the word count is increased by one count.

REGISTERS
The following mnemonics define the symbolic display register designations for use in Source and Destination address fields (SAR, DAR) of display instructions or programmed I/O operations.

REGISTER BIT POSITIONS

DD	2D	2DR	3D	Register	Mne	Hex	Octal	DAR	SAR	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
				Function switches Unit 1	FS.	0000	00000		0	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
				Function switch lamp 0-7	LTH.	0000	00000	0		L0	L1	L2	L3	L4	L5	L6	L7								
				Keyboard Character	KBR.	0001	00001		1	K0	K1	K2	K3	K4	K5	K6	K7								
				Function switch lamp 8-15	LTL.	0001	00001	1		L8	L9	L10	L11	L12	L13	L14	L15								
				Tablet X input	TIX.	0002	00002		2	TX0	TX1	TX2	TX3	TX4	TX5	TX6	TX7	TX8	TX9					STN	STP
				Tablet Y input	TIY.	0003	00003		3	TY0	TY1	TY2	TY3	TY4	TY5	TY6	TY7	TY8	TY9					XOS	YOS
X	X	X	X	Priority interrupt requests (& name)	PIR.	0004	00004	4	4	NAME								PID	PIC	PIP	PIT	PIK	PIS	PIW	SP1
X	X	X	X	Mode & Control (incl interr enables)	MCR.	0005	00005	5	5	MED	MEC	MEP	MET	MEK	MES	MDB	MPH	MS1	MS2	MS3	MS4	PB		MDR	MDW
X	X	X	X	Display Instruction	IR.	0006	00006		6	P	INST			ADDR MODE				MODIFIERS							
X	X	X	X	Word count	(1) WCR.	0007	00007		7																
X	X	X	X	X-Coordinate	XR.	0008	00010	8	8	±															
X	X	X	X	Y-Coordinate	YR.	0009	00011	9	9	±															
			X	Z-Coordinate	ZR.	000A	00012	10	10	±															
X	X	X	X	Auto-increment	AIR.	000B	00013	11	11	±															
X	X	X	X	Intensity offset (dimming)	IOR.	000C	00014	12	12	±															
			X	Intensity scale (cueing)	ISR.	000D	00015	13	13																
				Memory fetch address	(2) MAR.	000E	00016	14	14	DISPLACEMENT												PAGE			
				Stack pointer	(2) SPR.	000F	00017	15	15	DISPLACEMENT												PAGE			
				Temp. general purpose	(2) TGR	0010	00020	16	16																
	X	X	X	Picture scale	PSR.	0011	00021	17	17	±															
X	X	X	X	Name Byte	NMR.	0012	00022	18	18	NAME															
	X	X	X	Coordinate Scale	CSR.	0013	00023	19	19	±															
	X	X	X	Coordinate X displacement	DXR.	0014	00024	20	20	±															
	X	X	X	Coordinate Y displacement	DYR.	0015	00025	21	21	±															
			X	Coordinate Z displacement	DZR.	0016	00026	22	22	±															
		X	X	Rotation matrix X/X Scale	R11R.	0017	00027	23	23	±															
		X	X	Rotation matrix X/Y Scale	R12R.	0018	00030	24	24	±															
			X	Rotation matrix X/Z Scale	R13R.	0019	00031	25	25	±															
		X	X	Rotation matrix Y/X Scale	R21R.	001A	00032	26	26	±															
		X	X	Rotation matrix Y/Y Scale	R22R.	001B	00033	27	27	±															
			X	Rotation matrix Y/Z Scale	R23R.	001C	00034	28	28	±															
			X	Rotation matrix Z/X Scale	R31R.	001D	00035	29	29	±															
			X	Rotation matrix Z/Y Scale	R32R.	001E	00036	30	30	±															
			X	Rotation matrix Z/Z Scale	R33R.	001F	00037	31	31	±															
				Window mode control	WMCR.	0020	00040	32	32	MBI	MBO	MEI	MEO	MWH	PWD	PWI	PWO	PXH	PXL	PYH	PYL	PZH	PZL	ACM	
				Window boundry X high	XHR.	0021	00041	33	33	±															
				Window boundry X low	XLR.	0022	00042	34	34	±															
				Window boundry Y high	YHR.	0023	00043	35	35	±															
				Window boundry Y low	YLR.	0024	00044	36	36	±															
				Window boundry Z high	ZHR.	0025	00045	37	37	±															
				Window boundry Z low	ZLR.	0026	00046	38	38	±															
				Post X displacement	PDXR.	0027	00047	39	39	±															
				Post Y displacement	PDYR.	0028	00050	40	40	±															
				Color control	CCR.	0029	00051	41	41	ECFL					COLOR FIELD										
				Ex dev priority interr requests	PIRX.	002E	00056	46	46	PIP2	PIP3	PIP4	PIK2	PIK3	PIK4	PIS2	PIS3	PIS4					SP2	SP3	SP4
				Ex dev interrupt enables	MCRX.	002F	00057	47	47	MEP2	MEP3	MEP4	MEK2	MEK3	MEK4	MES2	MES3	MES4					PSL	FRM	
				Hard Copy Register	HCR.	0030	00060		48	STATUS															PENR (3)
				Function switches unit 2	FS2.	0034	00064		52	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
				Function switch lamps 0-7 unit 2	LT2H.	0034	00064	52		L0	L1	L2	L3	L4	L5	L6	L7								
				Keyboard character unit 2	KB2.	0035	00065		53	K0	K1	K2	K3	K4	K5	K6	K7								
				Function switch lamps 8-15 unit 2	LT2L.	0035	00065	53		L8	L9	L10	L11	L12	L13	L14	L15								
				Function switches unit 3	FS3.	0038	00070		56	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
				Function switch lamps 0-7 unit 3	LT3H.	003[illegible]	00070	56		L0	L1	L2	L3	L4	L5	L6	L7								
				Keyboard character unit 3	KB3.	0039	00071		57	K0	K1	K2	K3	K4	K5	K6	K7								
				Function switch lamps 8-15 unit 3	LT3L.	0039	00071	57		L8	L9	L10	L11	L12	L13	L14	L15								
				Function switches unit 4	FS4.	003C	00074		60	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
				Function switch lamps 0-7 unit 4	LT4H.	003C	00074	60		L0	L1	L2	L3	L4	L5	L6	L7								
				Keyboard character unit 4	KB4.	003D	00075		61	K0	K1	K2	K3	K4	K5	K6	K7								
				Function switch lamps 8-15 unit 4	LT4L.	003D	00075	61		L8	L9	L10	L11	L12	L13	L14	L15								
									62																
				Storage Display Unit	SDUR.	003F	00077	63	63	ERAC	SDUM	WTRU	SNON	VEEU											
				Picture X-coordinate	PX.	0040	00100		64	±															
				Picture Y-coordinate	PY.	0041	00101		65	±															
				Picture Z-coordinate	PZ.	0042	00102		66	±															
				Joystick X input Unit #1	JX.	0043	00103		67	±															
				Joystick Y input Unit #1	JY.	0044	00104		68	±															
				Joystick Z input Unit #1	JZ.	0045	00105		69	±															
				(4) First Dial Unit #1 (dial n = DIAL. +n)	DIAL.	0046	00106		70	±															
				Window acquisition X coordinate	CXR.	0050	00120		80	±															
				Window acquisition Y coordinate	CYR.	0051	00121		81	±															
				Window acquisition Z coordinate	CZR.	0052	00122		82	±															
				Joystick X input Unit #2	JX2.	0053	00123		83	±															
				Joystick Y input Unit #2	JY2.	0054	00124		84	±															
				Joystick Z input Unit #2	JZ2.	0055	00125		85	±															
				(5) First Dial Unit #2 (dial n = CD11. +n)	CD11.	0056	00126		86	±															
				Track ball X coordinate	TBX.	0060	00140		96	±															
				Track ball Y coordinate	TBY.	0061	00141		97	±															
				(6) Print from Monitor #1	PFM1.	007C	00174			THESE 4 ADDRESSES RESERVED FOR HARD COPY UNIT.															
				(6) Print from Monitor #2	PFM2.	007D	00175																		
				(6) Print from Monitor #3	PFM3.	007E	00176																		
				(6) Print from Monitor #4	PFM4.	007F	00177																		

(1) Not available on all systems.
(2) Subroutine Stack option.
(3) LP2 only.
(4) Dials 2 through 10 use HEX addresses 0047 through 004F.
(5) Dials 2 through 10 use HEX addresses 0057 through 005F.
(6) PI/O only.

101051-99

Figure 3-1. Display System Registers

3.5 SOURCE REGISTERS

All SAR registers may be read by means of a programmed input read operation. The register to be read must first be selected by setting its address in the source address register (SAR) through a programmed output write. After the designated register is read, the SAR is stepped by one, allowing successive registers to be read in sequence by successive programmed read operations. Registers which correspond to analog values (i.e., SAR = 64 - 97) will initiate an analog-to-digital conversion operation to obtain the input value. The conversion is automatically initiated whenever SAR addresses a new analog input value.

The PX, PY and PZ registers can be used to obtain the current beam position when the system includes an analog-to-digital converter. In that case:

$$PX = -1/2\ X',\ PY = -1/2\ Y',\ PZ = -1/2\ Z'$$

Where X' is (for a 2D system with rotation)

$$X' = PS\ [CS\ (X \cdot R11 + Y \cdot R12) + DX] + PDX$$

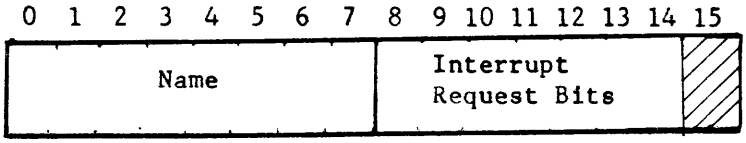
Y' and Z'are derived similarly.

Figure 3-1 gives the source and destination addresses for all display system registers and gives the register names and their mnemonics.

3.6 PROGRAMMED I/O

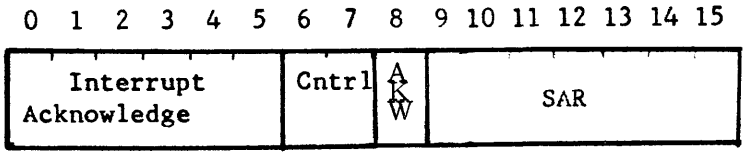
There are two programmed I/O operations: Programmed input read and programmed output write.

Programmed input is used by any computer program to read display state or status, transform or coordinate values, peripheral inputs, etc. The format of the word input matches that of the display register being read. For example, if the SAR specifies source register 4 (PIR), then bits 8 through 14 of the word read via programmed input constitute the interrupt request bits and are set by the individual device requesting an interrupt. Bit positions 0 through 7 of the input word can be used as a name field to identify interrupt requirements as on different pen-sensitive image constructs.



PIO Sample Input of PIR

Bit positions 0 through 5 of the word written via programmed output are interrupt acknowledge bits. These are used to reset the applicable interrupt request bit in the interrupt condition sense register (PIR) after the requested interrupt has been processed. Bit positions 6 and 7 control the starting and stopping of the display. Bit positions 9 through 15 of the output word indicate which source register is to be read next on the programmed input. The word format for programmed output words is shown in the following diagram.



Programmed Output Write Format

3.7 PROGRAMMED OUTPUT

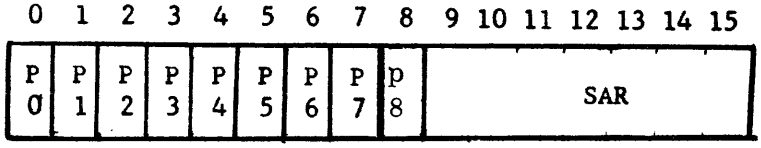
3.7.1 STANDARD PROGRAMMED OUTPUT

The programmed output write word is sent on the programmed I/O channel to the interrupt acknowledge and source address register. Bits 0 through 5 & 8, the interrupt acknowledge field, reset interrupt request bits in the priority interrupt request register (PIR). Bit 7 of the acknowledge field is used to clear and restart the display system processing of an instruction/data stream. Bit 6 is used to stop and clear the display system.

The programmed write output word can perform the following three functions:

- Acknowledge and release any enabled active requested interrupts which are pending, and restart the display if it was waiting
- Clear current display activities and start or stop display processing
- Designate the initial display register for subsequent programmed read operations

The programmed output write word is shown in the following diagram.



P0 Acknowledge Display Interrupt (AKD)

P1 Acknowledge Frame Clock Interrupt (AKC)

P2 Acknowledge Light Pen Interrupt (AKP) (and continue if waiting)

P3 Acknowledge Data Tablet Interrupt (AKT)

P4 Acknowledge Keyboard Interrupt (AKK)

P5 Acknowledge Function Switch Interrupt (AKS)

P6 Stop and Clear Display Controller (SCL)

P7 Reset and Start Display (CSD)

P8 Acknowledge window interrupt (AKW)

SAR Source Address Register (to be read) (P9 through P15)

3.7.2 EXTENDED PROGRAMMED OUTPUT

The extended programmed write output word* can perform the following functions:

- Acknowledge interrupts for extended devices.
- Control addressing beyond 32K.
- Control display transfer mode.

* not supplied with all configurations.

The extended programmed output write word (PIOX) is shown below.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P 0	P 1	P 2	P 3	P 4	P 5	P 6	P 7	P 8	P 9	P 10	P 11	P 12	P 13	P 14	P 15

P0 Acknowledge light pen #2 interrupt (AKP2).

P1 Acknowledge light pen #3 interrupt (AKP3).

P2 Acknowledge light pen #4 interrupt (AKP4).

P3 Acknowledge keyboard #2 interrupt (AKK2).

P4 Acknowledge keyboard #3 interrupt (AKK3).

P5 Acknowledge keyboard #4 interrupt (AKK4).

P6 Acknowledge function switch box #2 interrupt (AKS2).

P7 Acknowledge function switch box #3 interrupt (AKS3).

P8 Acknowledge function switch box #4 interrupt (AKS4).

P9 Enable extended memory address bits (above 32K) for non-subroutine stack (LEM).

P10 & P11 MAR 17, 16 extended memory address bits.

P12 & P13 SPR 17, 16 extended stack pointer register bits.

P14 Buffered display enable (BDE).

P15 Direct display enable (DDE).

3.8 PROGRAMMED INPUT

The word input by the programmed I/O channel contains the current contents of the display system register addressed by the source address register (SAR). The source address register is loaded by a programmed output operation, and after each programmed read the source address register is advanced by one count to indicate the next register to be read. Thus, any set of registers may be read consecutively after a programmed output specifying the address of the first.

The source address registers and their contents are illustrated in Figure 3-1.

3.9 PRECISION OF ADC VALUES

The conversion of an input analog value is triggered by SAR addressing the values register (whether SAR was set directly via PIO output or stepped after a previous read). The conversion generates the sign after 3μs; the remaining bits of the value are generated at one per 1.1μs.

Thus, if programmed PIO input-store-step and test loop takes 16μs, the full 12-bit precision values will be obtained without the need for any delays. In the case of many devices (dials, joystick), the original data is of much lower precision so that higher-speed input loops also need not wait.

3.10 PRIORITY INTERRUPTS

The priority interrupts in the display system are controlled by the contents of the display system's mode control register (MCR, Register 5) and the priority interrupt request register (PIR, Register 4).

Interrupt conditions set selected bits in the PIR register. These bits can be sensed by a programmed input read of PIR.

Interrupt enabling is performed by the MCR. If an interrupt condition occurs and its corresponding enabling bit in MCR is set, an interrupt request is sent to the computer.

3.11 MODE CONTROL REGISTER (MCR)

Interrupts are enabled by including in the display list a display instruction to set the mode control bits to 1 for each interrupt to be enabled. The following diagram illustrates the mode control register and its bit configuration:

0	1	2	3	4	5	6	7	8	9	10	11	12 13	14	15
MED	MEC	MEP	MET	MEK	MES	MDB	MPH	MS1	MS2	MS3	MS4	PB	MDR	MDW

MED	Enable display interrupt on P-bit halt
MEC	Enable frame clock interrupt
MEP	Enable light pen hit detect interrupt
MET	Enable data tablet interrupt
MEK	Enable keyboard character-ready interrupt
MES	Enable manual interrupt switch interrupt
MDB	Enable display blink
MPH	Enable light pen halt
MS1	Display 1 scope select
MS2	Display 2 scope select
MS3	Display 3 scope select
MS4	Display 4 scope select
PB	Pen Hit Byte (input only: Ø Ø = word, 1 1 = right byte, 1 Ø = left byte)
MDR	Run mode (input only)
MDW	Wait mode (input only)

A particular interrupt activity can be disabled by sending a display list with a register change instruction to set the applicable mode control interrupt-enabling bit to zero.

3.11A MULTI-DEVICE MODE CONTROL REGISTER (MCRX)

Multi-device interrupts are enabled by including in the display list a display instruction to set the MCRX Control bits to 1 for each interrupt to be enabled. The following diagram illustrates the MCRX and its bit configuration.

MCRX

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MEP2	MEP3	MEP4	MEK2	MEK3	MEK4	MES2	MES3	MES4					PSL	FRM	

MEP2 Enable light pen hit detect interrupt #2

MEP3 Enable light pen hit detect interrupt #3

MEP4 Enable light pen hit detect interrupt #4

MEK2 Enable keyboard character-ready interrupt #2

MEK3 Enable keyboard character-ready interrupt #3

MEK4 Enable keyboard character-ready interrupt #4

MES2 Enable manual interrupts switch interrupt #2

MES3 Enable manual interrupts swithc interrupt #3

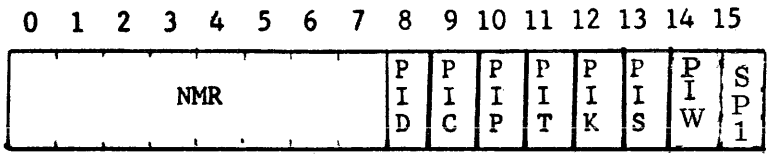
MES4 Enable manual interrupts switch interrupt #4

PSL Enable program slow light pen (LP 3)

FRM Enable frame start light pen (LP 3)

3.12 INTERRUPT REQUESTS

The device desiring an interrupt causes its interrupt bit in the PIR register to be set to 1. The following diagram illustrates the configuration of the PIR register and its interrupt/sense bit configuration.



NMR	Name field
PID	Display P-bit interrupt request
PIC	Frame clock interrupt request
PIP	Light pen interrupt request
PIT	Data tablet interrupt request
PIK	Keyboard character ready interrupt request
PIS	Manual interrupt switch interrupt request
PIW	Any window interrupt request
SP1	Light Pen 1 switch sense

If an interrupt request bit is set and its corresponding enabling bit in the MCR register is a 1, an interrupt request is generated and transmitted to the computer on the priority interrupt line. The computer program may then read the contents of the PIR register to determine the device requesting the interrupt. After the interrupt request is serviced, the program writes a word on the programmed I/O line to acknowledge the interrupt (Paragraph 3.7). The interrupt acknowledge bit resets the interrupt request bit in the PIR register.

3.13 DISPLAY P-BIT INTERRUPT (SUBROUTINE, JUMP FACILITY)

Display-interrupt generation is controlled by the P-bit (bit position 0) of all DMA display instructions. If the P-bit is a 1 in the NOOP or Halt instruction and the display interrupt has been enabled (MED in register 5 is set), display processing is halted and an interrupt request is generated.

On all other instructions, if the P-bit is a 1 and the display interrupt has been enabled, an interrupt request is generated when the terminate bit or terminate character is decoded in the last word of its data list.

If the P-bit is a zero, display processing continues with the next word following the terminate used as the next instruction.

The P-bit interrupt can be used to call a program which outputs data stored in noncontiguous areas of computer memory, thereby allowing for such operations as subimaging. For example, if a portion of a display, such as a circle, is required numerous times during the construction of the display, the coordinate data for generation of the circle can be stored in a contiguous area of memory disjoint from the main display list. Each time the circle is required during the display construction, the instruction for the desired circle display can be coded as a NOOP with the circle-list address and the P-bit. When processed, it will generate an interrupt request. The interrupt request can then be used to execute a driver program which will output the addressed circle display list prior to continuing with the main display list. The P and terminate bits in the circle sublist can then be used to cause a return to the main display list.

The P and terminate bits can also be used to call up routines to compose transforms for nested sublists with transformations, cause execution of programs to effect constraints, slave the display to a user program, or slave the display to on-line interactive device inputs. Use of the P and terminate bits is dependent on the desires of the user.

3.14 DISPLAY CONTROLLER STATUS

The last two bits of the MCR, bits MDR and MDW, indicate the current state of the display system.

By use of the programmed I/O, a programmed output write can acknowledge and reset PIR conditions and stop or start the display system. If bit 7 is a 1, the reset and start display operation is performed placing the display system in the run state:

MDR = 1

MDW = 0

While in the run state, the display system accepts words from the data channel and processes them for display. Instructions with associated data cause the successive words to be processed under the control of the instruction until a data word coded with a terminate condition is processed.

If a Halt instruction or an instruction with the P-bit set to 1 is processed, the display system halts or waits after the instruction and all its data have been processed.

The display system is then placed in the stop state:

MDR = 0

MDW = 0

No further information is accepted from the data channel.

The display system can also be set to pause upon detection of a light pen hit on any of a selected set of display elements by setting the pen-halt enable bit (MPH). The display system is then in a wait state:

MDR = 1

MDW = 1

During the processing of a light pen interrupt with pen-halt on, the display does not request or process any further instruction or data words from the data channel. Once the light pen hit has been processed, acknowledging and resetting the light pen interrupt request causes the display system to leave the wait state, and resume operation in the run state.

If the pen-halt is not on, a light pen interrupt will not cause the display system to leave run state. While the pen-interrupt program is being executed, the display will continue its processing beyond the display instruction at which the hit was detected.

If the pen interrupt (MEP) is not enabled (set = 1), no interrupt will occur. As with any interrupt condition, the corresponding PIR bit will be set and can be used by an executing background program as a sense bit.

3.15 DISPLAY INSTRUCTIONS

Instructions used in the display system fall into three main types: output instructions, used to generate image vectors or characters on the display; control instructions; and register change instructions, used to alter the contents of display system registers.

The following paragraphs contain functional descriptions of the various display-list instruction configurations processed by the display system as received over the DMA. Each instruction discussion includes a format diagram, a listing of both the octal and hexadecimal codes for the instruction variations, a definition of the applicable code, and a description of the purpose of the instruction.

The octal code given assumes the instruction to be an 18-bit instruction with the first two bits 0's. The hexadecimal code is given in single quotation marks and preceded by the letter X. For example, the hexadecimal notation for the decimal number 21 is written as X'15'.

The codes for fields are combined with the given instruction codes or data fields by a logical OR operation to obtain the value corresponding to any selection of mnemonics.

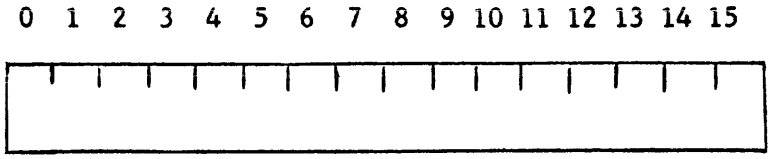
3.16 DATA LISTS

Operation of the display system consists of processing data words in accordance with their associated instructions. Instructions that draw lines or text strings process data words giving the end point coordinates of the lines or character codes of the text. Register destination instructions are followed by data words containing the information to be acted upon and written into the addressed register.

Data words are transmitted in a string or block following the applicable instruction. The last data word in the string must contain a coded terminate bit, field, or character to indicate that it is the last data word for that instruction.

3.17 WORD FORMATS

The display system uses as its basic informational element a 16-bit word with the bit positions numbered 0 through 15 as shown in the following diagram.

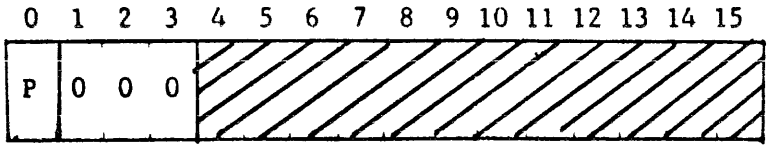


Bit position 0 represents the most significant portion of the word, and bit position 15 represents the least significant portion. This basic informational scheme is reflected in the operational registers and the internal elements of the display system.

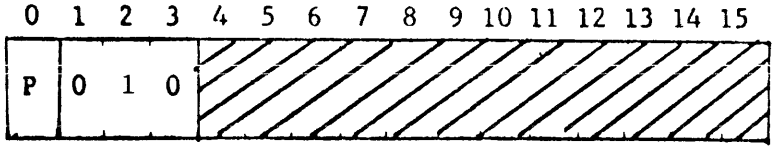
3.18 CONTROL DISPLAY INSTRUCTIONS

3.19 NO OPERATION

NOOP 000000_8 X'0000'



SPC 020000_8 X'2000'



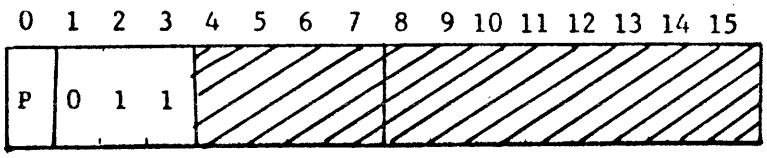
P Interrupt request

The NOOP or Special display instruction may be used to hold data or addresses. These can be used to label image portions or pass arguments to subimages or interrupt-called subprograms.

The P-bit permits extending the available display instructions or calling for the execution of arbitrary computer subroutines; the remaining bits (and/or following words) may give name, address, or arguments.

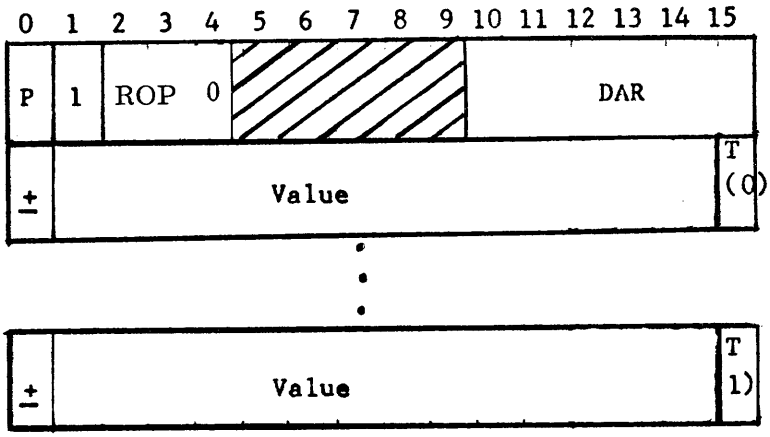
3.20 HALT

HLT 030000_8 X'3000'



The Halt instruction causes the display system to cease all operations. No further instructions or data words are accepted. The display system state is set to not-run, not-wait (MDR = 0, MDW = 0).

3.21 REGISTER CHANGE DISPLAY INSTRUCTIONS



P Interrupt request

ROP Register operation

DAR Initial register address

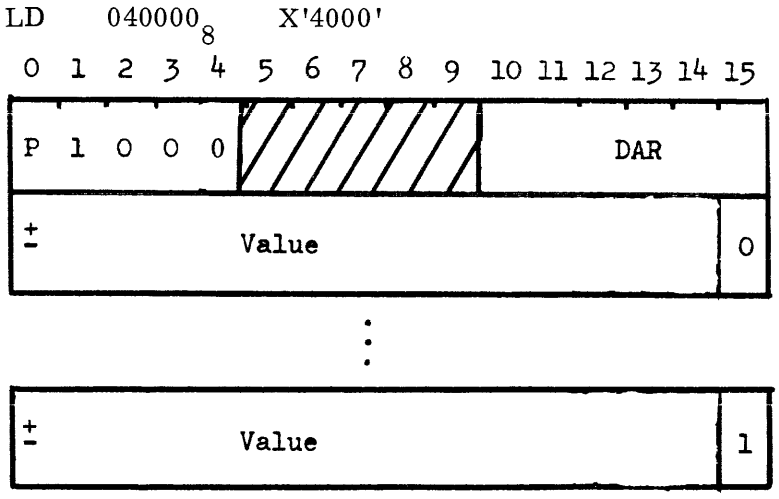
T Terminate data-list bit

The register-change instructions are used to alter the contents of any of the display system registers. The instruction designates whether new data is to be loaded into the register or an ADD, AND, or OR operation is to be performed between succeeding data and successive registers. The address contained in the DAR portion of the instruction specifies the first register in a sequence of registers to be affected by the data stream.

The register change data list words contain the new information to be placed into the destination register indicated by the destination address register (DAR). This information may be used to replace the data in the addressed register, added to the existing contents of the register, or logically OR'ed or AND'ed with the contents of the register. The word format for a register change data list is as shown.

When a register setting display instruction is processed, sequential addressing of the destination address register occurs until a terminate bit is decoded in the last of its data words. When the terminate bit is decoded the display beam position will be updated (blanked) to reflect the current values in the affected DARs.

3.22 LOAD REGISTERS

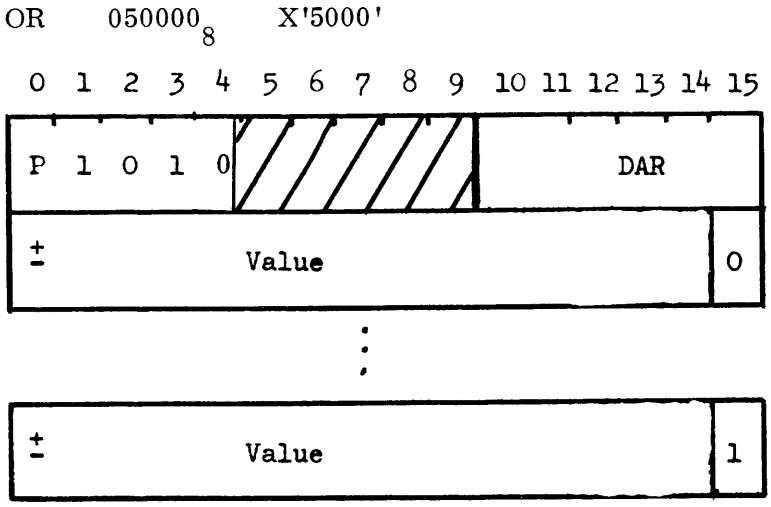


The Load Registers display instruction extracts the value field from its successive data words and loads them into succeeding display system registers, starting with the one designated by the DAR field.

Initial destination register address DAR:

PIR & NMR	04_8	X'4'	ISR	15_8	X'D'	R12R	30_8	X'18'
			PSR	21_8	X'11'	R13R	31_8	X'19'
MCR	05_8	X'5'	NMR	22_8	X'12'	R21R	32_8	X'1A'
XR	10_8	X'8'	CSR	23_8	X'13'	R22R	33_8	X'1B'
YR	11_8	X'9'	DXR	24_8	X'14'	R23R	34_8	X'1C'
ZR	12_8	X'A'	DYR	25_8	X'15'	R31R	35_8	X'1D'
AIR	13_8	X'B'	DZR	26_8	X'16'	R32R	36_8	X'1E'
IOR	14_8	X'C'	R11R	27_8	X'17'	R33R	37_8	X'1F'

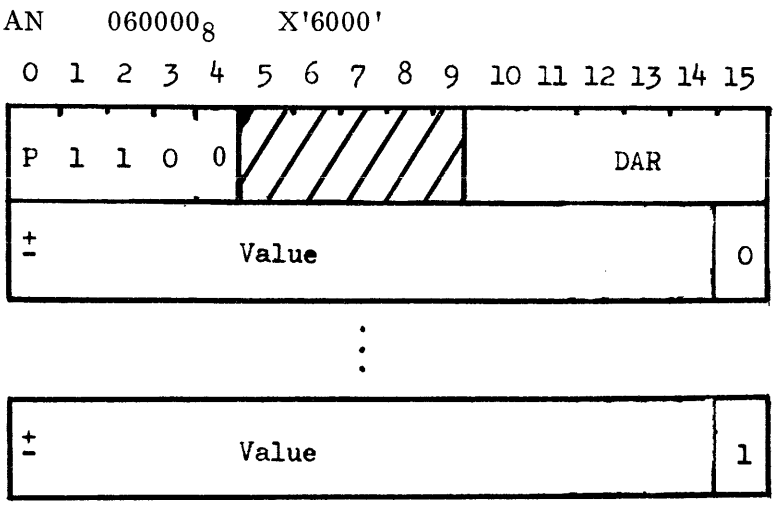
3.23 OR TO REGISTERS



The OR to Registers display instruction extracts the value field from its successive data words and OR's them to succeeding display system registers, starting with the one designated by the DAR field.

The DAR assignments are given in Paragraph 3.22.

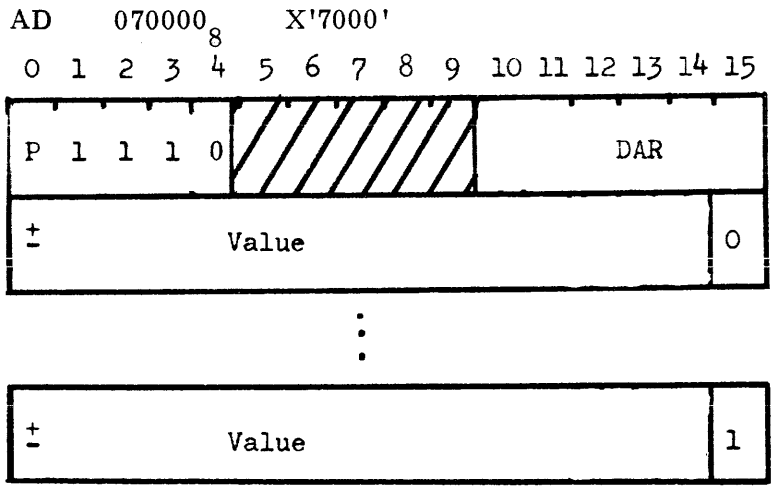
3.24 AND TO REGISTERS



The AND to Registers display instruction extracts the value field from its successive data words and AND's them to succeeding display system registers, starting with the one designated by the DAR field.

The DAR assignments are given in Paragraph 3.22.

3.25 ADD TO REGISTERS

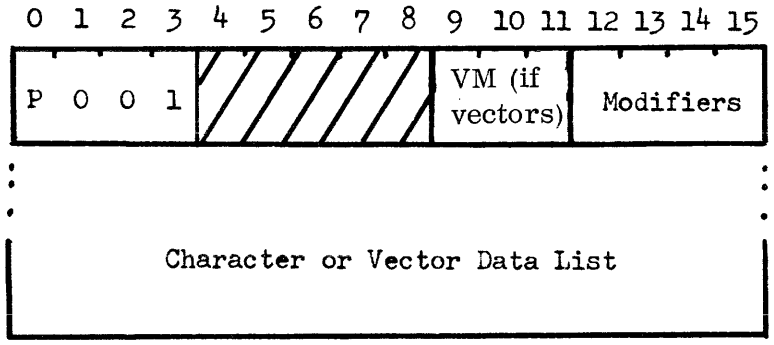


The Add to Registers display instruction extracts the value field from its successive data words and adds them to the high order 12-bits(0-11) of succeeding display system registers, starting with the one designated by the DAR field.

The DAR assignments are given in Paragraph 3.22.

3.26 DISPLAY WRITE INSTRUCTIONS

These display instructions and their following data are output as lists over the DMA channel to generate visual display elements. The basic word format is as shown in the following diagram:



P Interrupt request

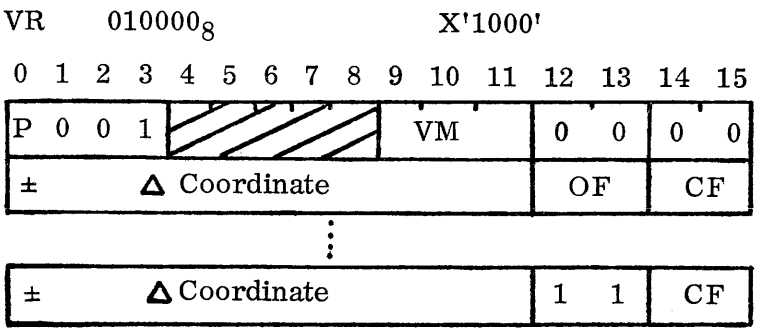
The image generation instructions are used to present display elements consisting of solid lines, dashed lines, dotted lines, or dash-dot-dashed lines between two positions on the display screen; points; and characters.

The modifier bits (12 through 15) of the image generation instruction specify if the data words that follow it are to be used for characters, absolute or relative vectors, X, Y, or Z autoincrementing, or 2D or 3D incremental vectors. The instruction also indicates the type of display (normal, dashed, dot, or point) and the incremental resolution or character scaling to be used.

The character generation instruction indicates the size of the characters to be displayed and whether they are to be displayed horizontally or vertically.

The following descriptions are given for no transformations imposed on the generated image prior to display. The user must load any transformation hardware with parameters to effect the desired transformation prior to processing any display generating instructions whose output is to be affected.

3.27 VECTOR RELATIVE



The display instruction for relative vectors generates a vector display whose coordinates are relative to the initial contents of the coordinate registers (XR, YR, ZR). The tipe of display generated by the moving beam is specified by the vector mode field(VM).

VM	Vector mode			
0 0 0	Line	blank	000000_8	X'0000'
0 0 1	Dashed line	DSH	000020_8	X'0010'
0 1 0	Dotted line	DOT	000040_8	X'0020'
0 1 1	End-point	PNT	000060_8	X'0030'
1 0 1	Dash-dot-dash		000120_8	X'0050'
1 1 0	Dash-dot-dash		000140_8	X'0060'

The operation field(OF) of each data word specifies if the beam is to be moved to a new position held in the coordinate registers; also, when moving the beam, it specifies if a vector type(VM) is to be drawn. The OF field also specifies the end of the data list.

OF	Operation field			
0 0	Load register	L	000000_8	X'0000'
0 1	Load, then draw vector	D	000004_8	X'0004'
1 0	Load, move beam (no draw)	M	000010_8	X'0008'
1 1	Load, draw, terminate	DT	000014_8	X'000C'

Each data word has a signed 12-bit coordinate increment to be added to a coordinate register or to the autoincrement register (AIR). The coordinate field (CF) of each data word specifies which register is to be updated by the coordinate increment.

CF	Coordinate field			
0 0	Autoincrement register (AIR)	AI	000000_8	X'0000'
0 1	X-coordinate register (XR)	X	000001_8	X'0001'
1 0	Y-coordinate register (YR)	Y	000002_8	X'0002'
1 1	Z-coordinate register (ZR)	Z	000003_8	X'0003'

CIRCLE/ARC GENERATOR (CAG)

To draw an arc with the CAG use the following procedure:

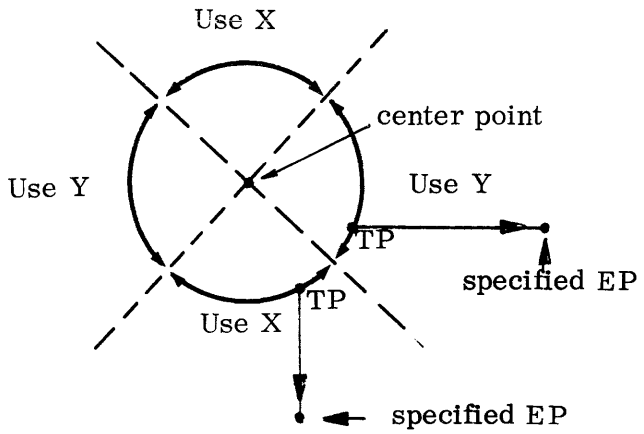
1. Using a VA or VR instruction load the X/Y coordinates of the arc start point and move or draw to that point. Do not terminate.
2. Load the X/Y coordinates of the arc end point.
3. Issue an arc command as follows:

For a clockwise arc	000004_8	X'0004'
For a counterclockwise arc	000010_8	X'0008'

4. Enter X coordinate of the arc center point using an OFCF field of 0001.
5. Enter Y coordinate of the arc center point with a LOAD Y and DRAW (OFCF = 1110 if a terminate is desired). The draw is now made around this centerpoint to the end point specified in step 2. NOTE: The beam position ends up at the arc end point, however, the X/Y registers hold the coordinates of the arc centerpoint.

A circle may be drawn by either using the procedure above and specifying the same value for both the start point and end point of the arc or by leaving step 2 of the procedure described above out entirely and just moving or drawing to the start point before issuing the arc command.

If the radius to start and to endpoint are not equal, that of the start point will be used and the CAG will use either the X or Y coordinate for terminating the draw in accordance with the following diagram. A straight line will be drawn from the terminating point on the radius to the specified endpoint.



3.28 VECTOR RELATIVE AUTO-X

VR IX 010001_8 X'1001'

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P	0	0	1	//////					V	M		0	0	0	1
±	Δ Coordinate													OF	CF

.
.
.

±	ΔCoordinate	1 1	CF

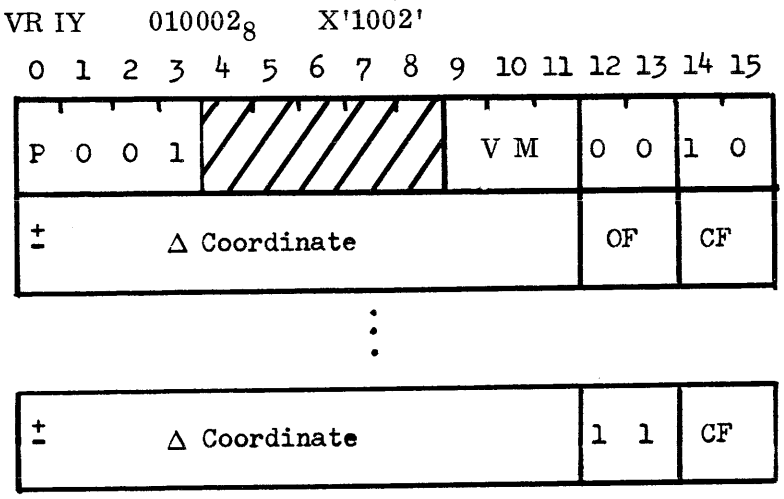
The display instruction for Vector Relative Auto-X processes its data as relative vectors. Each **Δ** coordinate value is added to the register designated by CF; then the vector generator performs any function specified by VM and OF. But, with each move or draw operation (OF = M, D, DJ), the X-coordinate register (XR) is incremented by the value in the autoincrement register (AIR) following the "load" portion but proceeding the "move" or "draw" portion of the operation.

The type of vectors generated is specified by VM as described in Paragraph 3.27.

Control of beam motion and blanking or list termination is specified by OF as described in Paragraph 3.27.

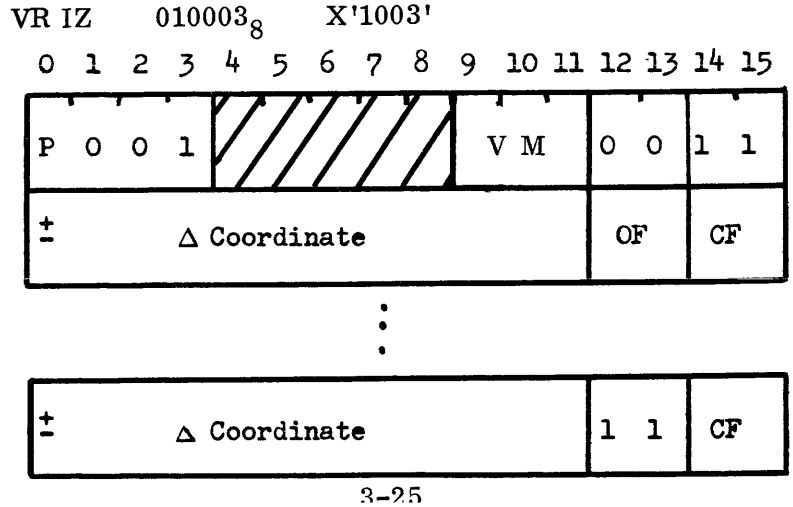
Specification of the register to be incremented by the Δ-coordinate value is given by CF as described in Paragraph 3.27.

3.29 VECTOR RELATIVE AUTO-Y



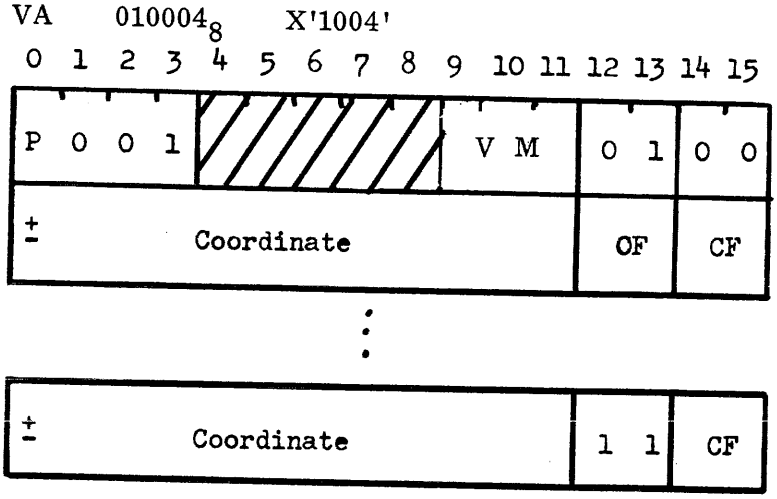
The display instruction for Vector Relative Auto-Y processes its data list as relative vectors. Each Δ coordinate value is added to the register designated by CF; then, the vector generator performs any function specified by VM and OF. But, with each move or draw operation (OF= M,D,DT) the Y-coordinate register (YR) is stepped by the increment in the autoincrement register (AIR) following the "load" portion but preceding the "move" or "draw" portion of the operation. The VM, OF, and CF fields are as described in Paragraph 3.27.

3.30 VECTOR RELATIVE AUTO-Z



The display instruction for Vector Relative Auto-Z processes its data list as relative vectors. Each Δ coordinate value is added to the register designated by CF; then, the vector generator performs any function specified by VM and OF. But, with each move or draw operation (OF= M,D,DT), the Z-coordinate register (ZR) is stepped by the increment in the autoincrement register (AIR) following the "load" portion, but preceding the "move" or "draw" portion of the operation. The VM, OF, and CF fields are as described in Paragraph 3.27.

3.31 VECTOR ABSOLUTE



The Vector Absolute display instruction loads the coordinate value from each of its data words directly into the register specified by CF, replacing the previous contents. The beam position is moved if called for by OF and a vector of type VM is drawn if required by CF. The VM, OF and CF fields for absolute vectors are the same as described for vector relative (paragraph 3.27).

CIRCLE/ARC GENERATOR (CAG)

To draw an arc with the CAG use the following procedure:

1. Move the beam or draw to the start point of the arc.
2. Load the endpoint into the coordinate registers.
3. Issue an arc command. The arc commands are:

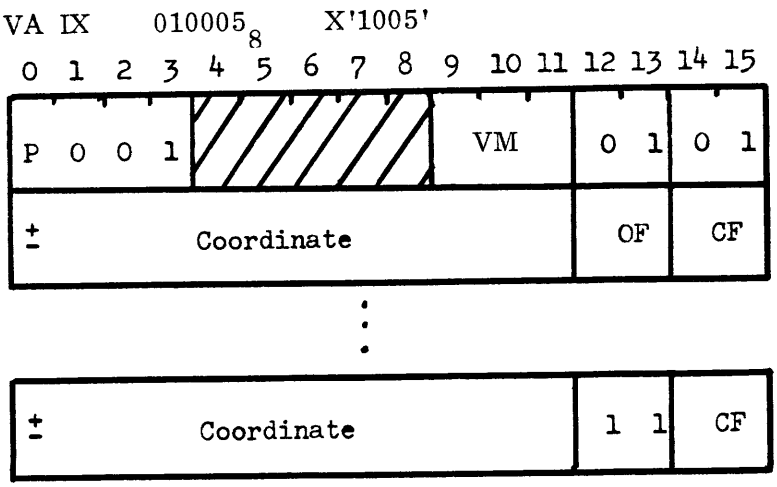
	OF-CF values	
Clockwise arc	000004_8	X'0004'
Counter clockwise arc	000010_8	X'0008'

4. Draw or draw-terminate to the centerpoint of the arc.

A circle may be drawn by either using the procedure described above and specifying the same value for both the start point and end point of the arc or by leaving step 2 of the procedure described above out entirely and just moving or drawing to the start point before issuing the arc command.

If the radius to start and to endpoint are not equal, that of the start point will be used and the CAG will draw an arc as described for the VECTOR RELATIVE command.

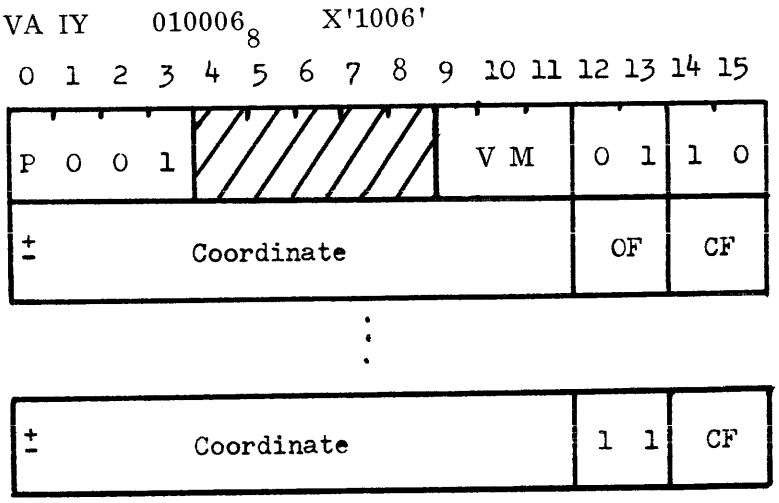
3.32 VECTOR ABSOLUTE AUTO-X



The display instruction for Vector Absolute Auto-X processes its data list as absolute vectors. Each coordinate value is loaded into the register designated by CF; then, the vector generator performs any move or VM-type draw operation if called for by OF. But, with each draw or move operation, (OF=M,D,DT), the X-coordinate register (XR) is stepped by adding the value from the auto-increment register (AIR) following the "load" portion but preceding the "move" or "draw" portion of the operation.

The VM, OF, and CF fields are used as descriged in Paragraph 3. 27.

3.33 VECTOR ABSOLUTE AUTO-Y

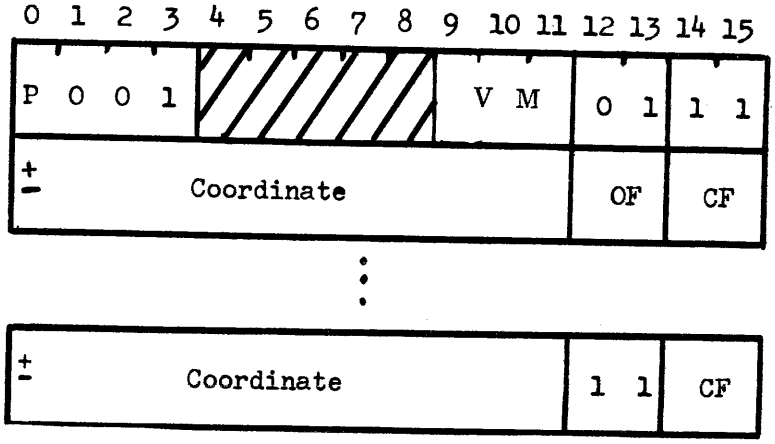


The display instruction for Vector Absolute Auto-Y processes its data list as absolute vectors. Each coordinate value is loaded into the register designated by CF; then, the vector generator performs any move or VM-type draw operation if called for by OF. But, with each draw or move operation, (OF=M. D. DT), the Y-coordinate register (YR) is stepped by adding the value from the auto-increment register (AIR) following the "load" portion but preceding the "move" or "draw" portion of the operation.

The VM, OF, and CF fields are used as described in Paragraph 3. 27.

3.34 VECTOR ABSOLUTE AUTO-Z

VA IZ 010007_8 X'1007'

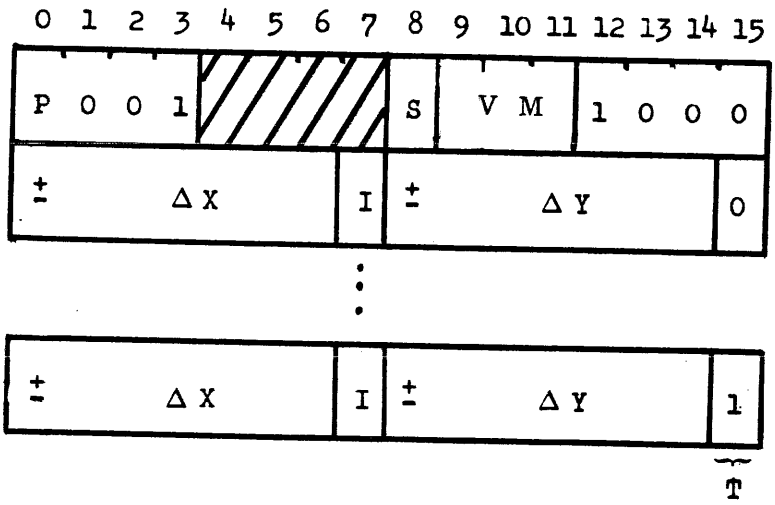


The display instruction for Vector Absolute Auto-Z processes its data list as absolute vectors. Each coordinate value is loaded into the register designated by CF; then, the vector generator performs any move or VM-type draw operation if called for by OF. But, with each draw or move operation, (OF= M, D, DT), the Z-coordinate register (ZR) is stepped by adding the value from the auto-increment register (AIR) following the "load" portion but preceding the "move" or "draw" portion of the operation.

The VM, OF, and CF fields are used as described in Paragraph 3. 27.

3.35 INCREMENTAL VECTORS, 2D

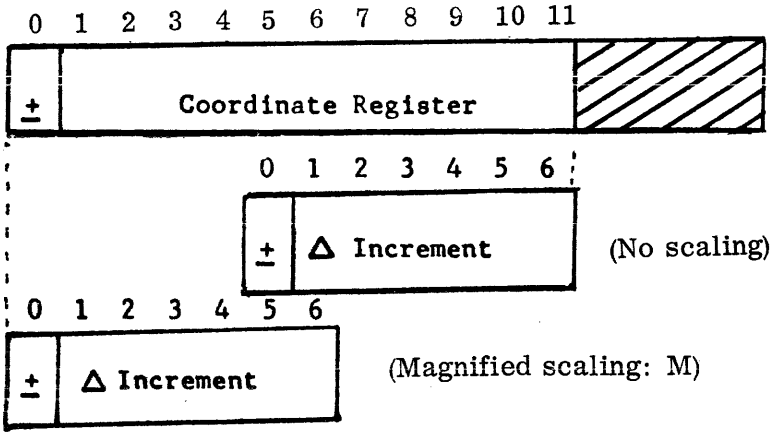
DVXY 010010_8 X'1008'



The 2D Incremental Vector display instructions generate an XY vector display whose coordinates are relative to the initial contents of the coordinate registers. Also, the maximum possible data rate has been doubled and the storage requirements halved (over those of relative vectors). This is done by reducing the Δ coordinate data field width to 7/12 and packing two values per data word. This performance increase can be exploited where the lower resolution data is adequate and the processing of packed values is not detrimental. The applicability of incremental vectors is enhanced by the scale field (S) which permits the data values to be applied as increments over a coarse or fine grid.

S	Increment scale			
0	No magnification: add Δ to 7 low-order bits	blank	000000_8	X'0000'
1	Magnified: add Δ to 7 high-order bits	M	000200_8	X'0080'

By specifying magnification, the coordinate increments are added to the high-order bits of the register being updated; otherwise the increment is sign-extended and added to the low-order bits:



The type of display generated by the moving beam is specified by the vector mode (VM) field (paragraph 3.27).

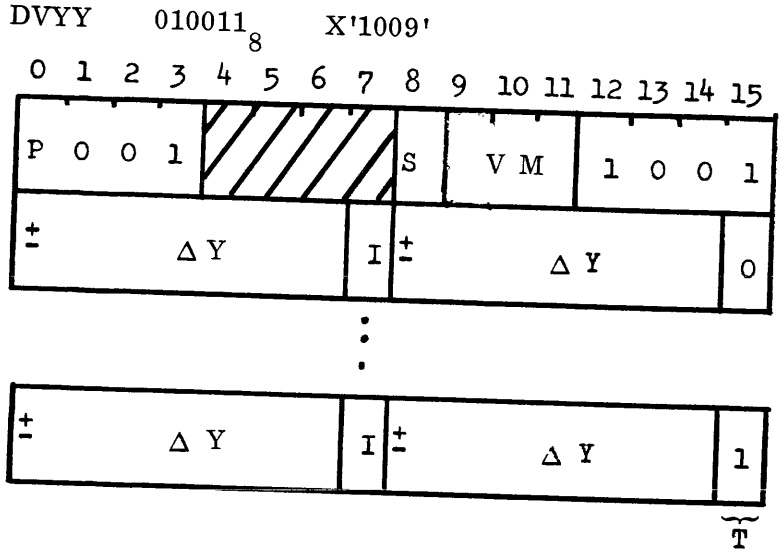
The I-field of the incremental vector data word controls beam blanking for processing of the entire data word.

I	Intensify field			
0	Move beam with no intensification	M	000000_8	X'0000'
1	Move beam and draw VM-type vector	D	000400_8	X'0100'

The last bit of an incremental vector data-list word is used to flag the end of the data list.

T	Terminate field			
0	Continue data list	blank	000000_8	X'0000'
1	Last word of data	T	000001_8	X'0001'

3.36 INCREMENTAL VECTORS, 2D AUTO-X

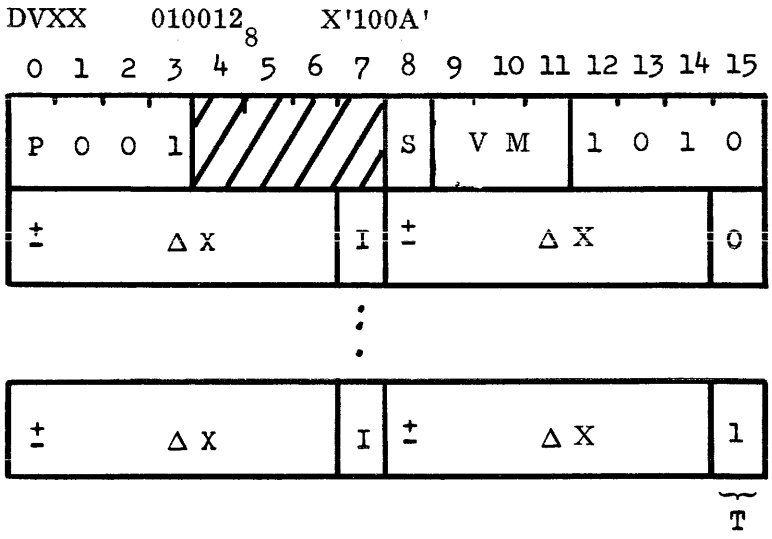


The 2D Auto-X display instruction generates a two-dimensional, relative, vector display from packed data increments; but the data words supply only Y-coordinate increments. The corresponding X-increments are taken as the constant held in the autoincrement register (AIR). This further doubles the possible vector rate and halves the core requirements for displays such as graphs, where one coordinate is stepped by a constant.

Each data word supplies two Y-increments and, therefore, is used to generate two vectors.

The S, VM, I, and T fields are coded and used as described in Paragraph 3.27, but the I-field applies to both vectors generated from its data word, and both vectors are generated from the final data word (T = 1).

3.37 INCREMENTAL VECTORS, 2D AUTO-Y



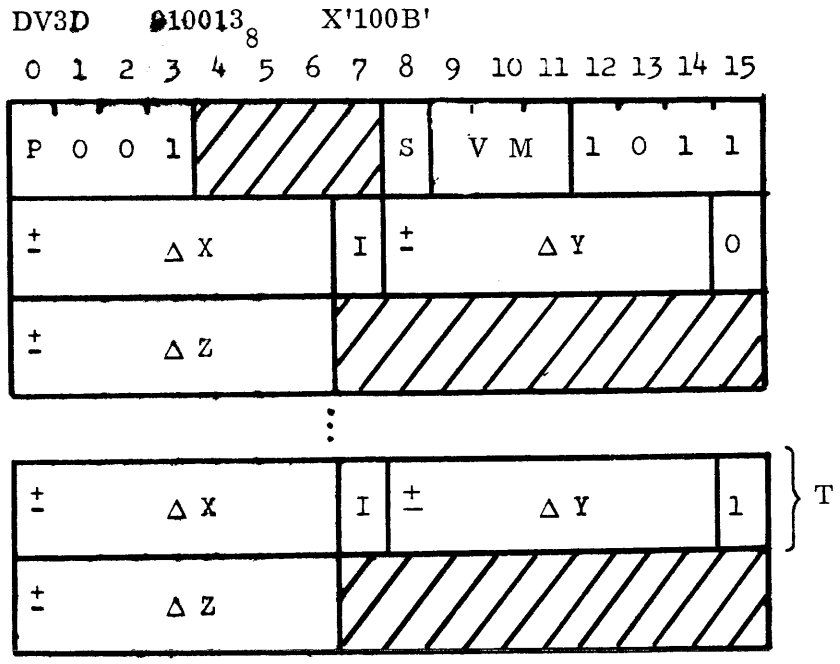
The 2D Auto-Y display instruction generates a two-dimensional, relative, vector display from packed data increments; but the data words supply only X-coordinate increments. The corresponding Y-increments are taken as the

constant held in the autoincrement register (AIR). This further doubles the possible vector rate and halves the core requirements for displays such as graphs, where one coordinate is stepped by a constant.

Each data word supplies two X-increments and, therefore, is used to generate two vectors.

The S, VM, I, and T fields are coded and used as described in Paragraph 3.27, but the I-field applies to both vectors generated from its data word, and both vectors are generated from the final data word (T = 1).

3.38 INCREMENTAL VECTORS, THREE DIMENSIONAL



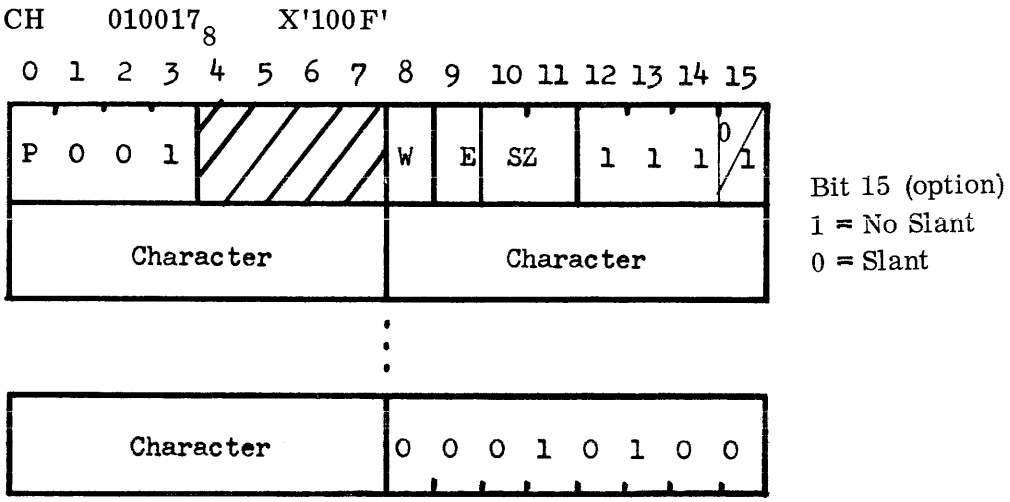
The 3D Incremental Vector display instructions generate an XYZ vector display whose coordinates are relative to the initial contents of the coordinate registers. Also, the maximum possible data rate has been increased and the storage requirements reduced (over those of relative vectors). This is done by shortening the Δ coordinate data field width to 7/12 and packing up to two values per

data word. This performance increase can be exploited where the lower resolution data is adequate and the processing of packed values is not detrimental. The applicability of incremental vectors is enhanced by the scale field (S) which permits the data values to be applied as increments over a coarse or fine grid. (Described in paragraph 3.35.)

One vector is generated for every two data words processed.

The S, VM, I, and T fields are coded and used as described in Paragraph 3.27.

3.39 CHARACTER GENERATION



The Character Generation display instruction processes its data as a string of extended ASCII character codes packed two per word.

Each successive character displays a symbol or performs a control function until a terminate character ASCII code DC4) is processed signaling the end of the instruction's data list.

The symbols available include all of the 96 ASCII graphics, plus a standard set of 96 additional symbols (programming, math, Greek, etc.), and an optional set of 32 user-specified special symbols.

The standard symbols and their codes are given in appendix A.

The direction field (W), when set, causes the characters to be displayed as if on a page which has been rotated 90^{o} counterclockwise.

W	Character write-direction			
0	Write characters horizontally	blank	000000_8	X'0000'
1	Write characters vertically	V	000200_8	X'0080'

The size field (SZ) is used to specify one of the four available string-controlled character sizes. The size-enable bit (E) causes the contents of the SZ field to be instated as the new character size for subsequent character generation.

E, SZ	Character size control			
0 XX	Use previous character size	blank	000000_8	X'0000'
1 0 0	Set size to 120 columns x 60 lines	S0	000100_8	X'0040'
1 0 1	Set size to 81 columns x 41 lines	S1	000120_8	X'0050'
1 1 0	Set size to 60 columns x 30 lines	S2	000140_8	X'0060'
1 1 1	Set size to 32 columns x 16 lines	S3	000160_8	X'0070'

Control Characters

Function	Character	Codes	
No display is generated and the beam is not stepped to the next character position	DELETE	X'7F'	077400_8 lh 000177_8 rh
	NULL	X'00'	000000_8 lh 000000_8 rh
Causes positioning to revert to the previous character position	BACKSPACE	X'08'	004000_8 lh 000010_8 rh

$X = X - CW_c$, where CW_c = current column width.

Control Characters (Cont.)

Function	Character	Codes	
Causes the current line position to be increased by one line $Y = Y - 2CH_c$	LINE FEED	X'0A'	005000_8 lh 000012_8 rh
Instates current character positioning at the first character of line 1, column 1 $X = -2048 + 1/2\ CW_3$ (Size 3 Character width) $Y = +2047 - 1/2\ CH_3$ (Size 3 Character height)	FORM FEED	X'0C'	006000_8 lh 000014_8 rh
Resets current column position to position 1, the left margin, and increases the current line position by one line $X = -2048 + 1/2\ CW_3$ $Y = Y - 2CH_c$	CARRIAGE RETURN (New line)	X'0D'	006400_8 lh 000015_8 rh
Reduces the current line position by one line $Y = Y + 2CH_c$	DC1	X'11'	010400_8 lh 000021_8 rh
Decreases the current character size by one size. Permits sub- and super-script sizes to be embedded in text. Size 0 is changed to Size 3.	DC2	X'12'	011000_8 lh 000022_8 rh
Increases the current character size by one size. Size 3 is changed to Size 0	DC3	X'13'	011400_8 lh 000023_8 rh
Terminates the data associated with a character generation instruction. If the instruction had P-bit set, display halts; if P-bit was not set, display continues and takes next word as a new instruction	DC4	X'14'	012000_8 lh 000024_8 rh
Resets the current column position to "horizontal center" and increases the current line position by one line	HORIZONTAL TAB (New line displaced) $X = 0 + 1/2CW_3$ $Y = Y - 2CH_c$	X'09'	004400 lh 000011 rh
Instates current character positioning to "horizontal center" of line one $X = 0 + 1/2CW_3$ $Y = +2047 - 1/2\ CH_3$	VERTICAL TAB (Form feed displaced)	X'0B'	005400 lh 000013 rh

SECTION IV

PROGRAM EXAMPLE

4.1 INTRODUCTION

This section contains a sample program for generating a simple display. Only a flow chart of computer instructions for the driver is given in the sample program since the actual instructions are dependent upon the individual computer. The sample program contains the display instructions and associated data words required to construct the display. The driver sends the display instructions and associated data words through the data channel in the form of block transfers.

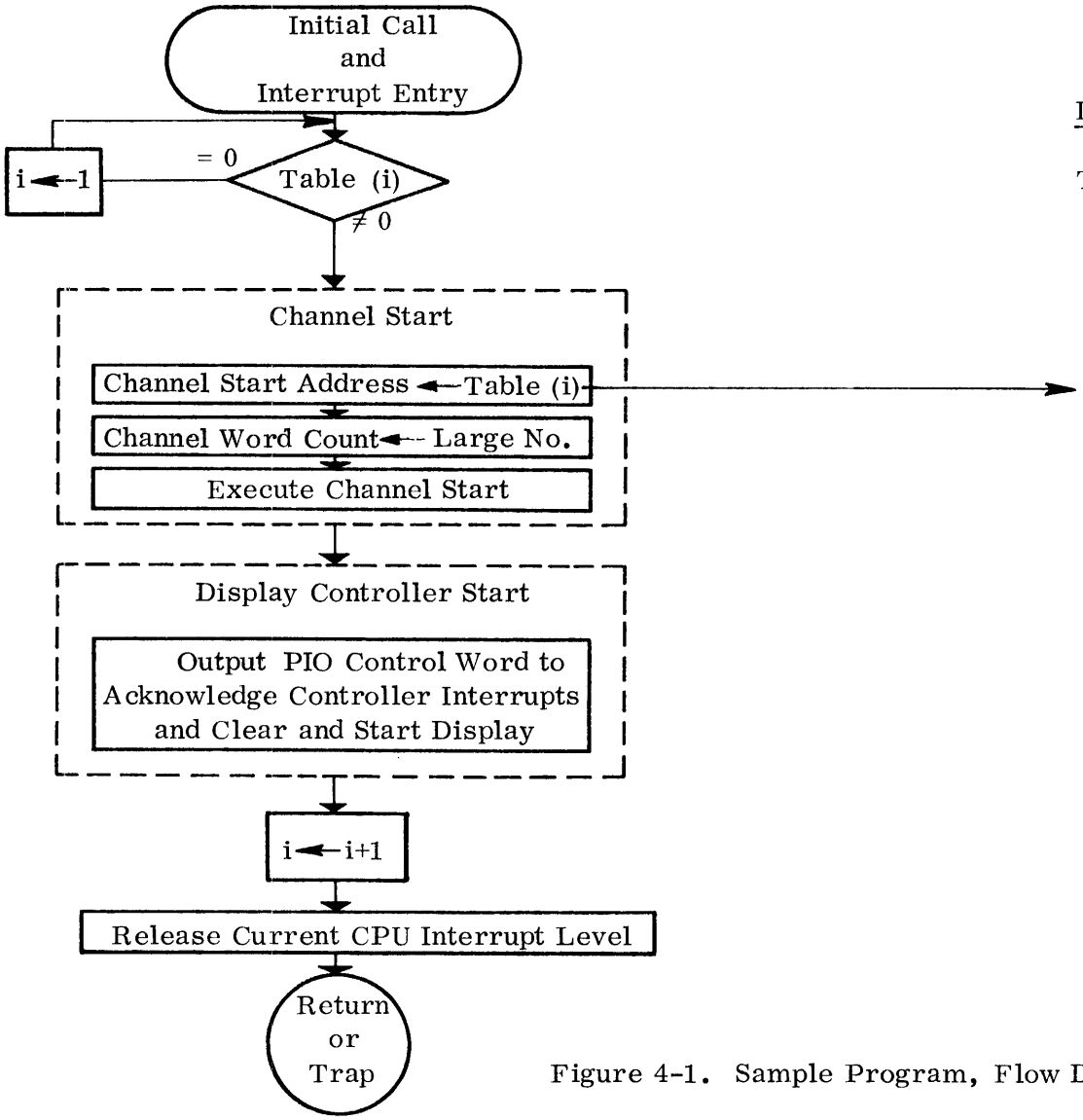
4.2 SAMPLE PROGRAM

Figure 4-1 is a flow diagram for the sample program driver code. This program constructs a large box and a small box each containing zigzag lines and the word Box as illustrated in Figure 4-3.

The program can be called up by a display interrupt request. The memory address associated with that interrupt request contains a branch instruction to the driver program illustrated in Figure 4-1. As indicated in Figure 4-1, each entry in the display pointer table points to one of five display lists. On the first pass, I = 1, the pointer table points through TABLE (1) to the location for list INITIAL (this initializes the system for a 2D transformation). After the channel has been started by the computer, a PIO control word is sent to the display system to start the display. The contents of the first list are then sent to the display

system. A one is added to the index of the pointer table and the process is repeated when the next display interrupt occurs.

The display lists pointed to by the addresses at TABLE (1) through TABLE (4) are used to generate the large box, the zigzag, and the word Box; lists pointed to by TABLE (5) through TABLE (7) are used to generate the small box, zigzag, and the word Box. The zeros in TABLE (8) indicate the end of TABLE. The actual display lists used to generate the picture in Figure 4-3 are given in Figure 4-2.



Display Pointer Table

Table:	Initial
	Trans 1
	Box
	Zigzag
	Trans 2
	Box
	Zigzag
	0

Figure 4-1. Sample Program, Flow Diagram

4005	040005	INITIAL	LD, MCR	LOAD MODE CONTROL (2D System)
C0F1	140361		MS1, MED, T	ENABLE DISPLAY INTERRUPT
400C	040014		LD, ICR	LOAD INTENSITY
7FF1	077761		2047, T	FULL SCALE BRIGHT
C011	140021		*LD, PSR	LOAD PICTURE SCALE
3FF1	037761		1023, T	HALF SCALE
C013	140023	TRANS1	*LD, CSR	LOAD BEGINNING WITH SCALE
3FF0	037760		1023	CSR: HALF SCALE
E000	160000		-511	DXR: -1/4 OFFSET LEFT
E001	160001		-511 , T	DYR: -1/4 OFFSET DOWN
1004	010004	BOX	VA	VECTOR ABSOLUTE INSTRUCTION
8001	100001		-2048, L, X	LOAD X COORDINATE
800A	100012		-2048, M, Y	LOAD Y COORDINATE AND MOVE
7FF5	077765		2047, D, X	LOAD X COORDINATE AND DRAW
7FF6	077766		2047, D, Y	LOAD Y COORDINATE AND DRAW
8005	100005		-2048, D, X	LOAD X COORDINATE AND DRAW
800E	100016		-2048, DT, Y	LOAD Y COORDINATE, DRAW AND TERMINATE
906F	110157		*CH, S2	CHARACTER GENERATION INSTRUCTION
1120	010440		'DC1 SP'	ASCII BYTES, NEGATIVE LINE FEED AND SPACE
426F	041157		"BO"	ASCII BYTES, B AND O
7814	074024		"X" 'DC4'	ASCII BYTES, X AND TERMINATE

NOTE: CODES ARE PRESENTED IN BOTH HEXADECIMAL AND OCTAL. FIRST CODE IS IN HEXADECIMAL NOTATION; SECOND CODE IS IN OCTAL NOTATION

*Interrupt (Display)

Figure 4-2. Sample Program Display Lists

```
4008    040010 ZIGZAG     LD, XR              LOAD STARTING WITH X-COORD
C000    140000            -1024               LOAD X COORDINATE WITH HALF FS
0001    000001            0, T                LOAD Y COORDINATE WITH ZERO
400B    040013            LD, AIR             LOAD INCREMENT REGISTER
07F1    001761            63, T               WITH 63
9009    110011          * DVYY                2D VECTOR INCREMENTAL,
                                              X AUTOINCREMENT
7E7E    077176              +63, M, +63       MOVE Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F82    077602              +63, D, -63       INCREMENT X, DRAW Y
7F83    077603              +63, D, -63, T    INCREMENT X, DRAW Y AND TERMINATE

C013    140023 TRANS2     * LD, CSR           LOAD BEGINNING WITH COORD SCALE
3F00    017760              512               LOAD CSR
3FF0    037760              1023              LOAD DXR
3FF1    037761              1023, T           LOAD DYR
```

NOTE: CODES ARE PRESENTED IN BOTH HEXADECIMAL AND OCTAL.
FIRST CODE IS IN HEXADECIMAL NOTATION; SECOND CODE
IS IN OCTAL NOTATION

Figure 4-2. Sample Program Display Lists (Cont.)

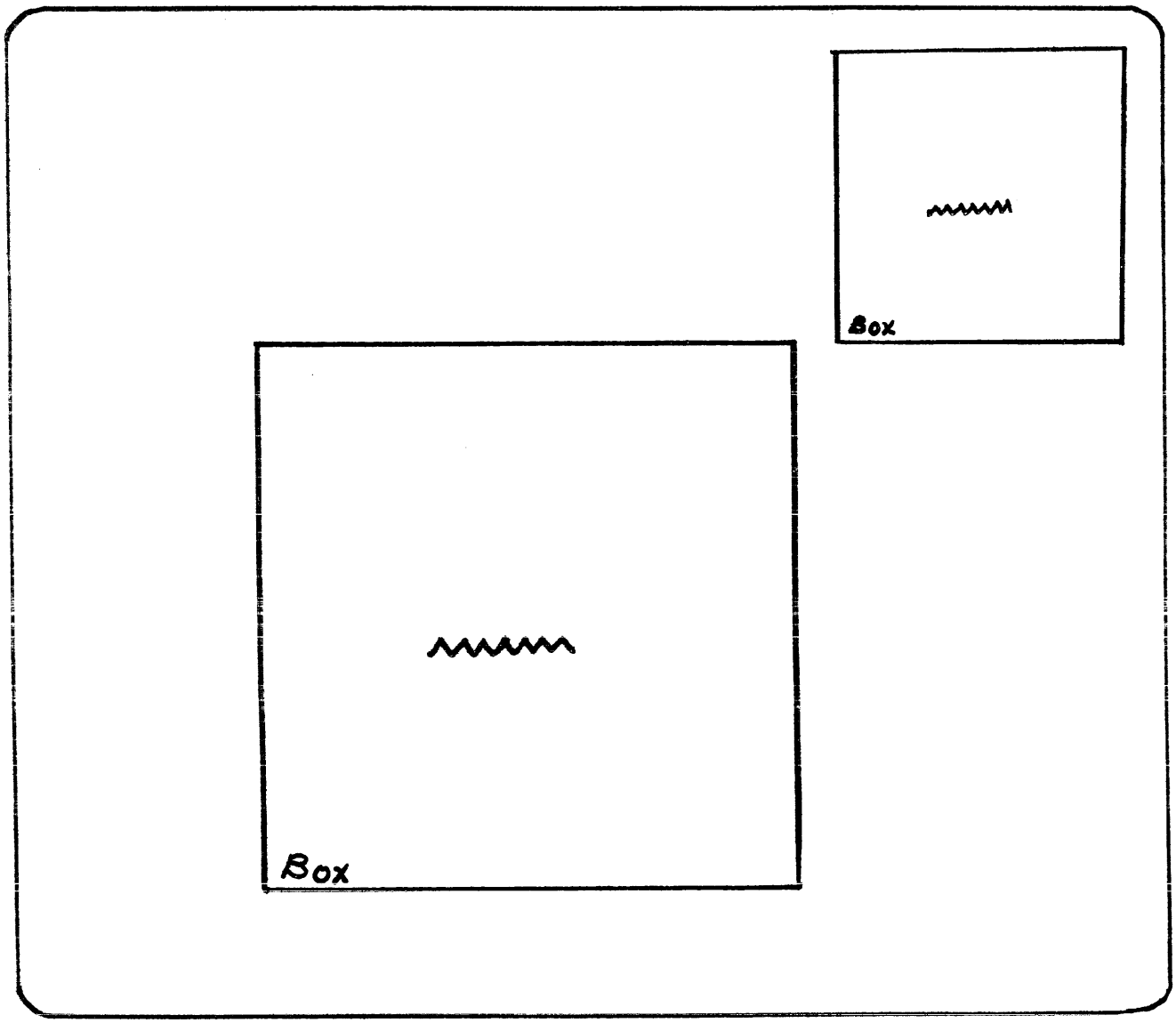


Figure 4-3. Sample Program, Typical Display

APPENDIX A

A.1 CHARACTER CODES

Table A-1 lists the ASCII* codes used by the display system for the various general and special characters. The codes are given in both octal and hexadecimal notation. The octal codes are given as though there were 18 bits in the data word instead of 16 bits. Since two characters can be given in each data word, the octal codes are given for the right half-word and the left half-word. The left half-word code is given as though there were no character in the right half-word. To obtain the complete code for the two characters in a word, the user must add the two codes together. For example, if the character C is to be in the left half-word and the character A is to be in the right half-word, the code would be:

C	041400	43
A	101	41
CA	041501_8	X'4341'

*American National Standard Code for Information Interchange

TABLE A-1. ASCII CHARACTER CODES

Hex	Octal Left	Right	Char. Gen. Sym.	Alpha. Num. Keyb. Keys
00	000000	000	NUL (ignored)	@ ctrl
01	000400	001	SO "	A ctrl
02	001000	002	STX "	B ctrl
03	001400	003	ETX "	C ctrl
04	002000	004	EOT "	D ctrl
05	002400	005	ENQ "	E ctrl
06	003000	006	ACK "	F ctrl
07	003400	007	BEL "	G ctrl
08	004000	010	BS	BS
09	004400	011	HT (LF, cent)	I ctrl
0A	005000	012	LF	LF
0B	005400	013	VT (top, cent)	K ctrl
0C	006000	014	FF (top, left)	L ctrl
0D	006400	015	NL (CR, LF)	CR
0E	007000	016	SE (ignored)	N ctrl
0F	007400	017	SI (ignored)	O ctrl
10	010000	020	DLE (ignored)	P ctrl
11	010400	021	DC1 (-LF)	Q ctrl
12	011000	022	DC2 (-SZ)	R ctrl
13	011400	023	DC3 (+SZ)	S ctrl
14	012000	024	DC4 (term)	T ctrl
15	012400	025	NAK (ignored)	U ctrl
16	013000	026	SYN (ignored)	V ctrl
17	013400	027	ETB (ignored)	W ctrl
18	014000	030	CAN (ignored)	X ctrl
19	014400	031	EM (ignored)	Y ctrl
1A	015000	032	SUB (ignored)	Z ctrl
1B	015400	033	ESC (ignored)	[ctrl
1C	016000	034	FS (ignored)	\ ctrl
1D	016400	035	GS (ignored)	] ctrl
1E	017000	036	RS (ignored)	∧ ctrl
1F	017400	037	US (ignored)	—
20	020000	040	Space	Sp bar
21	020400	041	!	1 shft
22	021000	042	"	2 shft
23	021400	043	#	3 shft
24	022000	044	$	4 shft
25	022400	045	%	5 shft
26	023000	046	&	6 shft
27	023400	047	'	7 shft
28	024000	050	(	8 shft
29	024400	051	)	9 shft
2A	025000	052	*	: shft
2B	025400	053	+	; shft
2C	026000	054	,	,
2D	026400	055	-	-
2E	027000	056	.	.
2F	027400	057	/	/
30	030000	060	ø	ø
31	030400	061	1	1
32	031000	062	2	2
33	031400	063	3	3
34	032000	064	4	4
35	032400	065	5	5
36	033000	066	6	6
37	033400	067	7	7
38	034000	070	8	8
39	034400	071	9	9
3A	035000	072	:	:
3B	035400	073	;	;
3C	036000	074	<	, shft
3D	036400	075	=	- shft
3E	037000	076	>	. shft
3F	037400	077	?	/ shft
40	040000	100	@	@
41	040400	101	A	A shft
42	041000	102	B	B shft
43	041400	103	C	C shft
44	042000	104	D	D shft
45	042400	105	E	E shft
46	043000	106	F	F shft
47	043400	107	G	G shft
48	044000	110	H	H shft
49	044400	111	I	I shft
4A	045000	112	J	J shft
4B	045400	113	K	K shft
4C	046000	114	L	L shft
4D	046400	115	M	M shft
4E	047000	116	N	N shft
4F	047400	117	O	O shft
50	050000	120	P	P shft

TABLE A-1. ASCII CHARACTER CODES (Cont.)

Hex	Octal Left	Right	Char. Gen. Sym.	Alpha. Num. Keyb. Keys
51	050400	121	Q	Q shft
52	051000	122	R	R shft
53	051400	123	S	S shft
54	052000	124	T	T shft
55	052400	125	U	U shft
56	053000	126	V	V shft
57	053400	127	W	W shft
58	054000	130	X	X shft
59	054400	131	Y	Y shft
5A	055000	132	Z	Z shft
5B	055400	133	[	[
5C	056000	134	\	\
5D	056400	135	]	]
5E	057000	136	^ (superscript)	^
5F	057400	137	_ (subscript)	_
60	060000	140	`	@ shft
61	060400	141	a	A
62	061000	142	b	B
63	061400	143	c	C
64	062000	144	d	D
65	062400	145	e	E
66	063000	146	f	F
67	063400	147	g	G
68	064000	150	h	H
69	064400	151	i	I
6A	065000	152	j	J
6B	065400	153	k	K
6C	066000	154	l	L
6D	066400	155	m	M
6E	067000	156	n	N
6F	067400	157	o	O
70	070000	160	p	P
71	070400	161	q	Q
72	071000	162	r	R
73	071400	163	s	S
74	072000	164	t	T
75	072400	165	u	U
76	073000	166	v	V
77	073400	167	w	W
78	074000	170	x	X
79	074400	171	y	Y
7A	075000	172	z	Z
7B	075400	173	{	[shft
7C	076000	174	|	\ shft
7D	076400	175	}	] shft
7E	077000	176	~	^ shft
7F	077400	177	del	DEL
80-9F	100000-117400	200-237	(*Note)	(**Note)
A0	120000	240	□ (cntr'd)	space spec
A1	120400	241	↓	1 shft spec
A2	121000	242	‖	2 shft spec
A3	121400	243	○ (cntr'd)	3 shft spec
A4	122000	244	£	4 shft spec
A5	122400	245	∨	5 shft spec
A6	123000	246	∫ (centered)	6 shft spec
A7	123400	247	√	7 shft spec
A8	124000	250	⊂	8 shft spec
A9	124400	251	⊃	9 shft spec
AA	125000	252	10 (subscript)	: shft spec
AB	125400	253	÷	; shft spec
AC	126000	254	≤	, spec
AD	126400	255	≡	- spec
AE	127000	256	≥	. spec
AF	127400	257	¬	/ spec
B0	130000	260	o	0 spec
B1	130400	261	↑	1 spec
B2	131000	262	∀	2 spec
B3	131400	263	□ (cntr'd)	3 spec
B4	132000	264	¢	4 spec
B5	132400	265	∧ (centered)	5 spec
B6	133000	266	∂	6 spec
B7	133400	267	∠	7 spec
B8	134000	270	∪	8 spec
B9	134400	271	∩	9 spec
BA	135000	272	• (center dot)	: spec
BB	135400	273	×	; spec
BC	136000	274	←	, shft spec

TABLE A-1. ASCII CHARACTER CODES (Continued)

Hex	Octal Left	Octal Right	Char. Gen. Symbol		Alpha. Num. Keyb. Keys
BD	136400	275	≠		- shft spec
BE	137000	276	→		. shft spec
BF	137400	277	∞		/ shft spec
C0	140000	300	••	(*)	@ spec
C1	140400	301	∀		A shft spec
C2	141000	302	—		B shft spec
C3	141400	303	⊥		C shft spec
C4	142000	304	Δ		D shft spec
C5	142400	305	∃		E shft spec
C6	143000	306	Φ		F shft spec
C7	143400	307	•	(*)	G shft spec
C8	144000	310	⊤		H shft spec
C9	144400	311	ψ		I shft spec
CA	145000	312	o		J shft spec
CB	145400	313	"		K shft spec
CC	146000	314	⋀		L shft spec
CD	146400	315	▥		M shft spec
CE	147000	316	η		N shft spec
CF	147400	317	Ω		O shft spec
D0	150000	320	⊓		P shft spec
D1	150400	321	∩		Q shft spec
D2	151000	322	⊥		R shft spec
D3	151400	323	Σ		S shft spec
D4	152000	324	Θ		T shft spec
D5	152400	325	∣		U shft spec
D6	153000	326	∱		V shft spec
D7	153400	327	∧		W shft spec
D8	154000	330	⌇		X shft spec
D9	154400	331	⌜		Y shft spec
DA	155000	332	—		Z shft spec
DB	155400	333	⌊		[spec
DC	156000	334	⇒		\ spec
DD	156400	335	⊢		] spec
DE	157000	336	—	(*)	∧ spec
DF	157400	337	•	(*)	___ spec
E0	160000	340	␣ (blinking)		@ shft spec
E1	160400	341	∝		A spec
E2	161000	342	β		B spec
E3	161400	343	▽		C spec
E4	162000	344	δ		D spec
E5	162400	345	ε		E spec
E6	163000	346	ø		F spec
E7	163400	347	γ		G spec
E8	164000	350	⊥		H spec
E9	164400	351	ζ		I spec
EA	165000	352	⧹		J spec
EB	165400	353	⟋	(*)	K spec
EC	166000	354	λ		L spec
ED	166400	355	μ		M spec
EE	167000	356	ν		N spec
EF	167400	357	ω		O spec
F0	170000	360	π		P spec
F1	170400	361	ϐ		Q spec
F2	171000	362	ρ		R spec
F3	171400	363	σ		S spec
F4	172000	364	τ		T spec
F5	172400	365	ↄ		U spec
F6	173000	366	○		V spec
F7	173400	367	⟋		W spec
F8	174000	370	⁞		X spec
F9	174400	371	∠		Y spec
FA	175000	372	⍀		Z spec
FB	175400	373	⌜		[spec shft
FC	176000	374	∣		\ spec shft
FD	176400	375	⊣		] spec shft
FE	177000	376	~	(*)	∧ spec shft
FF	177400	377	▥		DEL spec

(*) superscript

APPENDIX B

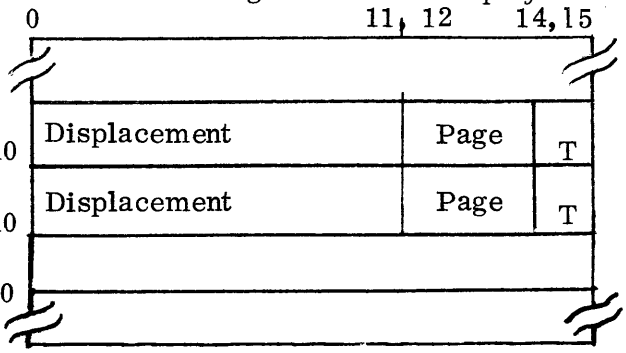
DISPLAY SUBROUTINE/STACK OPTION

The facilities described in this section are available as an optional extension to the display-computer interface in certain cases. These normally comprise those systems in which the display is not interfaced through a standard DMA data channel--in which case, the interface must implement the core-access and data channel functions.
The provision for the following extensions is made possible by the accessability of the host processor Memory-address register used for core data - word transfers.
The following facilities can all be implemented in the display driver programs and encoded in the display lists using P-bit interrupt calls as described in the Display System Reference Manual.

The advantage of hardware implementation is improved display speed, reduced processor execution time requirements and reduced core storage requirements for driver coding.

DISPLAY REGISTERS

This option provides the following three additional 15-bit registers in the display controller.



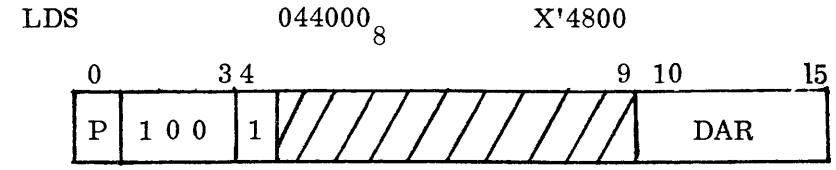
The display controller, MAR, holds the core address normally used when words are sent from memory to the display. Address registers are extended to 15 bits to permit addressing of 32K words. The low order 12 bits (displacement) are in the value field to permit address arithmetic within 4K pages. The address can be extended above 32K by the use of extension bits contained in the PIOX format (see section 3.7). Addresses are automatically incremented up to 32K boundaries but not through them. The MAR normally holds the address of the next display-list word to be processed. After use, it is incremented.

The SPR holds the core address used whenever words are sent from Display Registers back to memory. It can also be used to fetch words back to display system registers. Prior to use, the SPR is incremented on store operations. After use on fetch operations, the SPR is decremented.

The TGR is a general purpose 15-bit register useful for temporary storage of any display register, including the MAR and SPR. As with all other registers, the MAR, SPR and TGR can be loaded, Or-ed, And-ed, or Add-ed to, over the full 15 bits, via Display List Register Operations.

DISPLAY INSTRUCTIONS

Load from Stack



The load from Stack display instruction extracts the value field from a list of words in core and loads them into succeeding display system registers, starting with the one designated by the DAR field (DAR assignments are given in Section 3.21)

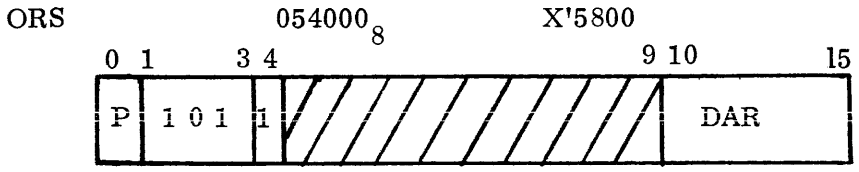
The list of values is in successively preceeding (lower address) cells of memory.

The first value (highest address) is in the cell initially addressed by the contents of SPR (bits 0 to 14).

The list is terminated by a word with bit 15 set to one (terminate).

After each word is transferred, its address held in SPR, is decremented.

Or from Stack



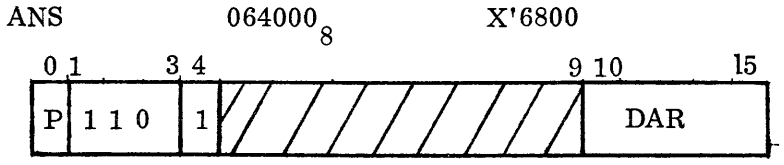
The Or from Stack display instruction extracts the value field from a list of words in core and OR's them into succeeding display system registers, starting with the one designated by the DAR field.

The first fetched word of the list, stored at the highest core location of the list, is at the address held in SPR.

The list extends through preceeding (lower) addresses to one containing a terminate bit (15) set to one.

The SPR is decremented after each word is fetched.

And from Stack



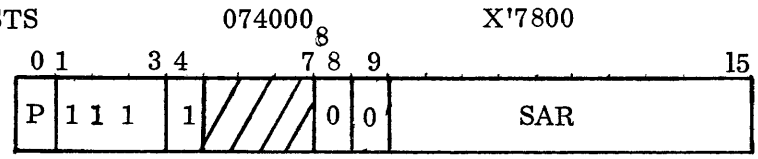
The And from Stack display instructionextracts the value field from a list of words in core and AND's them into succeeding display system registers, starting with the one designated by the DAR field.

The first fetched word of the list, stored at the highest core location of the list is at the address held in SPR.

The list extends through preceeding (lower) addresses to one containing a terminate bit (15) set to one.

The SPR is decremented after each word is fetched.

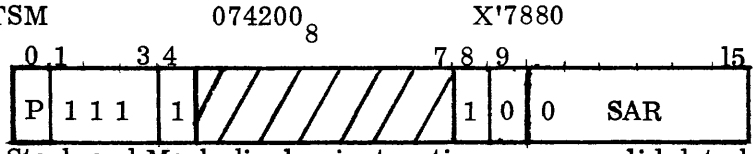
Store in Stack



The Store in Stack display instruction causes valid data bits in the display system register designated by instruction field SAR to be stored into the memory word addressed by the contents of SPR after incrementing it by one.
Note: SAR must be less than 64_{10} and

If SAR = 14_{10}, (MAR), the MAR contents are incremented by 2 words prior to storing.

Store in Stack and Mark



The Store in Stack and Mark display instruction causes valid data bits in display register at SAR to be stored into memory with bit 15 set = 1 terminate.

Store in Stack and Mark - Continued

The SPR is incremented and the result used as the core address into which the value is stored.

Notes: SAR must be less than 64_{10} and
If SAR = 14_{10}, the MAR is incremented by two words prior to storing.

EXAMPLES

The following series of display-code segments illustrate how the optional display-instructions may be used to code some desirable display functions.

Branch in list

```
JUMP (LOC)    LD  MAR
              Loc T
```

This sequence may be used to link disjointed display-list segments for processing as if they were one continuous display definition.

Subroutine list-jump

```
CALL (LIST)   STSM  MAR
              LD    MAR
              LIST  T
```

This sequence permits a single picture-defining display list to be processed many times as a sub-item used in defining a composite display.

This sequence also permits composite display-lists (containing sub-list call's) to be called as sub-lists of another display-list. Thus user defined displays may be used as basic elements in defining further displays.

Subroutine Exit

```
RETURN    LDS MAR
```

This display instruction can be used to terminate a sub-list definition and return to the calling list to resume its display generation, (if it was called as by the previous "CALL" sequence).

Stuff Data (Store "n" registers starting at register "REG".)

```
PUSH (Reg, n) =   STSM  REG + (n-1)
                  STS   REG + (n-2)
                   ⋮
                  STS   REG
```

This sequence may be used for nested saving and restoring of register data.

This need arises in making display-lists transparent to the effect of sub-list calls (i.e. alteration of coordinate registers, pen enable/detect, transformation state, etc)

Another use is for bracketing the effect of transformations over selected sequences of display-list items, and/or the nesting of such transformation effects.

Restore Data

```
POP = LDS REG
```

This display instruction will restore the registers saved by the preceeding (matching) PUSH operation.

Save Data

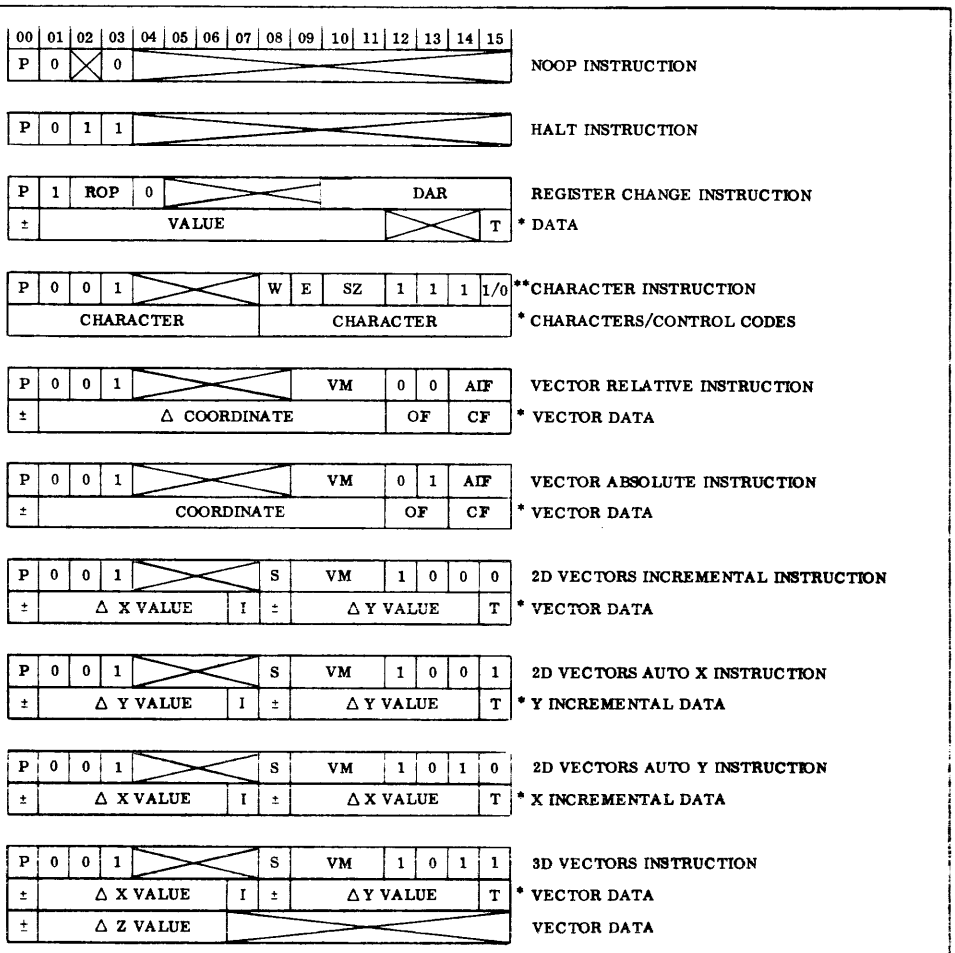
```
INPUT (REG, n, TABLE) = STSM    TREG
                        STSM    SPR
                        LDS     TREG
                        LD      SPR
                        TABLE   T
                        STSM    REG + (n-1)
                        STS     REG + (n-2)
                        :
                        STS     REG
                        STSM    TREG
                        LDS     SPR
                        LDS     TREG
```

This sequence may be used to read a set of successive display registers into core without using programmed PIO, interrupt processing, or any display driver coding execution.

In the example as given, the input buffer TABLE has an extra word used in restoring the stack-pointer, SPR; the temporary register, TREG, must be any display register with sufficient low-order bits to hold the memory addresses used.

APPENDIX C

INSTRUCTIONS AND DATA FORMATS



* REPEAT UNTIL TERMINATE DECODED.
** 1 = NO SLANT, 0 = SLANT (OPTION).

AIF	AUTO-INCREMENT FIELD
CF	COORDINATE FIELD
DAR	INITIAL DESTINATION ADDRESS REGISTER
E	ENABLE CHARACTER SIZE CHANGE
I	INTENSITY FIELD
OF	OPERATION FIELD
P	STOP AND INTERRUPT ON TERMINATE
ROP	REGISTER OPERATION
S	INCREMENT SCALE
SZ	CHARACTER SIZE FIELD
T	TERMINATE FIELD
VM	VECTOR MODE FIELD
W	CHARACTER WRITE DIRECTION

	I	W	T	S
0	MOVE	HORIZ	CONTINUE	LO ORDER Δ
1	DRAW	VERT	TERMINATE	HI ORDER Δ

	VM
000	LINES
001	DASHES
010	DOTS
011	POINTS
101	DASH-DOT-DASH
110	DASH-DOT-DASH

	AIF	ROP	OF	CF	SZ
00	NO AUTO INCR	LOAD	LOAD (NO STEP)	AIR	120 COL X 60 LINES
01	STEP XR	OR	LOAD & DRAW	XR	81 COL X 41 LINES
10	STEP YR	AND	LOAD & MOVE	YR	60 COL X 30 LINES
11	STEP ZR	ADD	LOAD & DRAW & TERMINATE	ZR	32 COL X 16 LINES

OFCF = 0100: ENTER CLOCKWISE ARC MODE.
1000: ENTER COUNTERCLOCKWISE ARC MODE.
1100: ENTER CLOCKWISE ARC MODE.
(FOLLOWING DATA WORDS SET CENTER POINT BEFORE DRAW IS STARTED).

101061-98

APPENDIX D

DISPLAY MNEMONICS AND CODES

Table D-1 and D-2 list the operation codes and variable field codes of the display system instructions and data with the applicable page numbers where the items are discussed.

Table D-1 lists the display instructions, the mnemonics, variable fields, and the field formats of their data words. The instruction codes are given in both octal and hexadecimal notation. The variable fields for both the instructions and the data words are given in Table D-2.

The correct code for the instruction desired can be determined by using the code given for the instruction and adding the code listed for the variable field to be used. For example, if a Vector Relative instruction is to be used with the variable field VM in the dot mode, the hexadecimal code for the Vector Relative instruction, X'1000', is obtained from Table D-1; the hexadecimal code for the variable field VM in the DOT mode, X'0020', is obtained from Table D-2. The resulting code would then be:

```
X'1000'
X'0020'
--------
X'1020'
```

Table D-1. Operation Codes

Display Instruction	Mnemonic-Fields/Data	Codes	
No Operation	p NOP/-	000000	X'0000'
Vector Relative	p VR vm/+ Δ of cf	010000	X'1000'
" Auto-X	p VR IX vm/± Δ of cf	010001	X'1001'
" Auto-Y	p VR IY vm/± Δ of cf	010002	X'1002'
" Auto-Z	p VR IZ vm/± Δ of cf	010003	X'1003'
Vector Absolute	p VA vm/± Value of cf	010004	X'1004
" Auto-X	p VA IX vm/± Value of cf	010005	X'1005'
" Auto-Y	p VA IY vm/± Value of cf	010006	X'1006'
" Auto-Z	p VA IZ vm/± Value of cf	010007	X'1007'
Incremental 2D	p DVXY s vm/Δx i Δy t	010010	X'1008'
" Auto-X	p DVYY s vm/Δy i Δy t	010011	X'1009'
" Auto-Y	p DVXX s vm/Δx i Δx t	010012	X'100A'
Incremental 3D	p DV3D s vm/Δx i Δy t/Δz	010013	X'100B'
Character	p CH w sz/ch ch	010017	X'100F'
Special No Operation	p SPC/-	020000	X'2000'
Halt	p HLT/-	030000	X'3000'
Load Registers	p LD dar/Value t	040000	X'4000'
OR to Registers	p OR dar/Value t	050000	X'5000'
AND to Registers	p AN dar/Value t	060000	X'6000'
ADD to Registers	p AD dar/Value t	070000	X'7000'

INSTRUCTION FIELDS

Instruction	Field Name (Mnemonic)	Value (Mnemonic)	Codes	
All	Interrupt Request (1)			
	Continue		000000	X'0000'
	Halt and Interrupt	*	100000	X'8000'
VR VA DV	Vector Mode (vm)			
	Lines	--	000000	X'0000'
	Dashes	DSH	000020	X'0010'
	Dots	DOT	000040	X'0020'
	Point	PNT	000060	X'0030'
	Dash-dot	DDOT	000120	X'0050'
DV	Increment Scale(s)			
	No Magnification	--	000000	X'0000'
	Magnify	M	000200	X'0080'
CH	Character Write Direct. (w)			
	Horizontal	--	000000	X'0000'
	Vertical	V	000200	X'0080'
CH	Character Size Control (sz) (SE = 1)			
	Use Previous	--	000000	X'0000'
	120 x 60	S0	000100	X'0040'
	80 x 40	S1	000120	X'0050'
	60 x 30	S2	000140	X'0060'
	32 x 16	S3	000160	X'0070'
LD OR AN AD	Destination Address Register (dar)			
	Function lights (hi)	LTH	000000	X'0000'
	Function lights (lo)	LTL	000001	X'0001'
	Interrupt Request & Name	PIR	000004	X'0004
	Mode Control	MCR	000005	X'0005'
	X Coordinate	XR	000010	X'0008'
	Y Coordinate	YR	000011	X'0009'
	Z Coordinate	ZR	000012	X'000A'
	Auto-Increment	AIR	000013	X'000B'
	Dimming	IOR	000014	X'000C'
	Depth Cueing	ISR	000015	X'000D'
	Fetch Addr	MAR	000016	X'000E'
	Stack Ptr	SPR	000017	X'000F'
	Temp	TGR	000020	X'0010'
	Picture Scale	PSR	000021	X'0011'
	Name	NMR	000022	X'0012'
	Coordinate Scale	CSR	000023	X'0013'
	X Displacement	DXR	000024	X'0014'
	Y Displacement	DYR	000025	X'0015'
	Z Displacement	DZR	000026	X'0016'
	Rotation Matrix	R11R	000027	X'0017'
	Rotation Matrix	R12R	000030	X'0018'
	Rotation Matrix	R13R	000031	X'0019'
	Rotation Matrix	R21R	000032	X'001A'
	Rotation Matrix	R22R	000033	X'001B'
	Rotation Matrix	R23R	000034	X'001C'
	Rotation Matrix	R31R	000035	X'001D'
	Rotation Matrix	R32R	000036	X'001E'
	Rotation Matrix	R33R	000037	X'001F'

Table D-2. Variable Field Codes

DATA FIELDS

Instruction	Field Name (Mnemonic)	Value (Mnemonic)	Codes	
	Operation Field (of)			
VR VA	Load	L	000000	X'0000'
	Draw	D	000004	X'0004'
	Move	M	000010	X'0008'
	Draw and Terminate	DT	000014	X'000C'
	Coordinate Field (cf)			
VR VA	AIR	AI	000000	X'0000'
	XR	X	000001	X'0001'
	YR	Y	000002	X'0002'
	ZR	Z	000003	X'0003'
	Combined (of cf)			
	Clockwise arc	CW	000004	X'0004'
	Counterclockwise arc	CCW	000010	X'0008'
DV	Intensity Field (i)			
	Move	M	000000	X'0000'
	Draw	D	000400	X'0100'
DV	Terminate Field (t)			
LD, OR	Continue	--	000000	X'0000'
AN, AD	Last word of data	T	000001	X'0001'
	Mode Control Bits			
LD OR AN AD	Enable P-Bit Halt	MED	1000000	X'8000'
	Enable Frame Clock Interrupt	MEC	040000	X'4000'
	Enable Pen Hit Detect Interrupt	MEP	020000	X'2000'
	Enable Tablet Interrupt	MET	010000	X'1000'
	Enable Keyboard Interrupt	MEK	004000	X'0800'
	Enable Sense Switch Interrupt	MES	002000	X'0400'
	Enable Display Blink	MDB	001000	X'0200'
	Enable Light Pen Halt	MPH	000400	X'0100'
	Select Scope 1	MS1	000200	X'0080'
	Select Scope 2	MS2	000100	X'0040'
	Select Scope 3	MS3	000040	X'0020'
	Select Scope 4	MS4	000020	X'0010'

Table D-2. Variable Field Codes

APPENDIX E

2D WINDOW OPTION

Without Intercept Acquisition

This option provides registers for the specification and control of an unrotated rectangular 2D window. The option allows the affected part of the display to be blanked inside or outside the window. The window may also be enabled to generate an interrupt on every beam entry and/or exit. There is a "halt control" similar to the light pen halt, and readable sense bits to allow current beam position versus window determination. As with other register controlled functions, the window may be respecified and controlled differently for various parts of the display list.

1. Window Specification

   Four (4) 12-bit registers are added to the Controller for establishing the window boundaries.

   XHR Controls the most positive X boundary

   XLR Controls the most negative X boundary

   YHR Controls the most positive Y boundary

   YLR Controls the most negative Y boundary

2. Blanking Control

   The display can be blanked either within the window boundaries or outside of window boundaries. Two bits control the mode.

   MBI Specifies blanking inside the window

MBO specifies blanking outside the window

If both bits are One, the entire picture is blanked and if both bits are Zero, the entire picture is unblanked.

3. Interrupts

When a window boundary is crossed an interrupt will be generated if mode control MEI or MEO is set.

MEI enables interrupt PIW if crossing a window boundary while entering the window area

MEO enables interrupt PIW if crossing a window boundary while leaving the window area

PIW is reset by a program output with bit 8 true (AKW). Sense bit PWC is also reset by AKW.

4. Sense Bits

Sense bits are used to locate the CRT beam position

PWI beam is in the window area

PXH the current X coordinate is more positive than the XH register contents

PXL the current X coordinate is more negative than the XL register contents

PYH the current Y coordinate is more negative than the YH register contents

PYL the current Y coordinate is more negative than the YL register contents

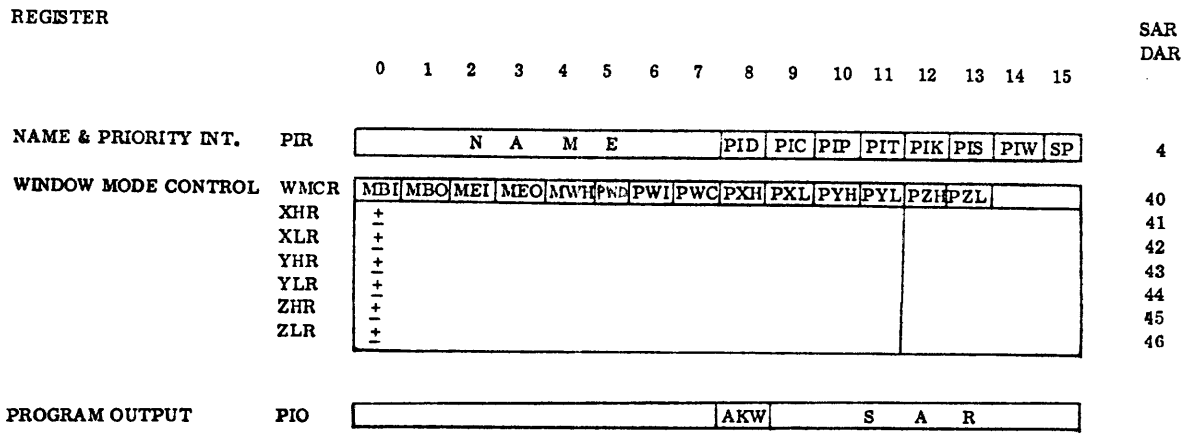
Status bit PWC indicates a boundary crossing during either a move or draw while PWD indicates a boundary crossing during a draw only. Both bits are reset at the start of each vector.

5. Halt Control

The display controller goes into a suspend mode when a window boundary crossing interrupt occurs if mode control MWH is true.

6. The listed sense bits (4 above) can only be read; all other bits in PIR WMCR and X-Y registers can be written by programmatic Direct Output or Register Operation list-instructions.

7. The PIR and WMCR bits (except sense bits 4. above) are cleared by System Reset signal.



GLOSSARY

ABSOLUTE VECTORS - the coordinates of absolute vectors are specified with respect to the zero position in the center of the Image Space (or screen for no transformations).

AUTOINCREMENTING VECTORS - the autoincrementing feature is used to step one coordinate at regular intervals while the other coordinate is open to program change.

CATHODE RAY TUBE (DM) - the cathode-ray tube generates an electron beam that shows as a spot of light on the face of the tube. An electromagnetic deflection system causes the spot to move in any direction on the tube face in response to signals from the vector generator. An input from the vector generator causes the brightness of the spot to vary and turns the spot off completely when desired.

CHARACTER GENERATOR - the character generator processes a data stream of ASCII* characters and generates the characters as text for the display. Any one of four sizes may be selected by the program.

CLOSURE - that measure of accuracy which indicates the closeness of a vector endpoint to a vector line segment.

CONTROL CHARACTERS - twelve codes in the character set are used for control purposes only and do not cause a display on the screen. The control characters and their functions are as follows: Null, Delete, Line Feed, Backspace, Formfeed, Carriage Return, DC1, DC2, DC3, DC4, Horizontal Tab, and Vertical Tab.

COORDINATE SCALE OPTION - to change the overall size of the image without changing its shape, a scale factor is loaded by the program into a 12-bit coordinate scale register.

COORDINATE TRANSFORMATION GENERATORS - three optional coordinate transformation generators are available.

- Two-dimensional for Scale and Translation (2D) - scales and translates two-dimensional constructs and displays them.
- Two-dimensional for Scale, Translation, and Single-axis rotation (2DR) - scale and translates two-dimensional constructs and displays them with rotation in a single plane.
- Three-dimensional (3D) - generates three-dimensional constructs and displays them with scaling, translation, and rotation about any axis.

---

* American National Standard Code for Information Interchange

GLOSSARY

DD - see Dual Digital to Analog Converter

DM - see Display Monitor

DMA - Direct Memory Access; see Data Channel

DATA CHANNEL - direct memory access channel used to output the picture being presented on the CRT screen.

DATA TABLET (DT) - the data tablet is a graphic input device with an X-Y coordinate grid which may be used corresponding to the grid on the CRT screen. Information is entered through the data tablet with a stylus.

DISPLACEMENT VECTOR OPTION - the displacement vector option performs the translation function in the image transformation feature by moving the image intact along any of the three axes.

DISPLAY CONTROLLER - the display controller processes all display functions, running asynchronously with the computer central processor. It also receives inputs from the external control devices.

DISPLAY MONITOR (DM) - see Cathode Ray Tube

DUAL DIGITAL TO ANALOG CONVERTER (DD) - the dual digital-to-analog converter (DD) is the standard version of the coordinate transformation generators; it converts the digital values from the display controller into analog signals for use in the vector generator.

END MATCHING - that measure of accuracy which indicates the closeness of two vector endpoints.

HARDWARE SCISSORING FACILITY - the ability to position an image out of the viewing area in any direction, without distorting any remaining visible portions, is accomplished by the Hardware Scissoring Facility.

IMAGE SPACE - prior to transformation and projection onto the Picture Space for viewing through the Visible Space, an object is defined in a coordinate system which is referred to as the Image Space.

IMAGE TRANSFORMATION - image transformation is an optional hardware feature that involves scaling, rotation, and position change (translation).

GLOSSARY

INCREMENTAL VECTORS - incremental vectors are used when data storage is limited. Data increments can be shorter than relative vector increments, with a resultant reduction in the amount of data needed. Incremental vector display, therefore, requires less data storage and improves performance by increasing the rate of output and presentation.

INTENSITY LEVEL - for two-dimensional display, 32 constant intensity levels can be selected by the program. These intensities can be applied to vectors and characters. The spot can be blanked as desired under program control during vector display.

INTENSITY MODULATION - intensity modulation is the name given to the depth cueing transformation used in all 3D systems, that shades the intensity of the displayed picture to give a 3-dimensional effect.

INTERACTIVE CONTROL DEVICES - six interactive control devices may be connected to the system: an alphanumeric keyboard (KB), 16 (or 32 optional) lighted function switches with manual interrupt (FS), a data tablet (DT), a joystick (JS), control dials (CD), and a light pen (LP).

INTERRUPT CHANNEL - used by display and device response interrupt to activate computer programs.

JOYSTICK (JS) - the joystick is a mechanical device used to enter coordinate values in the 12-bit joystick X, Y, and Z input registers.

LIGHT PEN (LP) - the light pen is used to point at an element of a display or to create information by "drawing" on the display. The light pen, a wand containing a photocell, is held over the face of the CRT by the viewer.

OPTIONAL FEATURES - the optional features are a character generator (CG), three coordinate transformation generators [two-dimensional (2D), two-dimensional with rotation (2DR), and three-dimensional (3D)], and any one of six interactive control devices [an alphanumeric keyboard (KB), 16 (or 32 optional) lighted function switches with manual interrupt (FS), a data tablet (DT), a joystick (JS), control dials (CD), and a light pen (LP)].

PIO - see Programmed I/O channel

PICTURE CONTROL OPTION - the picture control option is used for picture transformation after the transformation of individual images on the screen has been completed.

GLOSSARY

PICTURE SCALING - a hardware register is provided to hold a value that scales all the final transformed X and Y coordinate values. This scaling is used primarily to reduce full-scale, rotated, 3-dimensional images so that they fit into the display screen while permitting untransformed images, such as text pages and graphs, to fill the same display area.

PICTURE SPACE - this space, over which the vector generator accurately reproduces images, is a 30" x 30" plane of which the "Visible Space" is a 13" by 14" rectangle in the center.

PICTURE TRANSFORMATION - when a 3-dimensional image made up of characters and vectors has been transformed to obtain the desired scale, rotation, and translation, a 2-dimensional view can be extracted and presented as a picture on the display screen. The two operations involved in this final presentation are picture scaling, to change the size of the transformed image, and intensity modulation (optional) to give a 3-dimensional depth cueing effect.

PROGRAMMATIC INTERFACE - the interface between the display system and the computer consists of a single programmed I/O channel, a single priority interrupt level, and a single direct memory access channel.

PROGRAMMED I/O CHANNEL - used to start the controller, acknowledge interrupts, and provide access to the display controller and device registers.

PROGRAMMED INPUT - the method by which contents of the display registers are brought into computer memory.

PROGRAMMED OUTPUT - the method by which control and acknowledgement information is sent to the display; also used to set the Source Address Registrar (SAR).

RANDOM SCAN - the Vector General display uses the random scan method of controlling the movement of the spot. Random scan control involves steering the spot in a straight line between two points on the display screen. A series of these straight lines constitutes an image portion.

REFRESHED CRT - electronic regeneration of a picture image measured in frames per second.

RELATIVE VECTORS - the end-point coordinates of a relative vector are located with respect to the starting point coordinates. In other words, relative vector data is specified in the form of increments that are added to or subtracted from the previous coordinate values.

ROTATION - an image portion can be rotated around any of its axes by using the optional hardware rotation matrix. The desired rotation is specified by loading direction cosines, or the sums of triple products of trigonometric functions in the more elaborate cases, into the rotation matrix, which has registers for each coordinate axis.

SCALING - the scaling operation consists of changing the size of an image portion by multiplying each end-point coordinate by the desired scale factor before processing. The scale factor is specified by the program, and the current scale factor is maintained in a hardware register to be multiplied by the X, Y, and Z coordinate values.

SCISSORING - see Hardware Scissoring

SELF-TESTER - a hardware option designed to test both the display controller and monitor without using the host computer.

STANDARD FEATURES - the standard features are an interface unit, a display controller (DC), a dual digital-to-analog converter (DD), a vector generator (VG), and a cathode-ray tube (DM).

TABLET - see Data Tablet

TRANSLATION - an arbitrary image may be positioned anywhere in 3-dimensional space by adding a value to each of the scaled and rotated coordinate values every time an end point is specified. The valued added must be constant for each coordinate to maintain the original image configuration.

VG - see Vector Generator

VECTOR COORDINATES - the display system maintains the coordinates of the current position of the fluorescent spot in the 12-bit X- and Y-registers, with the inclusion of a 12-bit Z-register for the 3D option. These registers hold the X, Y, and Z coordinates respectively. The values in these registers are updated as new coordinate values are received on the data channel.

VECTOR GENERATOR (VG) - the vector generator accepts input from the coordinate transformation generators and uses it to present solid, dashed, or dotted lines between two positions on the display screen or to place a point at any given position.

VISIBLE SPACE - that rectangular portion of the CRT which can be viewed by a user.

WORD FORMAT - the display system uses as its basic informational elementa 16-bit word with the bit positions numbered 0 through 15.

21300 Oxnard Street • Woodland Hills, California 91364 • Phone: (213) 346-3410 • TWX: 910-494-2764