Digital Equipment Corporation
Maynard, Massachusetts

digital

PDP-15 Systems

Graphic-15
Reference Manual

PDP-15
GRAPHIC-15
REFERENCE MANUAL

CONTENTS

CONTENTS (Cont)

CONTENTS (Cont)

ILLUSTRATIONS

ILLUSTRATIONS (Cont)

TABLES

INTRODUCTION

The purpose of this reference manual is to familiarize the Graphic-15 Display System user with the purpose and use of the Graphic-15, describe the VT15 IOT instructions and display file instructions, and provide a brief description of programming at the machine-language level.

Additional reference manuals and maintenance manuals related to the Graphic-15 Display System are listed in the following table.

Title	Document Number
Graphic-15 Programming Manual	DEC-15-ZFSA-D
VT15 Graphic Processor Maintenance Manual	DEC-15-H2JB-D
VT04 Graphic Display Console Maintenance Manual	DEC-15-H2GA-D
PDP-15 Systems Reference Manual	DEC-15-BRZC-D
PDP-15 Interface Manual	DEC-15-HOAB-D
PDP-15 System Maintenance Manual	DEC-15-H2BB-D
PDP-15 Operator's Guide	DEC-15-H2CB-D
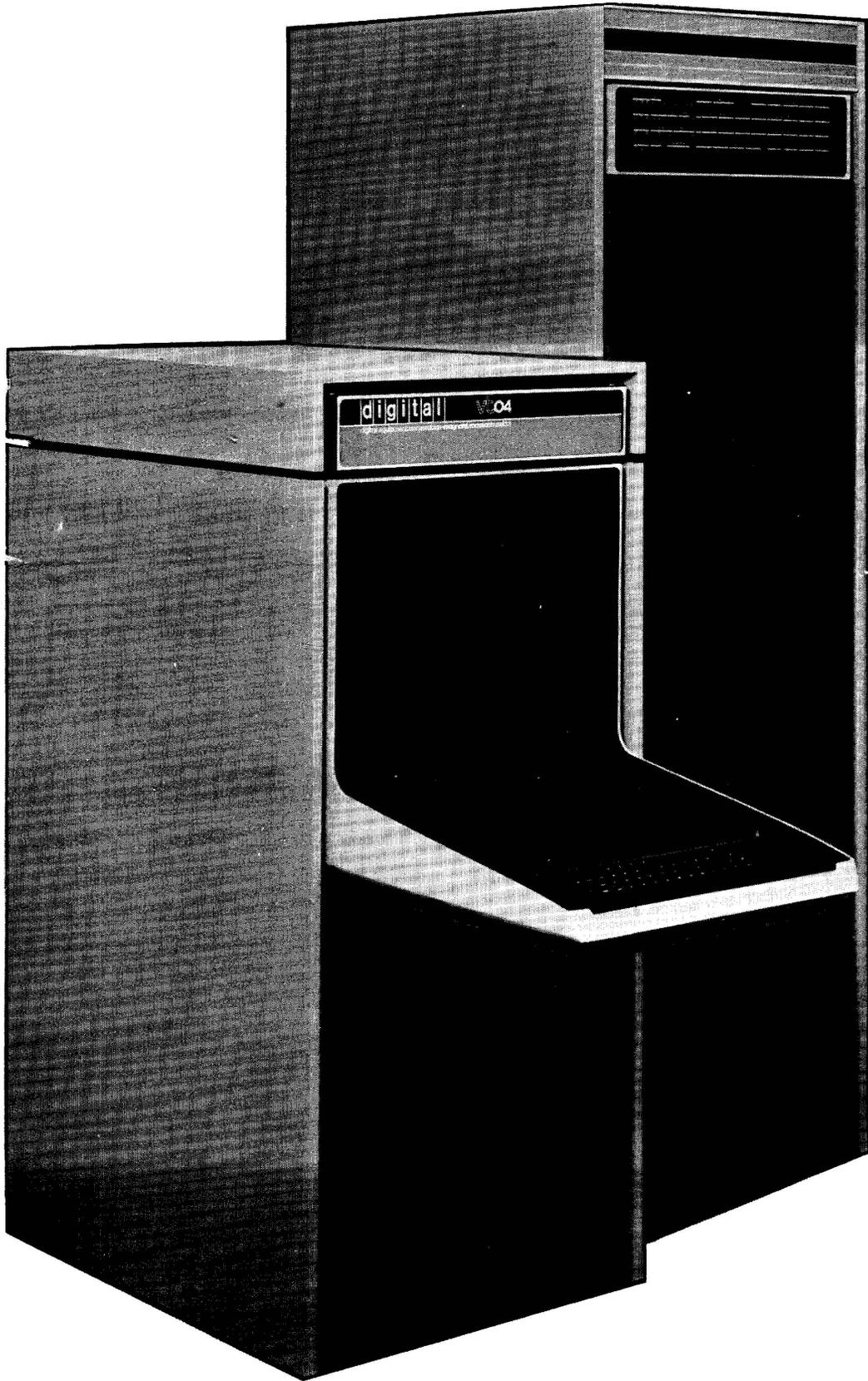
Figure 1-1   Type VT15 Graphic Display System

CHAPTER 1
GENERAL INFORMATION

1.1 GENERAL

This chapter provides a physical and brief functional description of the equipment comprising the Graphic-15 Display System and its interfacing arrangement with the PDP-15 Programmed Data Processor.

1.2 PURPOSE OF THE EQUIPMENT

Common applications of the Graphic-15 Display System permit immediate solution of a broad range of electrical, physical, and mechanical design and analysis problems. While there can be little doubt about the value of the computer in solving complex design problems, the graphic display system accelerates the exchange of data between the user and the computer. Visualization of a problem, whether it be a simple curve that displays the interaction of system variables or a complex network synthesis, permits rapid selection of new parameters or variables so that complex changes - the computer does the math - need not require laborious reprogramming. The VT15 Graphic Processor is an ancilliary to the basic PDP-15 computer that permits the user to create a visual model of an electrical, hydraulic, mechanical, or other physical system while the computer operates on its mathematical counterpart. Thus, parameters that define a point outside a curve can be immediately displayed as irrelevant data, the change in the cross section of a beam can immediately reveal its ability to withstand a given load, and a change in the value or placement of a resistor in a network synthesis can immediately demonstrate the relative stability of an electrical circuit.

1.3 SYSTEM DESCRIPTION

The Graphic-15 Display System (Figure 1-1) consists of a VT15 Graphic Processor and a VT04 or VT07 Graphic Display Console working with a PDP-15 Computer.

1.3.1 VT15 Graphic Processor

The VT15 Graphic Processor interacts with the PDP-15 computer to obtain its instructions. Because the PDP-15 computer and VT15 share the same core memory, their programs interact through the hardware. Furthermore, their programs interact to the extent that the PDP-15 computer program can directly modify the VT15 Graphic Processor program, which is called the display file. The display file consists of instructions and data upon which the VT15 operates to construct a desired graphic image on the display console CRT. The display file program in the PDP-15 computer core memory contains the instructions and data upon which the graphic processor operates and to which its digital control and analog outputting circuits respond. The VT15 has a set of 11 basic machine-language instructions that give the Graphic-15 Display System the utmost versatility in the display of points, basic vectors, graph plots, and ASCII characters.

The Graphic-15 Display System is a directed-stroke refreshed display system. The directed-stroke refresh system requires a local memory so that the display can be continually renewed or refreshed; the local memory used is the one contained in the PDP-15 computer. A directed-stroke system permits the graphic construction to be traced on a one-to-one basis so that real-time motions such as those that might be produced by an operator-controlled light pen or tablet can be quickly converted to an active CRT display.

An additional feature of the VT15 Graphic Processor is its hardware character generator. This character generator has a local read-only memory which permits any one of 68 ASCII-compatible characters to be generated at an average rate of 12 µs per character in a basic system. This high writing rate produces 80,000 characters per second on a basic Graphic-15 Display System. These characters are called by one of two instruction words in the VT15 Graphic Processor instruction set.

1.3.2 VT04 and VT07 Graphic Display Consoles

The VT04 Graphic Display Console contains the CRT monitor upon which the graphic images are displayed in accordance with the instructions and data operated upon by the VT15. In addition to the display CRT, this console also contains six operator-controlled pushbuttons that can be used to provide auxiliary functions. These pushbuttons may be programmed to produce skip signals, back to the PDP-15 computer or to the VT15, which permit exit from the current display file or entry into some other display file. The CRT monitor is a standard 17-inch cathode ray tube which uses electromagnetic deflection and electrostatic focusing.

The VT07 Graphic Display Console functions identically with the VT04. The CRT used in the VT07 is a standard 21-inch cathode ray tube.

1.4   EQUIPMENT OPTIONS

Options that are currently available with the Graphic-15 Display System and those that will be available in the future give the VT15 Graphic Processor a wide range of capability in many disciplines. These options and the Graphic-15 Software Library make the Graphic-15 the most versatile graphic display system available within its price structure.

1.4.1   VL04 Light Pen Option

The 370 Light Pen used in the VL04 is a photosensitive device used to detect light on the CRT screen when programmed to do so. The light pen is interfaced with the VT15 Graphic Processor through a connector on the display console. The light pen can be used to delete graphic constructions and to draw directly on the face of the CRT through suitable subroutines.

1.4.2   VV15 Arbitrary Vector Generator

The VV15 Arbitrary Vector Generator is a prewired plug-in option available with the VT15. This option permits any vector of arbitrary angle, to be drawn on the display console.

1.4.3   VW01 Sonic Digitizer Writing Tablet

The VW01 is an interactive digitizing device that permits the operator to input graphic instructions directly. This option is an excellent auxiliary for graphic designs.

1.4.4   LK35 Keyboard Option

The LK35 Keyboard option is a send-only keyboard, through which an operator at a remote display console station can enter data into the PDP-15 computer. The keyboard is the link between the operator, the VT15 Graphic Processor and the PDP-15 computer. Instructions or data are entered into the PDP-15 computer through the send-only keyboard, are processed by the computer, and stored in core memory as display-file data. A subroutine permits any characters struck on the send-only keyboard to be displayed on the CRT screen.

1.4.5   VM15 Display Console Multiplexer

The VM15 Display Console Multiplexer permits up to four display consoles and four VL04 Light Pen options to be interfaced with a single VT15 Graphic Processor. This option permits these equipments to be situated at remote locations and to share the use of the VT15 Graphic Processor and PDP-15 computer.

1.5 FUNCTIONAL DESCRIPTION

The VT15 Graphic Processor converts digital instruction inputs from a PDP-15 computer into analog signals that drive the X- and Y-axis deflection circuits of the display CRT. The digital inputs required by the graphic processor are entered into its input buffer directly from memory via the PDP-15 single-cycle word transfer facility when instructions are being transferred from the display file in the PDP-15 memory. This program consists of instructions to the VT15 Graphic Processor that tell the graphic processor what to do with the display file data. Because the graphic processor includes a hardware character generator, text characters or text strings can be ordered directly from the program by specifying the wanted ASCII characters.

The basic Graphic-15 block diagram is shown in Figure 1-2. Although the VT15 Graphic Processor uses PDP-15 program-controlled transfer facilities for initializing instruction transfers, for computer skip testing, and for read status functions, the basic mode of data and instruction transfers is the single-cycle word transfer facility of the PDP-15 computer.

To perform a single-cycle word transfer, the address register is contained in the interfaced peripheral device, which in this case, is the program counter (PC) in the VT15. Operation

Figure 1-2   Basic Graphic-15 Display
System, Block Diagram

of both the PDP-15 and the VT15 is interactively-controlled through the machine language instructions in the PDP-15 and by various functions of the VT15. For some interactive processes, the PDP-15 generates the display file. When the VT15 is commanded to perform some display function, a programmed display file stored in the PDP-15 core memory is entered through an initializing program-controlled input/output transfer (IOT). Once the display file has been initialized, operation is turned over to the graphic processor, which functions autonomously, with respect to the PDP-15, and frees the PDP-15 for computing or I/O tasks. The operation of the VT15 is asynchronously controlled, but interacts with the PDP-15 computer through the data-channel single-cycle data break and input/ output transfer (IOT) facilities. The VT15 Graphic Processor (Figure 1-3) has five major functional groups.

        a.   Data collectors

        b.   Timing and control

        c.   Position registers

        d.   Digital-to-analog converters

        e.   Analog function generator.

Figure 1-3   VT15 Graphic Processor, Simplified Block Diagram

These five major groups of the VT15 control the interactive functions of the PDP-15 computer, the VT15, and the generation of the graphic display called for in the display file. The functions listed in a. through d. above deal with the digital data supplied from the PDP-15 computer over the input/ output (I/O) bus. The analog function generator converts the inputted digital data to an analog output which is used by the display console to generate the graphic display.

1.5.1   Data Collectors

The data collectors have a bus receiver interface which receives outputs from the PDP-15 computer I/O bus and supplies them to the VT15 or receives outputs from the VT15 and supplies them to the PDP-15 I/O bus. Also included in the data collectors are two registers called the input buffer register and the data buffer register. The input buffer is a holding register that receives inputs from the I/O bus data lines, during data-break transfers, and retains them until called for by the VT15 timing and control function. This input buffer register is under control of the PDP-15. When called for, the data held in the input buffer register is transferred to the data buffer register. The data buffer register, under control of the VT15, is an active register that receives the input buffer register contents and operates on them according to the timing and control function demands.

1.5.2   Timing and Control

The timing and control processes the digital data held in the data collector data buffer register. Timing and control functions include: instruction and IOT decoding; updating of the PC register; determination of the graphic construction according to the contents of parameter and skip register, and direction and rotate logic; and updating of X- and Y-position registers.

Data processing functions are controlled by main timing circuits that are operated by a basic clock pulse having a 250 ns repetition rate. Each 250 ns interval is called a time state. In general, three time states of 250 ns are required to accomplish instruction-word processing functions. Instruction words and signals are processed as follows during the three active time states:

Time state 1 - is used to increment the display file program counter and to load the other registers of the control logic.

Time state 2 - is used for modification of the X-position register.

Time state 3 - is used for modifications of the Y-position register.

Modification of the X- and Y-position registers is accomplished through a digital adder in the main timing and control that is operated by a gating network controlled by main timing logic. This gating network performs a switching function that enables data to be processed during each time state. During the first time state, the PC is incremented through an adder. When this is done, the data currently held in the program counter, Cur PC Data, is summed with the incrementing bit and the content of the PC is updated to become New PC Data. During the second time state, X-Pos Data, is summed with the contents of a third or delta (Δ) register, New Δ Data, and becomes New X Data. During the third state, the Y-position register is updated so that the current content of the Y-position register, Y-Pos Data, is summed with the new Δ data and becomes New Y Data. There are seven instructions that cause the contents of the X- and Y-position registers to be modified; the Point/Graph Plot instruction, the Basic Vector instruction, Basic Short Vector, Character Input, Character String, Arbitrary Long Vector, and Arbitrary Short Vector.

When a point is plotted, the point-plot version of the Point/Graph Plot instruction is used. This instruction establishes the coordinates position of the point in both the X- and Y-axes in separate data entries, one for each axis. When one of these instructions is processed for a point plot, the 10-bit coordinate position digital data is transferred into the position register and the digital register of the X- or Y-axis digital-to-analog converter (DAC). Each axis movement requires from 8 to 20 μs. When the graph-plot version of this instruction is used by the programmer, he specifies only one axis in this version of the instruction; the other axis will be automatically incremented. When the instruction is processed by the VT15 Graphic Processor, the 10-bit coordinate position digital data of the specified axis is transferred into the position register and the register of the specified axis digital-to-analog converter, but the digital register of the unspecified axis analog-to-digital converter is updated with the digital value of some fixed increment previously chosen by the programmer. This version of the instruction need only be used once to specify both coordinates.

When using a basic vector instruction the programmer specifies any one of eight directions and the magnitude of the displacement of the line along either the X- or Y-axis. When a basic vector instruction is processed, the contents of the X- and Y-position registers are modified according to the

requirements of the programmer-specified three direction bits (part of the instruction) and the displacement of the wanted vector along either axis in the 10-bit Δ data word (also part of the instruction). If, for example, the basic vector is to be drawn at a 45° angle to the X axis with its start at the origin, the adder will sum the Δ word with the contents of both X- and Y-position registers because for a 45° angle, the displacement of the line along both axes is the same. During time state TS2, the X-position register gets modified by the Δ word and during time state TS3, the Y-position register gets modified by the Δ word.

1.5.3 Position and Δ Registers

There are three data holding registers, a position register for each axis and a magnitude of change (Δ) register. The purpose of the two axis-position registers is to hold the binary equivalent of the current position of the CRT writing beam. The purpose of the Δ register is to hold the binary equivalent of the relative displacement of the writing beam. Digital, rather than analog, holding registers are used because they are not subject to drift; they retain their data indefinitely until needed. When programming basic vectors or basic short vectors, the programmer must express a line with respect to its relative displacement from the last named position whose digital coordinate equivalents are always held in the X- and Y-position registers. The data that defines the magnitude of the relative displacement of the wanted new position from the old position, along either axis, is transferred to the Δ register during the first time state. It is the new Δ word that is always summed with the current position data held in the X- and Y-position registers.

During the given time state, the Δ word held in the Δ register may be added to or subtracted from the content of the X-position, the Y-position or both registers on the dictates of the programmer. The outputs from the position and Δ register, Δ word, X word, and Y word are applied to the digital-to-analog converters.

1.5.4 Digital-to-Analog Converters

There are three digital-to-analog converters (DACs) in the basic VT15. The purpose of the DACs is to convert the binary data word held in each of the position registers to a voltage whose magnitude is proportional to the magnitude of the digital data word. The digital-to-analog converters produce three analog voltages, one from each of the two position registers and the Δ register, which are fed to the analog function generator (AFG).

1.5.5 Analog Function Generator

The basic analog function generator (AFG) contains a comparator and a ramp generator. The ramp generator is used to drive the X- and Y-CRT deflection circuits. The ramp generator is used whenever the programmer specifies a basic vector, basic short vector or characters. This generator produces a ramp whose voltage causes the CRT beam to deflect at a minimum rate of 1/4 inch per microsecond. The instantaneous magnitude of this ramp voltage is compared with the analog equivalent of the digital content of the Δ register. When the ramp and Δ analog voltages are equal, the ramp generator is disabled. Both positive and negative values of the ramp voltages are produced, and depending on the direction of the wanted basic vector or basic short vector, the positive or negative voltages are summed with the X or Y DAC voltage to produce the CRT deflection voltages. When a point is desired by the programmer, the voltages equivalent to the contents of the X- and Y-position registers are summed in the analog function generator output drivers. The writing beam is deflected to the point on the CRT equivalent to the magnitudes of these two voltages.

Addition of the VV15 Arbitrary Vector Generator adds circuitry to modify the slope of the output ramps and therefore allows vectors other than the basic eight directions.

CHAPTER 2
VT15 GRAPHIC PROCESSOR INSTRUCTIONS

2.1 INTRODUCTION

This chapter describes how to use the basic VT15 Graphic Processor instruction set to generate machine language programs.

2.2 DISPLAY CAPABILITY

The Graphic-15 Display System has two primary drawing capabilities: point-plot and basic vector. The Graphic-15 is also capable of displaying text characters by use of the proper instruction. Characters are generated through a special instruction format which permits the generation of single characters or character strings.

Using the basic graphic instruction set, points, basic lines, and characters or character strings can be outputted from the graphic processor for display on the CRT of the display console. The display can be annotated with text at the discretion of the programmer.

The VV15 Arbitrary Vector option allows lines of arbitrary angles to be generated.

2.3 THE VT04 CRT DISPLAY

The CRT X-Y display (Figure 2-1) has a screen whose dimensions are 9-1/4 in. by 10-1/2 in. with horizontal plane (X-axis) being 10-1/2 in. in length. This screen is divided into two basic display (or image) areas. The major (largest) image area is a 9-1/4 in. square single-quadrant cartesian coordinate system whose origin is located at the lower lefthand corner of the visible portion of the display console screen. This square has 1024 addressable coordinate positions in both the X- and Y-axis planes. The minor (auxiliary) image area is a rectangular coordinate system whose effective display area is about 9-1/4 in. by 1-1/2 in. with the longer dimension in the vertical (Y-axis) plane. The basic auxiliary image area is divided into 1024 addressable coordinates along the Y-axis and approximately 161 addressable coordinates along the X-axis. The basic major image area represents a square paper area 9-1/4 in. by 9-1/4 in.; this area is the basic paper size. Through the use of a 2-bit paper

size field, in one of the input/output (IOT) instructions, this area can be increased to four times the basic 9-1/4 by 9-1/4 in. area to 37 by 37 in. as shown in Figure 2-2. The numbers in boxes indicate the binary word for each virtual paper expansion and the numbers in parentheses indicate the absolute value of the coordinate positions at the corners. Each increase in paper size provides a corresponding increase in the number of coordinate positions along each axis. When doubling the paper size 2048 coordinate positions are contained in each axis; and when doubled again, 4096 coordinate positions are contained in each axis.
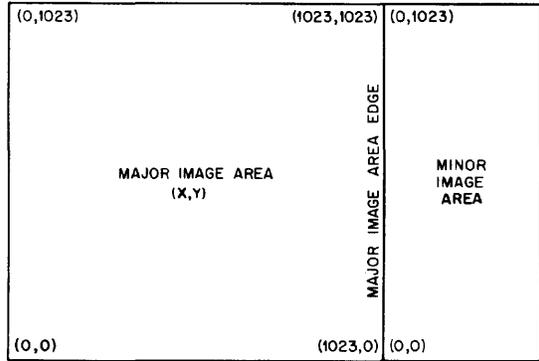
Figure 2-1   Visible Image Areas Display CRT

2.3.1   The Addressable Coordinates

Coordinates that specify points are expressed as absolute in terms of their displacement from the origin. Coordinates that specify vectors are expressed as relative in terms of their displacement from the last named coordinate positions. Any point within the major image area is defined both by its X-axis and Y-axis coordinates. Basic vector direction indicators permit vectors to be drawn in any one of eight directions. In machine language programming, the characteristics of the desired graphic form, such as character, point, graph plot, or basic vector are determined by the instruction used. Each of these instructions has a data field that permits an address (for characters), a position (for point or graph), or a magnitude (for basic vectors) to be expressed. Other characteristics of the wanted display are expressed through a parameter instruction.
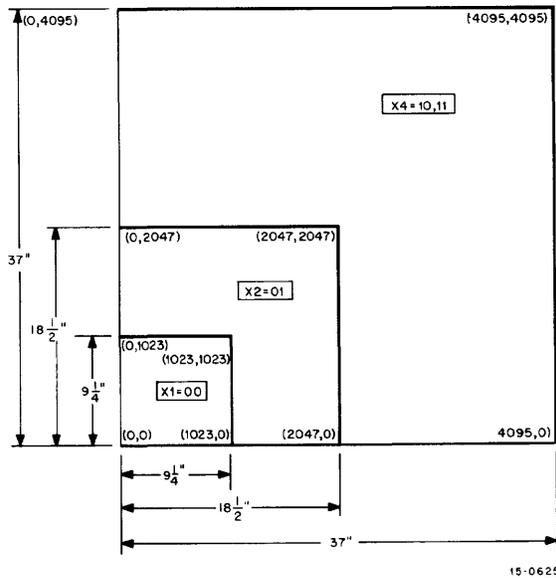
Figure 2-2   Relative Paper Size Dimensions

2-2

2.3.2 Visible Image Area

Although the entire CRT screen can be used to produce any desired graphics form which the VT15 is capable of drawing, only 86 square in. of this area (the major image area) can be used unless programmable modifications are made. These programmable modifications include changing the virtual paper size and/or using the minor and major image areas together. The ordinary basic data word that defines vector length or a coordinate position contains 10 binary bits, which permits a maximum length of 9-1/4 in. of viewing area to be used when generating a vector or defining a point. Vectors longer than 9-1/4 in. can be easily programmed. It is possible to program vectors up to 138-3/4 in. in length. They may be visible or invisible depending on how the programmer uses the available alternatives.

2.4 INPUT/OUTPUT TRANSFER (IOT) INSTRUCTIONS

There are 17 input/output transfer (IOT) instructions for the VT15. They are used by the programmer to initiate operation, execute programmed skips on the occurrence of some prescribed event or flag, to permit the programmer to read the status of flags, read the contents of the PC, and to read the contents of the X- and Y-axis position registers.

Of the 17 IOT instructions, six are read IOTs, seven are skip IOTs, and four are VT15 operate instructions. The read IOTs allow the programmer to read the status of various flags and registers into the PDP-15 accumulator. Access to these flags and register contents allows the programmer to write decision-making subroutines based on their status. Read IOTs are particularly useful when the programmer wishes to perform some program function other than skipping or interrupting. The read IOTs are also useful for trouble-shooting.

Skip IOTs cause the next instruction in the PDP-15 program (display file) to be skipped when certain hardware-controlled events have occurred. The occurrence of an event is signaled by a raised flag (a flip-flop is set).

2.4.1 IOT Instruction Format

The IOT instruction is a standard 18-bit word used by the PDP-15 computer to elicit specific responses from the VT15 Graphic Processor. This 18-bit instruction (Figure 2-3) contains four basic multi-bit parts: the computer operation code, the device selection code, the subdevice selection code and the input/output pulse (IOP) code.
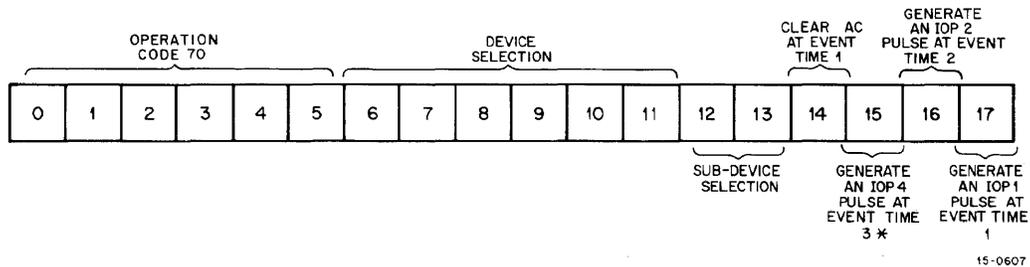
Figure 2-3   IOT Instruction Format

2.4.1.1   Computer Operation Code - The computer operation code uses the first four bits of the IOT instruction word; when an IOT is specified, bits 0, 1, and 2 are always set and bit 4 is always a zero (unset). Because bits 4 and 5 are unused, the total 6-bit word is always 111 000 (or 70_8).

2.4.1.2   Device Selection Code - The next six bits (6 through 11) of the IOT instruction word are the device selection code. The 6-bit length of this code permits up to 64 peripheral addresses. However, some peripheral devices require two or more codes because they perform a large number of discrete operations. For example, there are two device codes for the VT15; 011 000 and 011 011 (30_8 and 31_8).

2.4.1.3   Subdevice Selection Code - The next two bits (12 and 13) of the IOT instruction are the sub-device selection code. In the VT15 Graphic Processor these bits are used to increase the number of discrete IOT operations.

2.4.1.4   Accumulator Clear - One of the operations possible with the IOT instruction format is to clear the PDP-15 accumulator. This is done when bit 14 is set; this bit should be set when the read status IOT instructions are issued by the programmer or a logical "OR" of the data and ac contents will result.

2.4.1.5   IOP Code - The last three bits (15, 16, and 17) of the IOT instruction are the IOP codes. Setting any one of these three bits separately or in combination obtains a specified response from the VT15. These bits control a timing generator in the PDP-15 computer that provides serial output pulses on the input/output (I/O) bus which are used by the VT15 to perform its operation. For each of the output pulses, a separate response is provided by the VT15 depending on the IOT device codes.

2-4

2.4.2 Read Status 1 (RS1)

The Read Status 1 IOT (703002 or 703012 for AC CLR) is used to read the status of various flags and registers into the PDP-15 accumulator. The accumulator is loaded with the special format 18-bit word shown in Figure 2-4. The significance of each bit is listed in Table 2-1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED		OFFS.	ROT.	BNK	PA 0	PA 1	INCR 0	INCR 1	INCR 2	INCR 3	STOP INTR EN	LP INTR EN	EDGE INTR ENA	PB INTR EN	LP FLAG EN	LINE 0	LINE 1
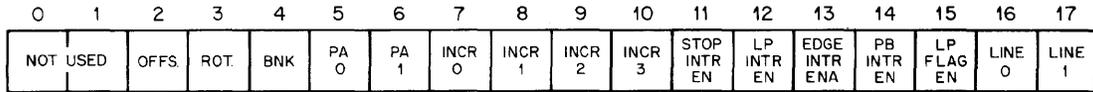
15-0601

Figure 2-4   Status 1 Word Format

Table 2-1
Read Status 1 Word Bit Functions

Bit Pos	Name	Function
2	OFFS	Indicates status of offset-parameter register. When set indicates offset has been enabled.
3	ROT	Indicates status of rotate-parameter register. When set indicates that all vectors are rotated 90° counterclockwise.
4	BNK	Indicates status of blink-parameter register. When set indicates that blink circuit is enabled.
5 and 6	PA0, PA1	Indicates effective paper size.
7 through 10	$INCR_0$ through $INCR_3$	Indicates content of increment-parameter register. When set in combination indicates relative magnitude of increment between 00_8 and 17_8.
11	STOP INTR EN	Indicates status of stop interrupt enable register. When set indicates that a stop flag will interrupt PDP-15 computer program.
12	LP INTR EN	Indicates status of light-pen flag interrupt register. When set indicates that a light-pen hit will interrupt PDP-15 computer program.
13	EDGE INTR ENA	Indicates status of edge-flag interrupt register. When set indicates that the PDP-15 computer program will be interrupt when the edge limit has been exceeded.
14	PB INTR EN	Indicates status of push-button flag interrupt register. When set indicates that a push-button hit will interrupt PDP-15 computer program.
15	LP FLAG	Indicates status of light-pen flag enable register. When set, a light pen hit will cause a flag.
16 and 17	$LINE_0$ and $LINE_1$	Indicates content of line parameter register. When set in combination indicates type of dashed lines in vectors.

2.4.3 Read Status 2 (RS2)

The Read Status 2 IOT (703022 or 703032 for AC CLR) loads the special format 18-bit word shown in Figure 2-5 into the PDP-15 accumulator. Table 2-2 lists the function of each bit of the status word.

NOTE

If the VT15 is running a display file that can modify the pushbuttons (Skip 1 instruction) care must be taken not to allow a Read Status 2 of the buttons at the same time or erroneous data will result.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
PB 0	PB 1	PB 2	PB 3	PB 4	PB 5		EDGE FLAG EN	BRT 0	BRT 1	BRT 2	NR 0	NR 1	NR 2	NR 3	NR 4	NR 5	NR 6
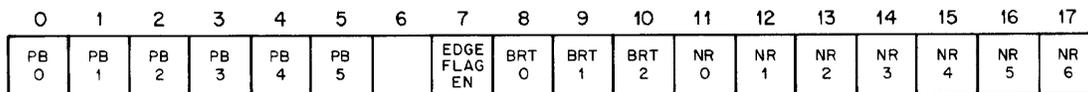
15-0602

Figure 2-5   Status 2 Word Format

Table 2-2
Read Status 2 Word Bit Functions

Bit Pos	Name	Function
0 through 5	PB_0 through PB_5	Indicates content of pushbutton skip register.
8 through 10	BRT_0 through BRT_2	Indicates content of brightness parameter register. When set in combination indicates relative brightness of display between 0_8 and 7_8; with 0_8 the darkest and 7_8 the brightest.
11 through 17	NR_0 through NR_6	Indicates content of name register.

2.4.4 Read Status 3 (RS3)

The Read Status 3 IOT (703142 or 703152 for AC CLR) loads the specially formatted 18-bit word shown in Figure 2-6 into the PDP-15 accumulator. Only the six most significant bits of this status word are used.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
STOP FLAG	PB FIND	LP FLAG	EDGE FLAG	EXT STOP	ARB VEC					NOT USED							
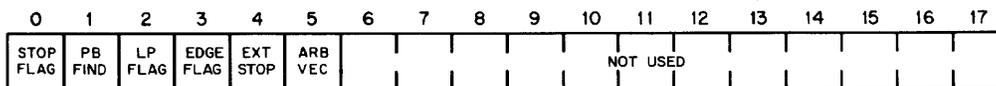
15-0603

Figure 2-6   Status 3 Word Format

2.4.5 Read X Position Register (RXP)

The Read X Position IOT (703102 or 703112 for AC CLR) loads the specially formatted 18-bit word, shown in Figure 2-7, into the PDP-15 accumulator. Because bits 0 through 4 of the 18-bit word are unused and bit 5 indicates an X-register overflow, the maximum word length is 12 bits. When 12 bits

2-6

are set, the maximum coordinate position has been named and is 4095. This register can accommodate all coordinate positions for paper sizes up to 37 in. (Refer to Paragraph 2.3.) The overflow bit is useful for indicating that the X-register capacity (full paper size) has been exceeded.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	NOT USED				X OVER FLO	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}

15-0604
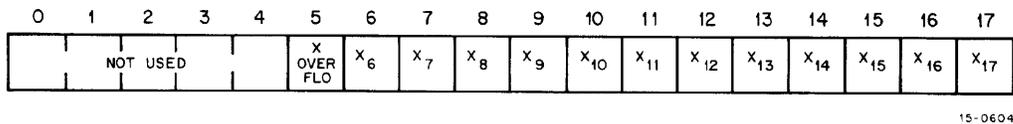
Figure 2-7   Read X Position Word Format

2.4.6   Read Y Position Register (RYP)

The Read Y Position IOT (703042 or 703052 for AC CLR) loads the specially formatted 18-bit word shown in Figure 2-8 into the PDP-15 accumulator. Bits of this word have the same significance for the Y axis as those of the Read X Position IOT described above.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
	NOT USED				Y OVER FLO	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}	Y_{17}

15-0605
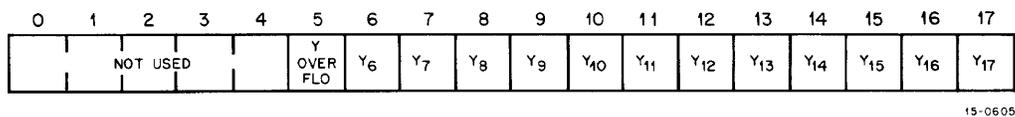
Figure 2-8   Read Y Position Word Format

2.4.7   Read Program Counter (RPC)

The Read Program Counter IOT (703062 or 703072 for AC CLR) loads the contents of the PC into the PDP-15 accumulator. Bit 0 is not used. The PC can hold a maximum of 131,072 addresses. Refer to **Figure 2-9.**

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7	PC_8	PC_9	PC_{10}	PC_{11}	PC_{12}	PC_{13}	PC_{14}	PC_{15}	PC_{16}	PC_{17}

15-0618
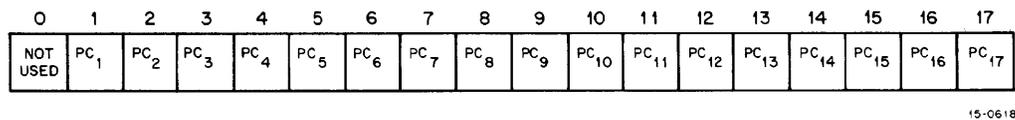
Figure 2-9   Program Counter Word Format

2.4.8   Skip IOT Instructions

Skips available to the programmer can originate either from the PDP-15 program or from the display file. Both types of skips are produced as a result of some VT15 Graphic Processor event, but they are generated in two ways. Display file skips are generated when ordered by a skip instruction from the VT15 Graphic Processor instruction set. PDP-15 program skips are generated when ordered by any of the seven skip IOTs listed in Table 2-3.

2-7

Table 2-3
Skip IOT Instruction Set

Mnemonic	Octal Code	Operation	Function
SPSF	703001	Skip on Stop flag.	Causes the computer program to skip next instruction when the stop flag is raised.
SPLP	703021	Skip on light pen flag.	Causes the computer program to skip next instruction when the light pen flag is raised.
SPPB	703041	Skip on pushbutton flag.	Causes the computer program to skip next instruction when any one of six pushbuttons on the display console is depressed.
SPEF	703061	Skip on edge flag.	Causes the computer program to skip next instruction when a vector exceeds a previously defined paper size edge.
SPDF	703101	Skip on any flag.	Causes the computer program to skip next instruction when any flag in the VT15 is raised.
SPDI	703121	Skip on any interrupting flag.	In the VT15 instruction set there is an interrupt and two skip parameter words. The parameter 3 instruction enables a program interrupt on the occurrence of any one of five flags. This IOT instruction will cause a skip when any of the five flags is raised only when the parameter 3 interrupt is also enabled. When the flag is raised and the interrupt enabled, the computer program will be interrupted and a program skip will occur.
SPES	703161	Skip on external stop flag.	One of the IOTs in this instruction set is an external stop instruction (STPD). This stop originates from the PDP-15 computer program via the STPD IOT. When the external stop IOT is received by the VT15 Graphic Processor the display will stop and the external stop flag will be raised. This IOT permits the PDP-15 computer program to skip when this flag is raised.

2.4.9 Operate IOT Instructions

There are four operate IOTs. These IOTs are used to initiate the display file, to clear flags, to enable interrupts, to establish paper size, to stop the display (VT15 Graphic Processor halts), and to resume the display file after a flag occurs or if a display stop has been previously ordered. The operate IOTs and their functions are listed in Table 2-4.

2-8

Table 2-4
Operate IOT Instruction Set

Mnemonic	Octal Code	Operation	Function
LSD	703004	Load and Start display.	This IOT is used to initialize a single cycle word transfer sequence by the PDP-15 computer and entry into the display file by the VT15. The transfer is initialized by an 18-bit address word which is first loaded into the accumulator. This address is the location of the first instruction in the display file. It is loaded into the VT15 PC when this IOT occurs.
STDP	703044	External stop display.	Provides an external stop of the display. Refer to the SPES instruction described in Table 2-3.
RES	703064	Resume display after flag.	This IOT causes the display to clear all flags and continue after a flag occurs.
SIC	703024	Set initial conditions.	This IOT is used to set initial conditions for computer interrupts, to clear flags, and to establish paper size. The initial conditions are determined by an 18-bit word formed in the PDP-15 accumulator. The word is shown in Figure 2-10. Bit functions are listed in Table 2-5.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
SET STOP FLAG INTR	SET LP FLAG INTR	SET EDGE FLAG INTR	SET PB FIND INTR	SET EXT STOP INTR	CLR STOP FLAG	CLR LP FLAG	CLR EDGE FLAG	CLR PB FIND	CLR EXT STOP	PA CHAN ENA	PA 0	PA 1	LP INTR CLR	SPB 0	SPB 1	EN LP FLAG EN	CLR LP FLAG EN
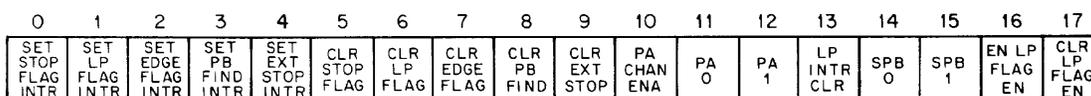
15-0598

Figure 2-10 Set Initial Conditions Word Format

Table 2-5
Significance of Set Initial Conditions Word Bits

Bit Pos	Name	Function
0	SET STOP INTR EN	When set, stop flag will produce a PDP-15 program interrupt.
1	SET LP INTR EN	When set, a light-pen flag will produce a PDP-15 program interrupt.
2	SET EDGE INTR EN	When set, an edge flag will produce a PDP-15 program interrupt.
3	SET PB INTR EN	When set, and the operator depresses any one of six pushbuttons on the VT04, a PDP-15 program interrupt will occur.

2-9

Table 2-5 (Cont)
Significance of Set Initial Conditions Word Bits

Bit Pos	Name	Function
4	SET EXT STOP INTR EN	When set, and External Stop IOT, STPD (703044) is issued from the PDP-15 computer program, a PDP-15 program interrupt will occur when the VT15 stops.
5	CLR STOP FLAG	When set, the stop flag will be cleared.
6	CLR LP FLAG EN	When set, the light-pen flag will be cleared.
7	CLR EDGE FLAG EN	When set, the edge flag will be cleared.
8	CLR PB FIND	When set, the pushbutton flag will be cleared.
9	CLR EXT STOP	When set, the external stop flag will be cleared.
10	PA CHAN EN	When set, the effective paper size will be changed according to the code given by bits 11 and 12.
11 and 12	PA00 and PA01	When set in combination, the effective size of the visible area of the CRT screen is as follows:

PA00	PA01	VT04 Size (inches)
0	0	X1 (9-1/4 x 9-1/4)
0	1	X2 (18-1/2 x 18-1/2)
1	0	X4 (37 x 37)
1	1	X4 (37 x 37)

Bit Pos	Name	Function
13	CLR LP INTR EN	When set, clear Light Pen Interrupt enable is cleared.
14 and 15	SLAVE PB00 and PB01	Set in binary combination to indicate which display console is being addressed by a subsequent RS2 IOT.

Note: Bits 0 through 9 are effective only if they are set. The cleared condition has no significance other than negation of a set bit.

2.4.10 Miscellaneous IOT Instructions

Table 2-6 lists two PDP-15 IOT instructions that affect VT15 Graphic Processor operation.

Table 2-6
Miscellaneous IOT Instructions

Mnemonic	Octal Code	Operation	Function
IORS	700314	Input/Output Read Status	Upon receiving this IOT, the VT15 will return a logic 1 to bit position 5 of the PDP-15 accumulator if any flag exists within the VT15.
CAF	703302	Clear All Flags	Upon receiving this IOT, the VT15 will clear all flags and reset the display to a "power clear" condition.

2.5 VT15 GRAPHIC PROCESSOR INSTRUCTION SET

There are eight 18-bit instructions in the basic VT15 instruction set. Each instruction has a separate format but generally consists of an instruction field, a parameter field, and an address or data field. There are three types of VT15 instructions: memory reference, graphic, and parameter. Memory reference instructions have 13-bit address and parameter fields. Graphic and parameter instructions have parameter and data fields. Data fields in operate instructions vary in length from three to 13 bits. Figure 2-11 shows the formats for each of the VT15 Graphic Processor instructions. The parameter/skip instruction is further augmented into three parameter instructions by a 2-bit register field and two skip instructions by a 1-bit subregister field.

INSTRUCTION BITS

INSTRUCTION NAME	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
CHARACTER INPUT	INSTRUCTION FIELD				NOT USED	PARA FLD	NOT USED						CHARACTER DATA FIELD					
CHARACTER STRING	INSTRUCTION FIELD				PARA FLD	ADDRESS FIELD												
POINT/ GRAPHPLOT	INSTRUCTION FIELD				PARAMETER FIELD			COORDINATE DATA FIELD										
PARAMETER/ SKIP	INSTRUCTION FIELD				REGISTER FIELD		SUB REG FIELD	PARAMETER FIELD										
SAVE/ RESTORE	INSTRUCTION FIELD				PARA FLD	13-BIT ADDRESS												
BASIC VECTOR	INSTRUCTION FIELD		PARAMETER FIELD				10-BIT DATA FIELD											
SHORT VECTOR	INSTRUCTION FIELD		PARAMETER FIELD				DATA FIELD				PARAMETER FIELD				DATA FIELD			
JUMP/ JMS	INSTRUCTION FIELD		PARAMETER FIELD				ADDRESS FIELD											

15-0620

Figure 2-11   VT15 Graphic Processor Instruction Format

The VT15 Graphic Processor instructions are used in combination to produce a display file. The display file is the VT15 Graphic Processor program. Two IOT instructions are usually used with each display file: The Set Initial Conditions IOT, SIC (703024) and the Load and Start Display IOT, LSD (703004). The Set Initial Conditions IOT is used only when its bits are of significance to display file operations. The Load and Start Display IOT must always be used when a display file is initiated.

Memory reference instructions are Character String, Save/Restore, and Jump/Jump-to-Subroutine. Each of these memory reference instructions operates on data at an absolute address specified by a 13-bit address field which is part of the instruction word.

The parameter instructions generally establish characteristics of points, graph plots, characters and vectors. These instructions also establish PDP-15 program interrupts when flags are raised in the VT15 or when pushbuttons on the display console are pressed by the operator; they permit the display file to skip on various programmable flags and on one or more of the six pushbuttons. Flags can be enabled by setting the proper bit in the parameter instruction.

With the exception of a 3-bit operate field in the Basic Vector and Basic Short Vector, and Jump instructions, each instruction is characterized by a 4-bit operate field. Parameter and skip instruction operate fields are characterized by an additional 2-bit sub-instruction field for parameters and a 3-bit sub-instruction field for skips. Sub-instruction fields are necessary to characterize augmentation of the basic parameter and skip instructions. Instruction fields tell the graphic processor what to do with each instruction. Each instruction requires a minimum of three 250-ns time states to be processed.

2.5.1 Character Input Instruction

The Character Input instruction format is shown in Figure 2-12. The Character Input instruction provides a convenient means of displaying individual text characters. When processed, the VT15 Graphic Processor operates directly on this instruction so that the character named in the ASCII field is displayed directly from the display file. The 8-bit ASCII code is listed in Appendix B.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	NOT USED	LP FLG EN		NOT USED						7 OR 8 BIT ASCII				
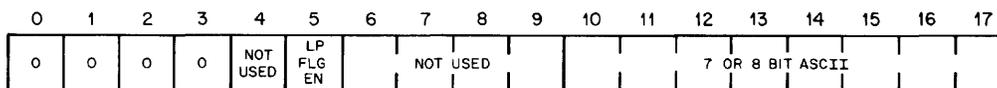
15-0608

Figure 2-12   Character Input Instruction Bit Format

This instruction is also useful when using the light pen to identify the generation of a character. When a display file is active and the light pen flag bit of this instruction is enabled, a skip or interrupt can be generated by properly encoding a previous parameter or skip instruction. This mode is also useful for displaying a limited number of characters, or in response situations, from a keyboard or TTY.

2-12

When specifying ASCII characters, any of six characteristics can be determined by the programmer; he may or may not specify them all:

a. Brightness – Brightness of any character is established by a 3-bit parameter field in the Parameter 1 instruction and is determined in a manner identical to that described for vectors in Paragraph 2.5.4.1.

b. Starting Place – Characters may be started anywhere on the specified paper size. Figure 2-13 shows the character generation format for all 64 displayed characters together with the 5 x 7 coordinate-point matrix. The numbers and arrowed lines in the matrix indicate the direction and number of vectors required to generate the illustrated character; solid lines indicate visible vectors and dashed lines indicate invisible vectors. All characters are automatically generated by the character generator when called for by the Character Input instruction. For example, to generate character A, six vectors are required, four visible and and two invisible. Character A starts at the coordinate dot at the beginning of vector 1 and ends at the arrowhead of vector 6. The dot location at the arrowhead of the final vector of the character is the starting place of the first vector of the next character. Therefore, only the starting place of a single character or the first character in a string of characters needs to be defined.
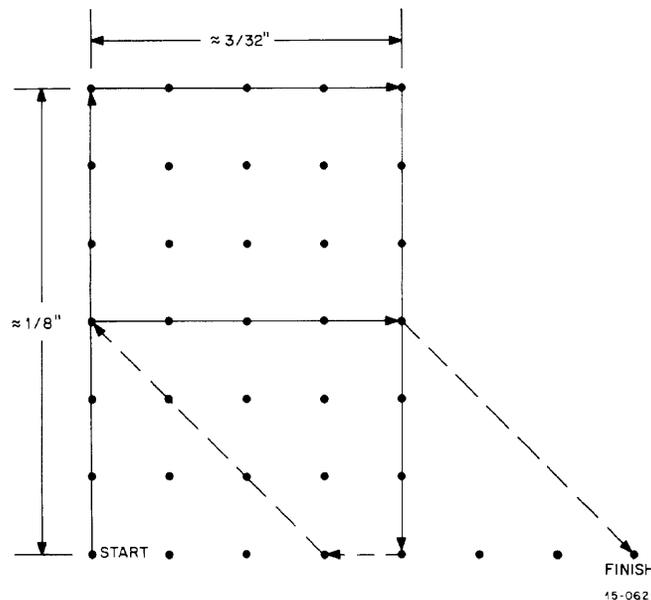
Figure 2-13   Character Generation Format

The starting place of any character can be specified using the Arbitrary Vector instruction (Paragraph 2.5.10), the Basic Vector instruction (Paragraph 2.5.7), or by using the Point/Graph Plot instruction (Paragraph 2.5.3).

c. Size – The size of any character can be continuously expanded from its basic dimensions of approximately 1/8 in. by 1/8 in., up to 15 times. The expander increments are determined by a 4-bit data field in the Parameter 1 instruction (refer to Paragraph 2.5.4). The contents of the increment register in the VR15 are processed for each Character Input instruction. When this register contains data, each vector in the character is repeated the number of times equivalent to the data word contained in the increment register.

2-13

d. Rotation - All characters can be rotated 90° counterclockwise so that characters can be written in the vertical and the horizontal planes. Rotation is selected by setting a parameter bit in the Parameter 2 instruction. (Refer to Paragraph 2.5.4.2.)

e. Light Pen Enable - If a light pen enable is desired on any character, bit 5 of the input character instruction should be set. Setting this bit will enable the light pen, but only on the character named in the data field.

f. Blink - The blink parameter permits any vector to be blinked at an approximate rate of four cycles per second (Paragraph 2.5.4.2).

NOTE: Characters may not be dashed.

2.5.2   Character String Instruction

The 18-bit Character String instruction format is shown in Figure 2-14. This instruction has a 4-bit operating code of 04_8 which occupies bits 0 through 3. Bits 5 through 17 form a 13-bit absolute address field capable of naming up to 8,192 locations. This address is the starting location in IOPS character format. (Refer to Figure 2-15.) If the address specified by the 13-bit address field does not occur in the same memory as the instruction, bit 4, which is used to specify an address in an alternate memory storage bank, must be set.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	1	IND ADDR						13 BITS OF ADDRESS							

15-0606
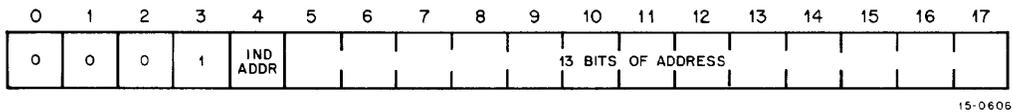
Figure 2-14   Character String Instruction Bit Format

When bit 4, the indirect addressing bit, is set, the address named in the 13-bit word does not give the starting location, but the address of the starting location. Indirect addressing permits the address word to be expanded to 17 bits so that any one of 128K core memory locations can be referenced.

The memory reference instruction directs the program counter in the VT15 to the named address which the programmer has assigned as the start of his ASCII file. The character generator un-packs the ASCII file and displays each character of the file in sequential order until a delimiter such

	FIRST 18-BIT WORD		SECOND 18-BIT WORD		
BIT POSITION	0 6	7 13	14 17 0 2 3	9 10 16	17
CHAR. POSITION	1st. CHAR.	2nd CHAR.	3rd CHAR.	4th CHAR.	5th CHAR.	0

15-0622
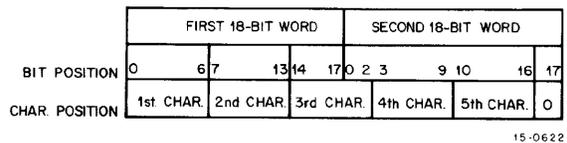
Figure 2-15   IOPS Character Format

as an ALT MODE terminates the character string operation mode. With the exception of the light pen hit, the same characteristics used to specify ASCII characters when using the Character Input instruction are used to specify characters in the Character String instruction.

A characteristic not available in the Character String instruction that can be specified in the parameter instruction is the character escape mode. Once initialized the character generator remains active

until the escape character has been written. When bit 6 of the Parameter 2 instruction is set and bit 7 of this instruction is also reset only the ALT MODE character will terminate the Character String instruction. If bit 7 is set then both ALT MODE or a CR will terminate the Character String instruction. All characters must also be specified in IOPS ASCII format (Figure 2-15).

2.5.3 Point/Graph Plot Instruction

The 18-bit Point/Graph Plot instruction format is shown in Figure 2-16. Point/graph plot instructions have an operation code of 14_8 which is defined by the operation-code field consisting of bits 0 through 3. This instruction is augmented into two instructions by bit 7 permitting the programmer to plot points in two ways. When bit 7 is reset (0), the VT15 interprets the Point/Graph Plot instruction as a point plot; when bit 7 is set (1), the instruction is interpreted as a graph plot.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
O	O	1	1	INT ENA	O	X=1 Y=0	GRP=1 PT=0				10-BITS OF COORDINATE DATA						
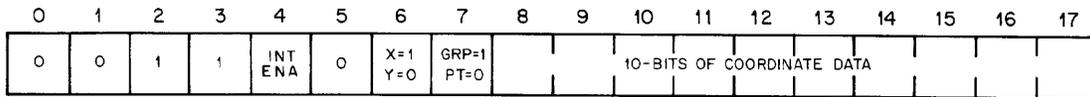
15-0599

Figure 2-16   Point/Graph Plot Instruction Bit Format

2.5.3.1   Point Plot - When a point plot is ordered by the programmer (bit 7 reset), the instruction must also name the axis in which the CRT writing beam is to move. Bit 6 of this instruction defines the axis in which the CRT beam is to be moved; when this bit is set, the beam is moved along the X axis. When unset, the beam is moved along the Y axis. Bits 8 through 17 are a 10-bit data field that permits the programmer to specify any point absolutely with respect to the origin along the named axis. Because the beam is displaced from its previous position only when the coordinate data field is used, if the beam is deflected in only one axis, only that axis need be named in bit 6.

If the beam is to be moved in both axes, then the Point/Graph Plot instruction must be used twice, once to move the beam along the X axis (bit 6 set) and once to move the beam along the Y axis (bit 6 unset) requiring from 8 to 20 μs/point instruction. Usually the point is written unintensified (invisibly) during the first beam movement.

When plotting points, there are four characteristics that can be determined by the programmer. Although he might not specify all of them, the following parameters should be considered by the programmer:

a. Brightness - Brightness of any point (or vector) is established by a 3-bit brightness field in the Parameter 1 instruction and is determined in a manner identical to that described for vectors in Paragraph 2.5.4.1.

b. Intensification - When points are moved in both axes, usually the first move is invisible and the second move visible or invisible. Beam intensification is selected in bit 4 of the Point/Graph Plot instruction. This intensification bit pertains only to the point named in the 10-bit coordinate data field. Points cannot be blinked.

c. Axis of Beam Deflection - When moving a point, only the axis of beam deflection need be named to execute movement. Bit 6 of the Point/Graph Plot instruction is used to name the axis of beam movement.

d. Absolute Displacement from Origin - When points are displayed, all beam movements are referenced to the origin - relative displacement of the beam from its previous position should not be considered when planning point plots. All coordinates between 0 and 1023 in the visible image area can be expressed in the 10-bit coordinate data field of the Point/Graph Plot instruction. All coordinates must be entered into the coordinate data field as octal numbers.

2.5.3.2   Graph Plot - When a graph plot is ordered by the programmer (bit 7 set), only one axis need be specified by the Point/Graph Plot instruction. The beam is automatically moved a predetermined amount by the hardware, according to the content of the increment register in the VT15. The increment register is loaded through the issuance of a Parameter 1 instruction. The Parameter 1 instruction has a 4-bit increment field defined by bits 14 through 17 of the parameter/skip instruction (Paragraph 2.5.4). This increment register permits the CRT beam to be moved in decimal increments between 1 and 15 rasters, or between 0.01 and 0.150 in., (approximately) in the unspecified axis.

Graph plots conserve 50 percent of core memory when a curve to be plotted for two variables is represented by a data file having all point for one variable solved for fixed increments in the other variable.

When plotting graphs, there are three characteristics that can be determined by the programmer. As with a point plot, he might not specify all of them, but the following parameters should be considered by the programmer.

a. Brightness and Intensification - When graph plotting, each coordinate point is moved in both axes, but only the axis for which data samples have been taken need be named for each point. Bit 4, the intensification bit, will permit the final graph/plot point to be visible; the point in the unnamed axis will remain unintensified. Bit 6 (Figure 2-16) of the Point/Graph Plot instruction is used to specify the axis for which data samples have been taken. Graph plots cannot be blinked.

b. Absolute Displacement of Data Points - When graphs are plotted, movement of the CRT beam in the specified axis is a function of the data samples taken. The displacement of the beam in the specified axis should be referenced absolutely with respect to the origin. For a continuous graph, (increments of 1) 1024 samples should be taken between 0 and 1023. However, when 512 samples are taken, in increments of 2, between 0 and 1023 the eye will just begin to resolve individual points. It is possible to take data increments as large as 15, which establishes a requirement for 64 data samples. Using the arbitrary (stroke) vector option, it is possible to connect points with a series of straight lines. All coordinates between 0 and 1023 in the visible image area can be expressed in a manner identical to that used to express point-plot coordinates (in octal) as described above.

c. Increment of Data Points - Data point increments should be determined on the basis of the number of sample points taken and the desired resolution of the dot pattern described by the graph plot. For graph plotting, the data point increments are determined by a 4-bit data field in the Parameter 1 instruction. The content of the increment register in the VT15 is processed for each graph plot and causes a relative displacement of the CRT beam, in the unnamed axis, with respect to its last previous position. Hence, a graph will automatically be plotted for successive fixed increments in the unnamed axis when data points in the specified axis are separated by that increment.

2.5.4 Parameter/Skip Instruction

The parameters of the Parameter/Skip instruction format are shown in Figure 2-17. This instruction has 6-bit operating codes of 20_8, 21_8, 22_8, or 23_8, which are used to indicate any one of three parameter instructions or a skip instruction. The skip instruction (Figure 2-19) is further augmented into two sub-instructions having codes of 230_8 and 234_8.

PARAMETER 1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	0	0	STOP FLAG	INT ENA	3-BITS BRIGHT			NOT USED		INCR ENA	4-BITS OF INCREMENT DATA (SCALE)			

PARAMETER 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	0	1	ESCP EN	1=CR 0=ALT MODE	BLINK ENA	ON=1 OFF=0	EDGE FLAG ENA	1=ON 0=OFF	ROTATE ENA	VERT=1 HORZ=0	LP ENA	ON=1 OFF=0	OFFSET ENA	ON=1 OFF=0

PARAMETER 3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	1	0	NOT USED				INTR ENA	STOP INTR ENA	LP INTR ENA	EDGE INTR ENA	PB INTR ENA	LINE ENA	2-BITS LINE	

15-0597

Figure 2-17 Parameter Instruction Bit Formats

The parameter instructions perform no direct operations such as drawing points or vectors, or characters, but are used to establish one or more characteristics of these drawing functions. The parameter instructions can be thought of as augmenting instructions, but they apply equally to all drawing

instructions in the display file when used. With but one exception (the stop-flag bit), each of the parameters that can be named in the three parameter instructions has an enabling bit. When a wanted parameter is specified, its enabling bit must also be specified. Each of the three parameter instructions permits more than one parameter to be specified; hence, without an enabling bit, each time one of the parameter fields is used to specify a new parameter or a change in that parameter, all other parameters in the same instruction would have to be repeated because all bits of the instruction are usually processed. The presence of the enabling bit ensures that only the bits of the enabled parameters will be processed by the VT15; thus, the need for bookkeeping on the part of the programmer is eliminated. If the enabling bit was not present with each of the parameter bits, each time a parameter instruction is used, the programmer would have to remember how he had previously specified other parameters in the same instruction. For the VT15, this would require keeping track of 14 variables at all times. The 14 parameters that can be specified by the three parameter instructions are listed in Table 2-7.

Table 2-7
Parametric Variables

Parameter 1	Parameter 2	Parameter 3
a. Stop flag b. Brightness c. Increment	a. Escape Mode b. Blink c. Edge d. Rotate e. Light Pen f. Offset	a. Stop flag interrupt b. Light pen interrupt c. Edge interrupt d. Pushbutton interrupt e. Line Type

2.5.4.1   Parameter 1 – The Parameter 1 instruction (Figure 2-17) permits the programmer to specify the three parameters listed in the Parameter 1 column of Table 2-7. Each of these parameters is described in a., b., and c. below:

a. Stop flag – When bit 6 of the Parameter 1 instruction is set (Figure 2-17) the stop flag will be raised and the VT15 will halt. The programmed stop flag can be used to produce a computer skip or it can be tested through the issuance of the Read Status 3 IOT instruction (Paragraph 2.4.2). The stop flag status bit position for the Read Status 3 word is shown in Figure 2-6. A program interrupt can be specified on the occurrence of this flag by setting bits 10 and 11 of the Parameter 3 instruction.

b. Brightness – The brightness of the programmed display is determined by bits 8, 9, and 10 of the Parameter 1 instruction, but only if brightness enable bit 7 is also set. The brightness of the display can be controlled in eight incremental steps between maximum dark and maximum light by specifying any octal code, representing the wanted brightness, between 0 and 7, respectively. The

b.   brightness parameter code is usually established at the beginning of a display file,
(cont)   and is seldom changed. However, if the brightness of a coordinate point, a
vector, or a character to be generated is to be different from all previously
generated points, vectors, or characters, a new Parameter 1 instruction must be
issued in the display file; this new Parameter 1 instruction must have enabling
bit 7 set and the new value of brightness coded into bits 8, 9, and 10, the
brightness parameter field. After the character, vector, or point whose brightness
has been altered is written, the former Parameter 1 instruction must be restored in
the display file if subsequent characters are to have the originally selected brightness.

The brightness code can be tested through the issuance of the Read Status 2 IOT
instruction (Paragraph 2.4.3). The brightness bit positions for the Read Status
2 word are shown in Figure 2-9.

c.   Increment - The 4-bit increment parameter field in the Parameter 1 instruction
is used in the following ways:

1.   It permits any vector or character to be repeated up to 15 times. This
feature is useful in zooming and scaling.

2.   It permits any graph-plot coordinate-point increment between 1 and 15
to be specified for beam deflection in the unnamed axis when using the
graph-plot feature of the Point/Graph Plot instruction.

When a vector, character, or point plot instruction is processed by the VT15, the parameter registers
are examined as part of the instruction execution. When vectors are written either as lines or charac-
ters, the increment register is examined for scale. When a graph-plot point is written in the unspeci-
fied axis, the increment register is examined for beam displacement.

For vectors and characters, the scale code is usually established at the beginning of the display file.
This is also usually true for graph-plot coordinates and the increment code. These parameters are sel-
dom changed. However, if the vector repeatability or plot increment to be generated is to be differ-
ent from all previously generated repeats or increments, a new Parameter 1 instruction must be issued
in the display file; this new Parameter 1 instruction must have enabling bit 13 set and the new value
of the scale or increment coded into bits 14 through 17, the increment parameter field.

After the new scale or increment has been written, the former Parameter 1 instruction must be restored
in the display file if subsequent vectors, characters, or graph plots are to have the originally selected
number of repeats or basic increments.

The increment code can be tested through the issuance of the Read Status 1 IOT instruction (Paragraph
2.4.2). The increment bit positions for the Read Status 1 word are shown in Figure 2-4.

2.5.4.2   Parameter 2 - The Parameter 2 instruction (Figure 2-17) permits the programmer to specify
the six parameters listed in the Parameter 2 column of Table 2-7. Each of these parameters is de-
scribed in a. through f. which follow:

a. Escape Mode – The escape mode is used when a Character String instruction is used to specify the starting address of a character string. When escape mode enable bit 6 is set in the Parameter 2 instruction escape code, bit 7 will be read when the instruction is processed. If bit 7 is set then both an ALT MODE or CR will cause termination of the Character String instruction. If bit 7 is reset, only an ALT MODE code will terminate the character string.

b. Blink – When the programmer desires to blink a vector, a character, or character string, he uses the blink parameter bits of the Parameter 2 instruction. When blink enable bit 8 is set in the Parameter 2 instruction and bit 9 is set for ON condition, the blink bit will be loaded into the blink register. Subsequent characters or vectors will then blink until the blink register is reset through the Parameter 2 instruction again. The status of the blink bit can be tested through the issuance of the Read Status 1 IOT instruction. The blink status bit for the Read Status 1 word is shown in Figure 2-4.

The blink bit is usually used just for single characters and vectors, and is often changed. When the vector or character is written, it will blink at an approximate rate of four times a second.

c. Edge – When the programmer desires to know when the paper size he has chosen is exceeded, he uses the edge bits of the Parameter 2 instruction. When edge flag enable bit 10 is set in the Parameter 2 instruction, the edge control bit and edge will be loaded into the edge flag enable register. If edge bit 11 is set, when a vector exceeds the chosen effective paper size, an edge flag will be raised by the VT15 Graphic Processor. Refer to Paragraph 2.3 for a description of paper size selection. The edge flag can be used to produce a computer skip through the issuance of the Skip IOT instruction SPEF (Paragraph 2.4.8).

The status of the edge flag bit can be tested through the issuance of the Read Status 3 IOT instruction. The edge flag status bit for the Read Status 3 word is shown in Figure 2-6.

Note that if a line violates the edge of a viewing area, it will be completely blanked. For example, if either or both end points of a line are outside of the viewing area, the entire line will disappear from view. However, the display file will be updated logically. If the portion of the picture is drawn entirely outside of the viewing area, the vectors are not drawn. Instead, the processor simply accomplishes its additions and continues, providing a great saving in drawing time.

Whenever a vector's starting point is outside the viewing area, but its end point is within the viewing area, an additional 20-μs delay is imposed to allow the CRT monitor to settle before the next element is drawn. Refer to Figure 2-18.

d. Rotate – When the programmer wishes to rotate his displayed vectors or characters 90° counterclockwise, he uses the rotate bits of the Parameter 2 instruction. When rotate enable bit 12 is set in the Parameter 2 instruction, the rotate control and rotate bit will be loaded into the rotate register. If rotate bit 13 is set, the vectors or characters subsequently generated will be rotated 90° counterclockwise. The status of the rotate bit can be tested through the issuance of the Read Status 1 IOT instruction. The rotate status bit for the Read Status 1 word is shown in Figure 2-4.

The rotate bit is most often used when generating characters to annotate a part of a display. If a character or character string is to be rotated, a new Parameter 2 instruction must be issued in the display file. This Parameter 2 instruction must have enabling bit 13 and rotate bit 14 set. When subsequent vector or characters are written they will be rotated 90° counterclockwise with respect to their starting point.
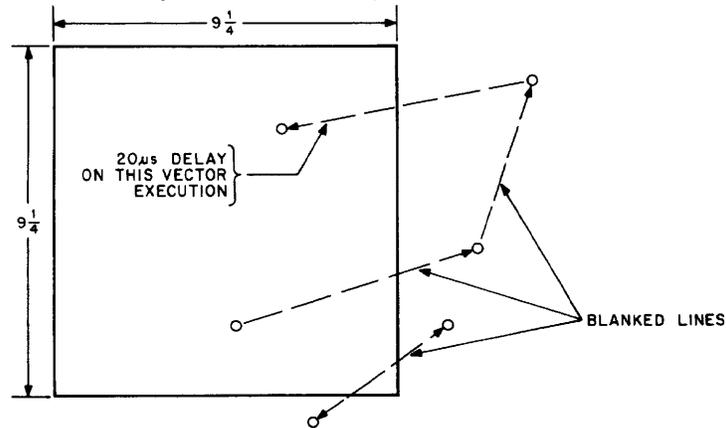
Figure 2-18   Screen Edge Violation Examples

e. Light Pen – When the programmer wishes to use the light pen to detect any intensified part of the display, he uses the light pen bits of the Parameter 2 instruction.

When light pen enable bit 14 is set in the Parameter 2 instruction, the light pen bit will be read when each character, vector, or point/graph plot instruction is executed. If light pen bit 15 is set, whenever a light pen hit occurs, a light pen flag will be raised by the VT15. However, because of phosphor rise time limitations, light pen throughput time, and the time delay of the data and control bus, the light pen flag may not always set during the instruction that actually caused the light pen to respond. For example, during the point word, the phosphor is excited for only 250 ns. Within another 500 ns, the next instruction is brought into the data buffer. However, the rise time of the phosphor (i.e., before light can be sensed) may be as long as 3 μs (1.5 μs is typical). Therefore, because of the fast drawing rate and efficient double-buffering of the I/O, the next instruction could appear before the light pen flag is sensed.

The light pen flag can be used to produce a computer skip. To produce a program skip, the Skip IOT instruction SPLP (Paragraph 2.4.8) must be issued.

The occurrence of the light pen flag can also be tested through the issuance of the Read Status 3 IOT instruction. The light pen flag status bit for the Read Status 3 word is shown in Figure 2-6. A program interrupt can also be specified at the occurrence of this flag by setting bits 10 and 12 of the Parameter 3 instruction.

f. Offset – The offset bits of the Parameter 2 instruction are enabled when the use of the minor image area is desired. When offset enable bit 16 is set in the Parameter 2 instruction, the offset control bit 17 will be loaded into the offset register. A point X (PX) and point Y (PY) instruction must follow each entry and exit from the offset area.

2-21

2.5.4.3   Parameter 3 - With the exception of the line selection bits (dash), the Parameter 3 instruction (Figure 2-17) is used to enable program interrupts for various skip conditions. There are four flags whose occurrence can enable an interrupt if their interrupt bits are set in the Parameter 3 word:

    a.   Stop flag (bit 11)

    b.   Light pen hit flag (bit 12)

    c.   Edge flag (bit 13)

    d.   Pushbutton find (bit 14)

The programmer can enable the program interrupt facilities when any event, which causes any (or all) of the above flags to be raised, is sensed. The stop, light pen, and edge flags cause a display file halt. Further, the programmer can use any one of the six pushbuttons on the display console to perform some manually initiated display function. There are actually nine sources of program interrupts in the Graphic-15 System. There are two types of interrupts in the PDP-15 computer system: program interrupts (PI) and automatic priority interrupts (API).

The program interrupt feature of the computer is enabled by programming an ION instruction. When the interrupt enabling flag is raised, the PDP-15 computer program traps to location 0 at the conclusion of the instruction currently being executed. When this occurs the program count is stored at location 0 and the next instruction, location 1, names a programmer specified subroutine that will identify the source of the interrupt. Peripheral devices (Teletype for example) must be identified in this manner. Once the interrupt producing device is identified, then its particular interrupt-producing flag must be identified. In the case of the VT15, this can easily be done by testing each interrupt source using the Read Status 3 IOT instruction and a masking word. When the KA15 Automatic Priority Interrupt option is used in the PDP-15, the first level (peripheral isolation) of interrupt identification is automatically performed by the hardware. For each of the peripheral equipments having a data channel assignment, the PDP-15 program requires a service subroutine that is entered through a hardware-defined trap address; for the Graphic-15 System, this trap address is 54 at priority level 2. When used, this address will contain a memory reference instruction that directs operation of the computer to the VT15 Graphic Processor service subroutine in the program.

The mechanism for controlling the interrupt enabling signals on the occurrence of raised flags is a series of flip-flops in the hardware, which when set, produce a computer program interrupt when the flag is raised. These flags are: the stop, light pen, edge, and any one of six pushbutton flip-flops. However, before the status of any one of the interrupt registers can be changed, bit 10 of the Parameter 3 instruction must be set, or the Set Initial Conditions (SIC) IOT must be issued with the proper bits set in the SIC word.

Also located in the Parameter 3 instruction are line selection bits 16 and 17 and line enabling bit 15. Whenever a line selection is made by the programmer, this bit must be set. The line selection codes permit the programmer to identify any vector as a dashed line. The two line selection bits permit three alternate dashed-line configurations. Table 2-8 lists the possibilities in terms of line increments lighted and unlighted, or on to off. Any dashed line may be blinked just like any vector. Characters may not be dashed.

Table 2-8
Line Pattern Codes

Line Pattern	Line Code		Display Ratio (On to Off)
	Bit 16	Bit 17	
───────	0	0	Always on
───── ──	0	1	6 lines on; 2 lines off
─ ── ── ──	1	0	3 lines on; 1 line off
─ ─ ─ ─ ─ ─	1	1	1 line on; 1 line off

The skip producing part of the parameter/skip instruction format is shown in Figure 2-19. The skip instruction has an operating code of 23_8 and is augmented into two subinstructions having operating subcodes of 230_8 and 234_8. Skips provide the programmer with one of the most powerful decision making tools available in the VT15 Graphic Processor instruction set. These skips, however, are display file skips, not computer skips. Computer skips are programmed through the VT15 IOT instruction set; the computer skip IOTs are described in Paragraph 2.4.8. The two VT15 Graphic Processor skip instructions do not deal with skips exclusively.

SKIP 1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	1	1	0	LP SENSE IND	CLR AFTER TEST	SKIP ON PB=1 SET PB=0	PB 0	PB 1	PB 2	PB 3	PB 4	PB 5	UNIT	SELECT

SKIP 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	1	1	1	HLT & RESUME IN SYNC	SKP UNCON	LOAD NAME	SKIP ON NAME			7 BITS OF NAME				
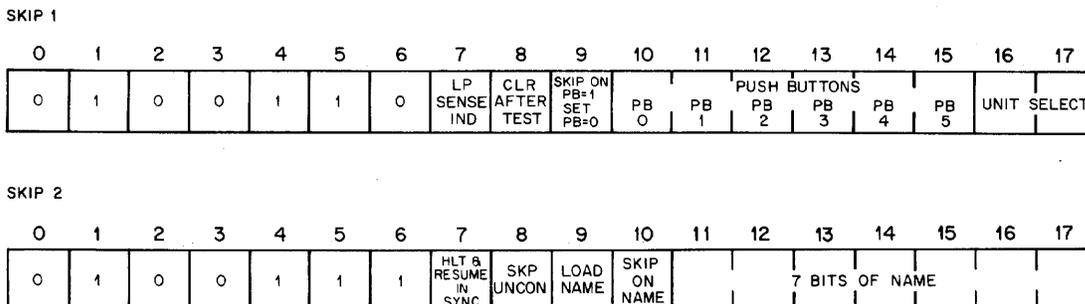
15-0600

Figure 2-19   Skip Instruction Bit Formats

2-23

Table 2-9
Skip Instruction Variables

Skip 1	Skip 2
a. Skip on light pen sense indicator b. Clear pushbutton after test c. Skip on pushbutton d. Pushbutton enable e. Select pushbutton bank	a. Sync to line b. Skip unconditionally c. Load name register d. Skip on name word e. Name word enable

2.5.4.4   Skip 1 – The Skip 1 instruction permits the programmer to specify the variables listed in the Skip 1 column of Table 2-9. The skips, with programmer-controlled pushbutton functions, provide the programmer with both software and hardware decision-making powers. The skips can be used in the display file to skip on light pen sense indicator, to branch on pushbutton register functions, and to identify flag producing pushbutton functions devised by the programmer. A wide variety of programmed alternatives is possible through the use of the Skip 1 pushbutton bits, the Skip 2 name register bits, and the Read Status 2 IOT instruction.

a.   Skip on Light Pen Sense Indicator – This instruction is used to skip whenever a light pen hit occurs but the light pen flag, which is enabled and set by bits 14 and 15 in the Parameter 3 instruction, is not enabled. When bit 7 of the Skip 1 instruction is set, any light pen hit will cause a display file skip without computer intervention. The display processor does not stop since the flag was not set.

b.   Clear any Pushbutton PB_0 through PB_5 – When bit 8 of the Skip 1 instruction is set to a 1, the pushbutton flip-flop is cleared and the lamp turned off.

c.   Skip on Pushbutton – When bit 9 of Skip 1 instruction is set, where any of the pushbuttons PB_0 through PB_5 on the display console are set by the programmer, and if the corresponding pushbutton bit has been set in the Skip 1 instruction word, a display file skip will occur. For example, if the programmer wishes to cause a skip when he sets pushbutton PB_4 he sets bits 9 and 14 in the Skip 1 instruction; this gives the octal code 230140. Thus, when the operator presses pushbutton PB_4 on the display console and the PB_4 flip-flop register in the VT15 sets, a display file skip will occur when the above Skip 1 instruction is executed.

To clear any pushbutton flip-flop register, bit 8 in the Skip 1 instruction must be set. If the PB_4 flip-flop register is to be cleared immediately after the skip, bit 8 of the Skip 1 instruction should also be set. This gives the equivalent octal code, 231410, which results in the following sequence: when the operator presses pushbutton PB_4 and the PB_4 pushbutton flip-flop register sets, the pushbutton indicator lights, and a display file skip occurs when the above Skip 1 instruction is executed. Subsequently, the pushbutton flip-flop register is cleared and the PB4 indicator goes out. The sequence occurs so rapidly that no visible indication takes place.

c. If bit 9 of the Skip 1 instruction is not set and a pushbutton bit in the Skip 1
(cont) instruction is set, the corresponding pushbutton register will be set and the
pushbutton will light. The status of the pushbutton registers can be tested through
the Read Status 2 IOT instruction as previously described in Paragraph 2.4.3.
Computer skips on combinations of pushbuttons can be interactively programmed
by using mask words with the Read Status 2 IOT instruction, but the name
registers provide the programmer with easier programming possibilities.

d. Pushbuttons PB_0 through PB_5 – Bits 10 through 15 in the Skip 1 instruction refer
to the pushbutton registers PB_0 through PB_5 and are used with bits 8 and 9 as
previously indicated.

e. Select Pushbutton Bank – Used with the VM15 Display Console Multiplexer where
more than one display console is interfaced with the VT15 Graphic Processor.
Bits 16 and 17 of the instruction are set or reset in combination to give four
possible combinations for selection of a pushbutton bank or LPSI flip-flop. The
address codes are as listed in Table 2-10.

Table 2-10
Pushbutton Bank Address Codes

Instruction Bit		Pushbutton Bank
16	17	
0	0	Used for only one bank
0	1	Optional Bank 1
1	0	Optional Bank 2
1	1	Optional Bank 3

2.5.4.5 Skip 2 – The Skip 2 instruction permits the programmer to specify the variables listed in the
Skip 2 column of Table 2-9.

The name register is a 7-bit register in the VT15 hardware. This register can hold 128 combinations of
binary codes. These codes can be used to tag subroutines for the purpose of future identification. The
name register is loaded through bits 11 through 17 of the Skip 2 instruction and load name bit 9.

For example, repeatable subroutines can be assigned 7-bit binary tags by the programmer which he can
encode as he calls them. Then, if later the graphic presentation drawn by the called subroutine is to
be deleted, he can identify the subroutine using the previously assigned name register tags.

a. Sync to Line Frequency – A synchronizing register in the VT15 permits the programmer to synchronize operation of the VT15 with the incoming line frequency. Synchronizing operation of the VT15 with the line frequency eliminates a visible swimming effect on the CRT monitor when such interference occurs. When bit 17 of the Skip 1 instruction is set, the display file will stop. When the incoming sample line frequency signal passes a preset threshold, a trigger pulse is produced by the hardware. When the trigger pulse is sensed, the display file is started again, but is now synchronized with the incoming line frequency.

b. Skip Unconditionally – This instruction is used to skip unconditionally in the display file. Whenever bit 8 in the Skip 2 instruction is set, an unconditional skip will occur in the display file.

c. Load Name Register – This instruction is used to load the hardware name register with any binary code between 0 000 000 and 1 111 111, which gives the programmer 128 encoding possibilities. When bit 9 of the Skip 2 instruction is set, 7-bit name word given in bits 11 through 17 of the Skip 2 instruction will be loaded into the hardware name register.

d. Skip on Name Word – This instruction is used to skip test for the word currently stored in the name register. The content of the name register can be examined through a skip test by setting bit 10 and setting the wanted name-register code into bits 11 through 17 of the Skip 2 instruction. If the content of the name register and the current skip code in the name register bits are identical, a display file skip will occur.

e. Name-Word Bits NW_0 through NW_6 – Bits 11 through 17 of the Skip 2 instruction are used in two ways: to load the name register or to skip test the content of the name register depending on whether bits 9 or 10 of the Skip 2 instructions are set.

2.5.5 Save/Restore Instruction

The 18-bit Save/Restore instruction format is shown in Figure 2-20. This instruction has a 4-bit operating code of 24_8 which occupies bits 0 through 3. Bit 4 of this instruction indicates whether the parameters listed below should be saved from or restored into the display file. When bit 4 of this instruction is set, the parameters should be restored into the display file; when this bit is not set (a zero) the parameters should be saved. Bits 5 through 17 of this Save/Restore instruction form a 13-bit address field capable of naming up to 8,192 locations in the PDP-15 core memory. The 13-bit address is used to name the location where the parameters listed below are to be saved, or from which they are to be restored into the display file:

a. Offset

b. Rotate

c. Blink

d. Edge flag enable

e. Brightness

f. Stop interrupt enable

g. Light pen interrupt enable

h. Edge interrupt enable

i. Pushbutton interrupt enable

j. Light pen flag enable

k. Line bits

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	1	SAVE =0 RESTORE =1					13-BITS OF ADDRESS								
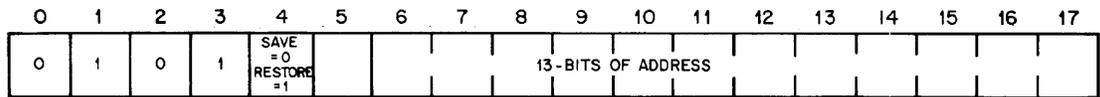
15-0616

Figure 2-20   Save/Restore Instruction Bit Format

When the Save/Restore instruction is used, the parameters listed in a. through k. above are collected from their respective registers into a formatted status word. The status word format is shown in Figure 2-21. The content of the addressed save or restore location can be easily examined by simply issuing a LAC instruction which names the status word location. This instruction is useful when it is desired to change one or two parameters in the display file for a small part of the display file. Parameters that are normally unchanged are saved into some location in the PDP-15 core memory and then restored after the display file functions have been programmed. This instruction is also useful when jumping to subroutines. The status of previously programmed parameters may be saved at some memory location and then restored when leaving the subroutine which permits full nesting of any level of subroutines.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
INCR 0	INCR 1	INCR 2	INCR 3	BLNK	ROT	OFFSET	EDGE FLAG ENA	BRT 0	BRT 1	BRT 2	STOP INTR EN	LP INTR EN	EDGE INTR EN	PB INTR EN	LP FLAG EN	LINE 0	LINE 1
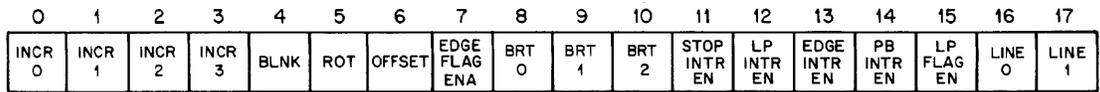
15-0617

Figure 2-21   Save/Restore Instruction Status Format

2.5.6   Basic Vector Instruction

The 18-bit Basic Vector instruction format is shown in Figure 2-22. This instruction has a 3-bit operating code of 4_8 which occupies bits 0, 1, and 2. This instruction provides the parameter and data fields necessary to generate any vector up to full screen in length in any one of eight directions. This instruction also has a light pen and intensity bit. When enabled, light pen bit 3 will produce a light pen enable only on the vector defined by this instruction. Intensity bit 4 permits the programmer to draw the vector either visibly (bit set) or invisibly (bit unset).

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	LP ENA	INT	3-BITS DIRECTION					10-BITS OF Δ VECTOR							
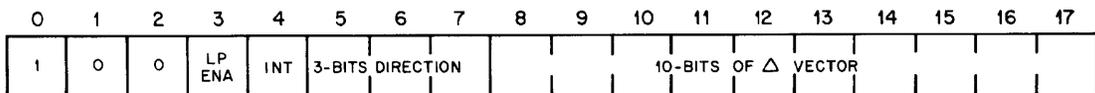
15-0595

Figure 2-22   Basic Vector Instruction Format

The VT15 draws lines on the CRT through an analog function generator in any one of eight possible directions. In addition to the specification of the vector directions, the vector line length, up to full screen, must also be specified. The linear vector is generated from a relative position either intensified (visible) or unintensified (invisible). The basic vector instruction is used to specify three characteristics

of the wanted line: intensification (visible or invisible), direction, and displacement (called delta).
Initially, one or more basic parameter instructions that specify other characteristics of the overall display or functions of the graphic processor are used to determine other characteristics of vectors such as brightness, 90° rotation, and scale.

Figure 2-23 shows eight vectors superimposed on a basic cartesian coordinate system. The direction for each vector is shown by the arrowhead on each line. Bits 5 through 7 establish vector direction. It can be seen from the figure that vectors lying along the axes of the coordinate system can be expressed by their direction 0, 2, 4, or 6 and the displacement of the line (delta) along the axis upon which they are located. The coordinates of these lines need not be expressed because they are already defined by delta and the line directions. For example, a vector that takes direction 4 will move to the left on (or parallel with) the X-axis.

Examining vectors 1, 3, 5, and 7, it can be seen that their displacement along both axes has the same delta; vector 3 is displaced 3 delta units when projected to the negative X-axis and 3 delta units when projected to the positive Y-axis. Again, these coordinates do not need to be expressed because they are defined by delta and the direction that vector 3 takes. To express any basic vector only direction and delta (relative displacement along an axis) need be expressed to generate that vector. Note that the length of any odd-numbered direction vector will be 1.416 times delta.

Figure 2-23   Basic Vector Directions

Bits 8 through 17 of the Basic Vector instruction permit the programmer to specify any vector length (delta word) between 0.009 to 9-1/4 in. Each binary increment, from 0 to 1024 increases the vector length approximately .009 in. (VT04).

2.5.7   Basic Short Vector Instruction

The 18-bit Basic Short Vector instruction format is shown in Figure 2-24. This instruction has a 3-bit operating code of 5_8 which occupies bits 0, 1, and 2. The Basic Short Vector instruction provides the parameter and data fields necessary to generate two short vectors up to approximately 0.06 in. (prior to scaling) in length in any of the eight basic directions. This instruction permits the programmer to pack two short vectors into one instruction; thus conserving core space. The use of this instruction

2-28

rather than the Basic Vector instruction makes more efficient use of core memory and I/O bus timing. Using this instruction together with the scale code bits of the Parameter 1 instruction permits vectors up to approximately 1-1/4 in. to be drawn.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	LP ENA	INT	3-BITS DIRECTION				Δ		INT	3-BITS DIRECTION			Δ		

15-0613
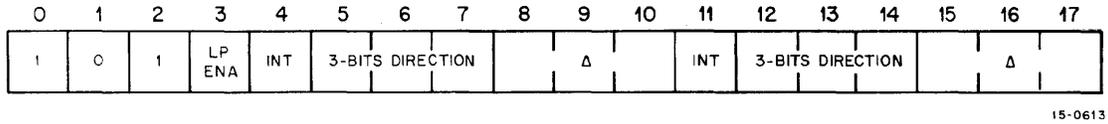
Figure 2-24 Basic Short Vector Instruction Bit Format

Bit 3 of the Basic Short Vector instruction is also the light pen bit for the vectors defined by both direction and data fields of this instruction. Light pen limitations described in Paragraph 2.5.4 apply here as well. Bits 4 and 11 are the intensity bits for the first and second short vectors, respectively. The two intensity bits are provided so that they may be drawn visibly (bits set) or invisibly (bits unset) independently of each other.

Bits 5, 6, and 7 establish direction for the vector whose delta word is defined by bits 9, 10, and 11. Bits 12, 13, and 14 establish direction for the vector whose delta word is defined by bits 15, 16, and 17.

When programming basic short vectors, the same considerations given basic vectors should also be given basic short vectors.

2.5.8 Jump/Jump-to-Subroutine Instruction

The 18-bit Jump/Jump-to-Subroutine instruction format is shown in Figure 2-25. This instruction has a 3-bit operating code of 6_8 which occupies bits 0, 1, and 2. This instruction can be used either as a Jump or a Jump-to-Subroutine instruction, depending on the state of bit 3. If bit 3 is set, the instruction is a Jump-to-Subroutine; if bit 3 is reset the instruction is a Jump.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	0	JMS	INDR				13-BITS OF ADDRESS									

15-1614
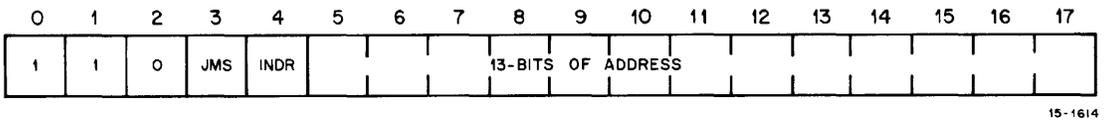
Figure 2-25 Jump/Jump-to-Subroutine Instruction Bit Format

The Jump, or DJMP instruction permits the program to repeat instructions already performed or to jump over display file instructions in the normal programmed sequence. This instruction is used within one memory bank. The hardware loads the effective address of the location named in the instruction address field into the VT15 program counter, thereby changing the normal programming sequence.

2-29

When the Jump-to-Subroutine (DJMS) instruction is ordered by the programmer, the DJMS instruction stores a pointer address (bits 5 through 17) in the first location of a subroutine and transfers program control to the second location of the subroutine. After the subroutine has been executed, the pointer address identifies the next instruction to be executed. This scheme provides the programmer with a method of exiting the basic display file to perform some graphic display function, and a means of returning to the correct location in the basic display file at the completion of the subroutine.

The indirect address bit (bit 4) provides a means of addressing to any one of 128K core locations. The 13-bit address (bits 5 through 17) of this instruction does not give the pointer word location, but the location of the pointer word address. Indirect addressing permits the pointer word to be expanded to 17 bits.

2.5.9 Slave Instruction

The Slave instruction allows the programmer to selectively blank or intensify up to four multiplexed display consoles when the VM15 Display Console Multiplexer option is implemented. The instruction also controls the light pen enable function on each display. The Slave instruction bit format is shown in Figure 2-26.
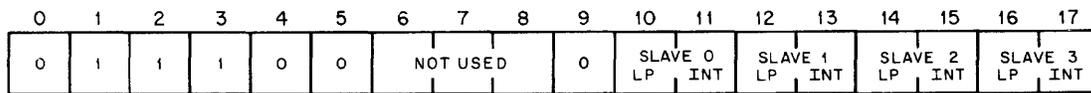
Figure 2-26   Slave Instruction Bit Format

The instruction is used by setting the proper bits to a logical 1 to enable either the intensity, or light pen, or both, on any combination of up to four display consoles.

2.5.10 Arbitrary Long Vector Instruction

When the VV15 Arbitrary Vector Generator option is installed in the VT15 Graphic Processor, vectors other than the eight basic vectors shown in Figure 2-23 can be drawn without resorting to time consuming software algorithms that use the Basic Vector instruction. For example, a software algorithm to draw a 22.5° vector requires 512 instructions. When the VV15 option is implemented, this same vector requires only two instructions. Bit formats for the 2-word Arbitrary Long Vector instruction are shown in Figure 2-27.

The operation code for both words is 10_8. In the first word, bit 4 is set to intensify the vector. The light pen is enabled when bit 5 is set. Bit 6 specifies the direction of X. When bit 6 is a logic 1,

the direction is negative. When bit 6 is reset, the direction of X is positive. Bit 7 is not used. Bits 8 through 17 define the ΔX component length of the vector.

WORD 1-ΔX

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	0	INT	LP EN	DIR 1 = − 0 = +	0				10 BITS OF ΔX VECTOR						

WORD 2-ΔY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	0	0	0	DIR 1 = − 0 = +	0				10 BITS OF ΔY VECTOR						

15-0611
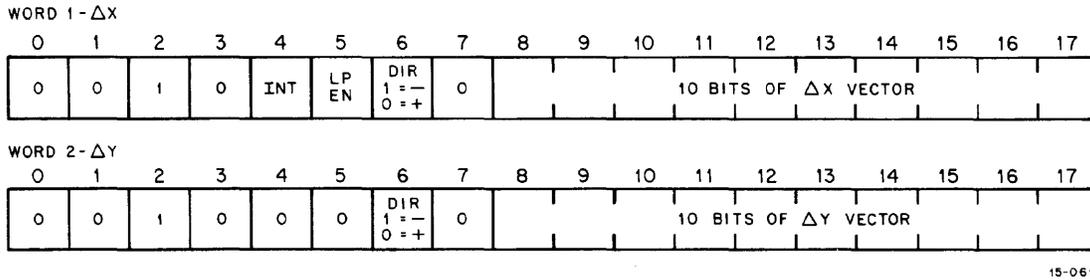
Figure 2-27   Arbitrary Long Vector Instruction Bit Format

In the ΔY word, bit 6 specifies the direction of the ΔY component and bits 8 through 17 specify the length of the ΔY component. Both words must be used in the proper sequence to execute the vector. The condition that ΔX and ΔY both equal zero is illegal. If this condition should occur, the VT15 will cycle indefinitely in a "normalization" loop.

2.5.11   Arbitrary Short Vector Instruction

The Arbitrary Short Vector instruction can be used instead of Arbitrary Long Vector to conserve core space and execution time when only short vectors are to be generated. As indicated in the bit format, Figure 2-28, both the ΔX and ΔY components of the vector are contained within a single instruction word.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	0	INT	LP EN	DIR 1 = − 0 = +		5 BITS OF ΔX VECTOR				DIR 1 = − 0 = +	5 BITS OF ΔY VECTOR				

15-0612
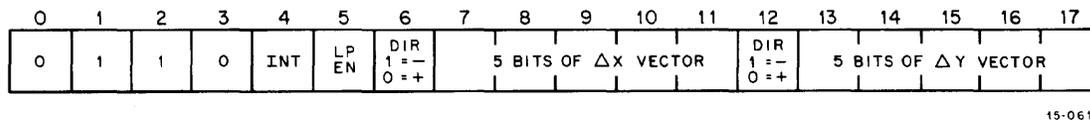
Figure 2-28   Arbitrary Short Vector Instruction Bit Format

Because of the reduced ΔX and ΔY fields, the length of a vector drawn by the Arbitrary Short Vector instruction is limited to 31 raster units. However, by using the scaling register, the length can be increased.

2-31

CHAPTER 3

PROGRAMMING EXAMPLES

3.1 INTRODUCTION

This chapter describes some very simple subroutines that a novice can use to familiarize himself with the VT15 IOT and graphic processor instructions. These subroutines will acquaint the user with sufficient basic concepts so that he can examine further capabilities and possibilities on his own.

The subroutines are brief so that they can be manually deposited into core, and altered, from the PDP-15 operator's console. Programs are written at the machine language level.

NOTE

The VT15 Graphic Software System is not described in this chapter. Refer to the Graphic-15 Programming Manual, DEC-15-ZFSA-D, for system software information and programming.

3.2 MACRO-15 MNEMONICS

Table 3-1 is a partial list of mnemonics assigned to the Graphic-15 instruction set. The MACRO-15 Assembler recognizes these symbolic mnemonics to assemble the program in machine language. The equivalent machine language instruction is listed in octal base. The mnemonics are included in the programming examples to familiarize the reader with their usage.

3.3 MICROCODING INSTRUCTIONS

Microcoding allows the programmer to combine more than one instruction in a single display file location. An exclamation mark (!) is used between mnemonics of the instructions to be microcoded. The MACRO-15 Assembler interprets the exclamation mark to indicate microcoding and performs an inclusive OR of the machine language instructions in the accumulator to provide the combined microcoded machine language instruction.

Table 3-1
Instruction Mnemonics

Mnemonic	Octal	Definition
IOT Read Instructions [†]		
RS1	703012	Read Status 1
RS2	703032	Read Status 2
RS3	703152	Read Status 3
RPC	703072	Read program counter
RYP	703052	Read Y position register
RXP	703112	Read X position register
IOT Skip Instructions		
SPSF	703001	Skip on stop flag
SPLP	703021	Skip on light pen flag
SPPB	703041	Skip on pushbutton flag
SPEF	703061	Skip on edge flag
SPDF	703101	Skip on any flag
SPDI	703121	Skip on any interrupting flag
SPES	703161	Skip on external stop flag
IOT Operate Instructions		
LSD	703004	Load and start display
SIC	703024	Set initial conditions
STDP	703044	External stop display
RES	703064	Resume display after flag
Character Generation		
CHARI	000000	Character input
CHARS	040000	Character string
Point/Graph Plot Instructions		
GY	142000	Graphplot mode, Y direction
GX	146000	Graphplot mode, X direction
PY	140000	Point mode, Y direction
PX	144000	Point mode, X direction

[†] The Clear AC bit is included in the IOTs above.

Table 3-1 (Cont)
Instruction Mnemonics

Mnemonic	Octal	Definition
Parameter 1 Instructions		
DNOP	200000	Display no operation
INTO	202000	Intensity 0
INT1	202200	Intensity 1
INT2	202400	Intensity 2
INT3	202600	Intensity 3
INT4	203000	Intensity 4
INT5	203200	Intensity 5
INT6	203400	Intensity 6
INT7	203600	Intensity 7
SCALE	200020	Set scale
STOP	204000	Stop display
Parameter 2 Instructions		
LPON	210014	Enable light pen
LPOF	210010	Disable light pen
OSETN	210003	Offset on
OSETF	210002	Offset off
ROTON	210060	Rotate on
ROTOF	210040	Rotate off
BKON	211400	Blink on
BKOF	211000	Blink off
EDGON	210300	Edge flag enable
EDGOF	210200	Edge flag disable
Parameter 3 Instructions		
ENSTP	220300	Enable stop interrupt
ENLP	220240	Enable light pen interrupt
ENEDG	220220	Enable edge flag interrupt
ENPB	220210	Enable pushbutton interrupt
SOLID	220004	Enable solid lines
DASH 1	220005	Dashed line, length 1
DASH 2	220006	Dashed line, length 2
DASH 3	220007	Dashed line, length 3
DISABL	220200	Disable all interrupts
Skip 1 Instructions		
SKPB	230400	Skip if indicated pushbuttons are set
STPB	230000	Set indicated pushbuttons
CLPB	231000	Clear indicated pushbuttons

Table 3-1 (Cont)
Instruction Mnemonics

Mnemonic	Octal	Definition
	Skip 1 Instructions (Cont)	
B0	230200	Pushbutton 0
B1	230100	Pushbutton 1
B2	230040	Pushbutton 2
B3	230020	Pushbutton 3
B4	230010	Pushbutton 4
B5	230004	Pushbutton 5
SLPSI	232000	Skip on light pen sense indicator
	Skip 2 Instructions	
SKIP	235000	Unconditional skip
LDNM	234400	Load name
SPNM	234200	Skip on name
SYNC	236000	Resume in sync
	Save/Restore Instructions	
SAVE	240000	Save mode
RSTR	260000	Restore mode
	Basic Vector Instructions	
V0	400000	Vector direction 0
V1	402000	Vector direction 1
V2	404000	Vector direction 2
V3	406000	Vector direction 3
V4	410000	Vector direction 4
V5	412000	Vector direction 5
V6	414000	Vector direction 6
V7	416000	Vector direction 7
INT	020000	Intensify point or vector
INCR	500000	Basic increment mode
DJMP	600000	Display jump
DJMS	640000	Display jump-to-subroutine
LPEN	010000	Light pen enable-arbitrary vector and char input modes
VLPEN	040000	Light pen enable-vector and increment modes

For example, the programmer can use a single statement to set the parameters for a specific operation, such as intensity and scale. The individual mnemonics INT4 and SCALE are coded as:

INT4 ! SCALE

When the program is assembled, the machine language code for each instruction will be ORed in the accumulator as follows:

Mnemonic	Octal	Machine Language
INT4	203000	010 000 011 000 000 000
SCALE	200020	010 000 000 000 010 000
INT4 ! SCALE	203020	010 000 011 000 010 000

NOTE

Microcoding can only be performed on parameter and skip instructions within the same group, having the same basic op code. In the above example, both instructions were selected from the Parameter 1 group listed in Table 3-1. If the instructions are not in the same group, the inclusive OR operation may result in an undesired op code.

3.4 BASIC SQUARE

This subroutine can be manually toggled into memory using the PDP-15 operator console. When the program is started at location 100, a square as shown in Figure 3-1 will be displayed on the CRT.

Address (Octal)	Contents (Octal)	Mnemonic	Comments
50	001000		/Vector square routine
100	200050	LAC50	/Load VT15 starting address
101	703004	LSD	/Load and start display IOT
102	740040	HLT	/Program halt
.			
.			/VT15 display file starts at 1000
1000	202220	INT1!SCALE 0	/Parameter 1: Intenisty 1,
			/Scale 0 increments
1001	211252	LPOF!OSETF!ROTOF!BKOF!EDGOF	/Parameter 2:
			/All items turned off.
1002	220004	DISABL!SOLID	/Parameter 3: all interrupts
			/disabled, solid line selected
1003	144000	PX!0	/Point plot X = 0 units
1004	140000	PY!0	/Point plot Y = 0 units
1005	420040	V0!INT!40	/Vector in 0 direction, 40 units
1006	424040	V2!INT!40	/Vector in 2 direction, 40 units
1007	430040	V4!INT!40	/Vector in 4 direction, 40 units
1010	434040	V6!INT!40	/Vector in 6 direction, 40 units
1011	601000	DJMP!1000	/Display Jump to square box routine
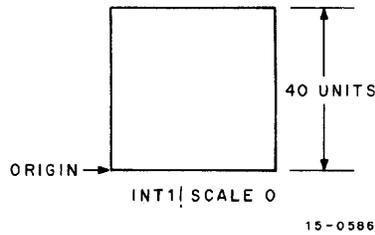
Figure 3-1   Basic Square

The VT15 Graphic Processor will continue to cycle through the display file. The PDP-15 halts at address 102. To experiment with a few instructions, stop the program and modify the display file example as indicated in the following paragraphs.

3.4.1   Relocation

Modify two locations in the display file to change the position of the square on the display:

Address	Contents	Mnemonic	Comments
1003	144600	PX!600	Point plot X = 600 units
1004	140600	PY!600	Point plot Y = 600 units

The results are shown in Figure 3-2.
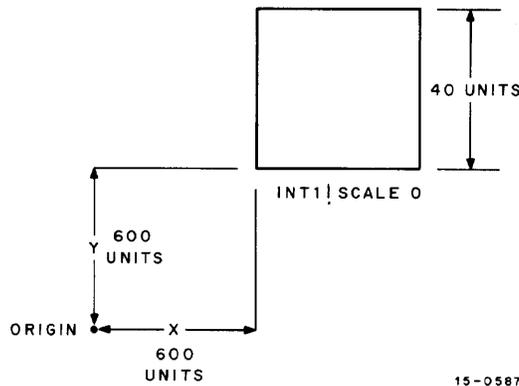
Figure 3-2   Relocation

3.4.2   Increase Intensity and Size

Modify the Parameter 1 instruction in the display file to increase the intensity and size of the square:

Address	Contents	Mnemonic	Comments
1000	203230	INT5!SCALE10	/Parameter 1: /Intensity 5, /Scale 10 units

3-6

The results are shown in Figure 3-3.

ORIGIN ●◄——— X ———►
600
UNITS

INT5 ¦ SCALE 10

400 UNITS

600
Y UNITS

15-0588
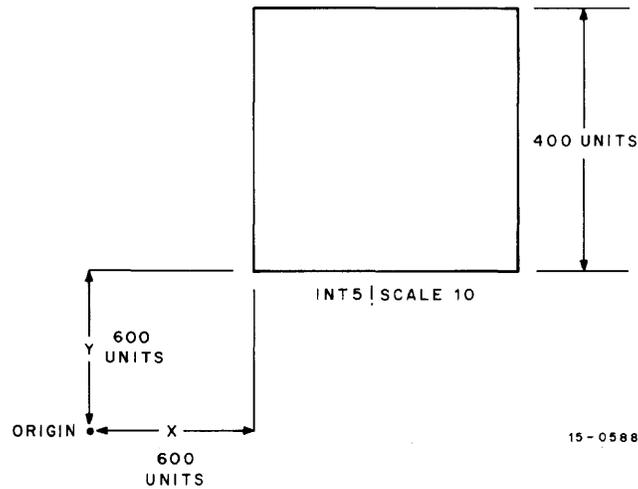
Figure 3-3   Increase Intensity and Size (Parameter 1)

3.4.3   Blink

Modify the Parameter 2 instruction in the display file to show one of the enable bits in operation.
The bit associated with blink has been selected.

Address	Contents	Mnemonic	Comments
1001	211652	LPOF!OSETF!ROTOF!BKON!EDGEOF	

As a result of this modification, the square shown in Figure 3-3 will blink on and off approximately four
times per second.

3.4.4   Parameter 3 Modification

Modify the Parameter 3 instruction in the display file to produce a dashed-line display.

Address	Contents	Mnemonic	Comments
1002	220007	DASH3	/Selects one line on and one line off /as listed in Table 2-7.

The result of this modification is shown in Figure 3-4.

3-7
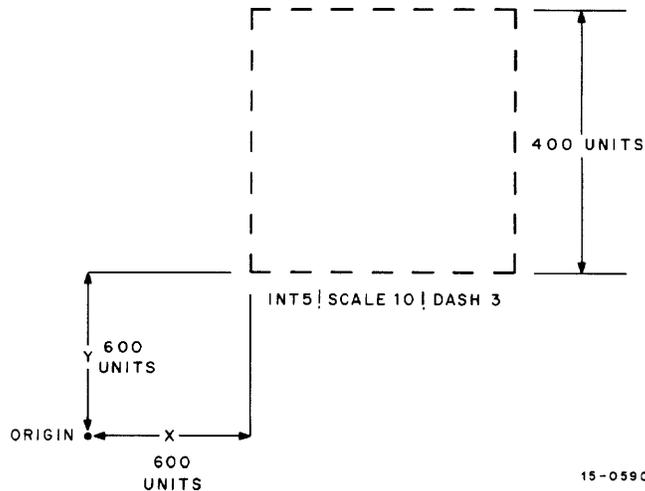
Figure 3-4   DASH3 Modification (Parameter 3)

3.5   CHARACTER INPUT

This subroutine can be manually-deposited into memory to demonstrate the use of the Character Input instruction.

Address	Contents	Mnemonic	Comments
60	002000		/Character input routine
200	200060	LAC 60	/Load VT15 starting address
201	703004	LSD	/Load and start display
202	740040	HLT	/IOT
			/Program halt
2000	203020	INT4!SCALE0	/Parameter 1: Intensity 4,
			/Scale 0 increments
2001	211252	LPOF!OSETF!ROTOF!BKOF!EDGOF	/Parameter 2: All items
			/turned off.
2002	220004	DISABL!SOLID	/Parameter 3: All interrupts
			/disabled, solid line selected
2003	145000	PX!1000	/Point plot X = 1000 units
2004	141000	PY!1000	/Point plot Y = 1000 units
2005	000304[†]	CHARI!D	/Character D input
2006	000305[†]	CHARI!E	/Character E input
2007	000303[†]	CHARI!C	/Character C input
2010	602000	DJMP 2000	/Display jump to
			/Character input
			/routine
[†]7-bit ASCII codes may also be used.			

The Graphic-15 continues to cycle through the character input display file, displaying the characters DEC on the CRT as shown in Figure 3-5.
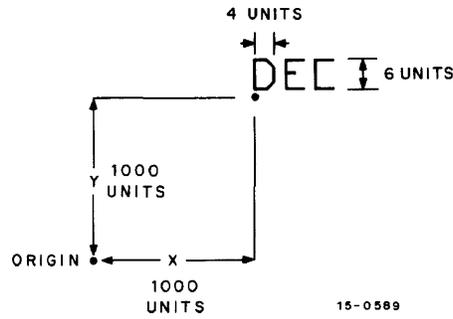
Figure 3-5   Character Input

3.6   DISPLAY FILE CHAINS

The various routines in the display file chain can be chained together to provide combinations as required.   For example, the basic square shown in Figure 3-4 can be displayed with the characters shown in Figure 3-5 by modifying the routines as follows:

Address	Contents	Mnemonic	Comments
1011	602000	DJMP!2000	/Go to character input /routine
2010	601000	DJMP!1000	/Go to basic square /routine

Start at location $100_{(8)}$ or $200_{(8)}$.   The VT15 will cycle through both routines in the display file to provide both displays.

APPENDIX A
WORD FORMATS

A.1 IOT STATUS WORD FORMATS

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED		OFFS	ROT	BNK	PA 0	PA 1	INCR 0	INCR 1	INCR 2	INCR 3	STOP INTR EN	LP INTR EN	EDGE INTR ENA	PB INTR EN	LP FLAG EN	LINE 0	LINE 1

15-0601

Read Status 1 Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
PB 0	PB 1	PB 2	PB 3	PB 4	PB 5		EDGE FLAG EN	BRT 0	BRT 1	BRT 2	NR 0	NR 1	NR 2	NR 3	NR 4	NR 5	NR 6

15-0602

Read Status 2 Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
STOP FLAG	PB FIND	LP FLAG	EDGE FLAG	EXT STOP	ARB VEC						NOT USED						

15-0603

Read Status 3 Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED					X OVER FLO	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	X_{16}	X_{17}

15-0604

Read X Position Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED					Y OVER FLO	Y_6	Y_7	Y_8	Y_9	Y_{10}	Y_{11}	Y_{12}	Y_{13}	Y_{14}	Y_{15}	Y_{16}	Y_{17}

15-0605

Read Y Position Word Format

Read Program Counter Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
NOT USED	PC_1	PC_2	PC_3	PC_4	PC_5	PC_6	PC_7	PC_8	PC_9	PC_{10}	PC_{11}	PC_{12}	PC_{13}	PC_{14}	PC_{15}	PC_{16}	PC_{17}

15-0618

Read Program Counter Word Format

Set Initial Conditions Word Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
SET STOP FLAG INTR	SET LP FLAG INTR	SET EDGE FLAG INTR	SET PB FIND INTR	SET EXT STOP INTR	CLR STOP FLAG	CLR LP FLAG	CLR EDGE FLAG	CLR PB FIND	CLR EXT STOP	PA CHAN ENA	PA 0	PA 1	LP INTR CLR	SPB 0	SPB 1	EN LP FLAG EN	CLR LP FLAG EN

15-0598

Set Initial Conditions Word Format

A.2 VT15 INSTRUCTION SET

Character Input Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	0	NOT USED	LP FLG EN		NOT USED						7 OR 8 BIT ASCII				

15-0608

Character Input Instruction Bit Format

Character String Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	0	1	IND ADDR					13 BITS OF ADDRESS								

15-0606

Character String Instruction Bit Format

Point/Graph Plot Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	1	INT ENA	0	X=1 Y=0	GRP=1 PT=0			10-BITS OF COORDINATE DATA							

15-0599

Point/Graph Plot Instruction Bit Format

PARAMETER 1

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	0	0	STOP FLAG	INT ENA	3-BITS BRIGHT			NOT USED		INCR ENA	4-BITS OF INCREMENT DATA (SCALE)			

PARAMETER 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	0	1	ESCP EN	1=CR 0=ALT MODE	BLINK ENA	ON=1 OFF=0	EDGE FLAG ENA	1=ON 0=OFF	ROTATE ENA	VERT=1 HORZ=0	LP ENA	ON=1 OFF=0	OFFSET ENA	ON=1 OFF=0

PARAMETER 3

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	1	0	NOT USED				INTR ENA	STOP INTR ENA	LP INTR ENA	EDGE INTR ENA	PB INTR ENA	LINE ENA	2-BITS LINE	

15-0597

Parameter Instruction Bit Formats

A-2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
												PUSH BUTTONS					
0	1	0	0	1	1	0	LP SENSE IND	CLR AFTER TEST	SKIP ON PB=1 SET PB=0	PB 0	PB 1	PB 2	PB 3	PB 4	PB 5	UNIT SELECT	

SKIP 2

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	0	1	1	1	HLT & RESUME IN SYNC	SKP UNCON	LOAD NAME	SKIP ON NAME		7 BITS OF NAME					

15-0600

Skip Instruction Bit Formats

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	0	1	SAVE =0 RESTORE =1					13-BITS OF ADDRESS								

15-0616

Save/Restore Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
INCR 0	INCR 1	INCR 2	INCR 3	BLNK	ROT	OFFSET	EDGE FLAG ENA	BRT 0	BRT 1	BRT 2	STOP INTR EN	LP INTR EN	EDGE INTR EN	PB INTR EN	LP FLAG EN	LINE 0	LINE 1

15-0617

Save/Restore Instruction Status Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	0	LP ENA	INT	3-BITS DIRECTION				10-BITS OF Δ VECTOR								

15-0595

DIRECTION BITS

15-0596

Basic Vector Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0	1	LP ENA	INT	3-BITS DIRECTION				Δ			INT	3-BITS DIRECTION			Δ	
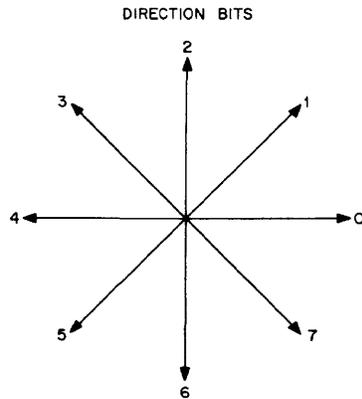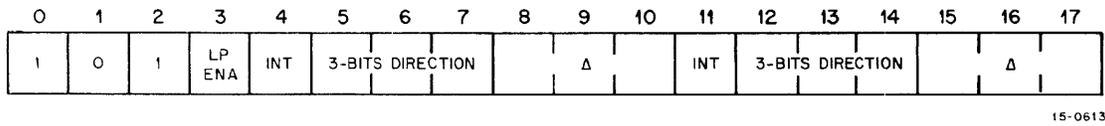
15-0613

Basic Short Vector Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1	0	JMS	INDR				13-BITS OF ADDRESS									
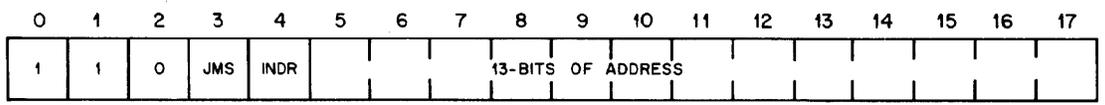
15-1614

Jump/Jump-to-Subroutine Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	0	INT	LP EN	DIR 1 = − 0 = +	0					10 BITS OF ΔX VECTOR					

WORD 2 - ΔY

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	0	1	0	0	0	DIR 1 = − 0 = +	0					10 BITS OF ΔY VECTOR					
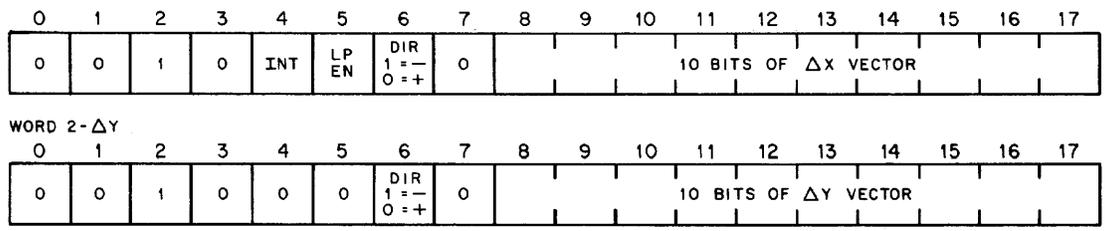
15-0611

Arbitrary Long Vector Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	0	INT	LP EN	DIR 1 = − 0 = +	5 BITS OF ΔX VECTOR					DIR 1 = − 0 = +	5 BITS OF ΔY VECTOR				
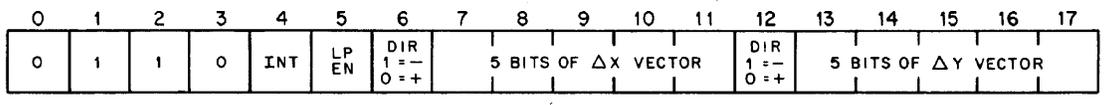
15-0612

Arbitrary Short Vector Instruction Bit Format

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1	1	1	0	0	NOT USED			0	SLAVE 0 LP | INT		SLAVE 1 LP | INT		SLAVE 2 LP | INT		SLAVE 3 LP | INT	
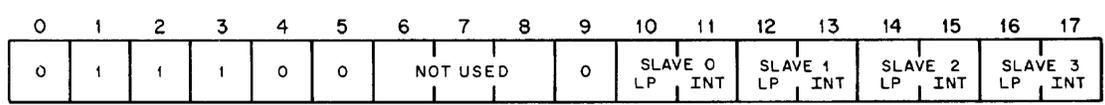
15-0610

Multiplexer (Slave) Instruction Bit Format

A-4

APPENDIX B

EIGHT-BIT ASCII OCTAL CHARACTER CODES

Character	8-bit ASCII	Character	8-bit ASCII
@	300	(Space)	240
A	301	!	241
B	302	"	242
C	303	#	243
D	304	$	244
E	305	%	245
F	306	&	246
G	307	'	247
H	310	(	250
I	311	)	251
J	312	*	252
K	313	+	253
L	314	,	254
M	315	-	255
N	316	.	256
O	317	/	257
P	320	0	260
Q	321	1	261
R	322	2	262
S	323	3	263
T	324	4	264
U	325	5	265
V	326	6	266
W	327	7	267
X	330	8	270
Y	331	9	271
Z	332	:	272
[	333	;	273
\	334	<	274
]	335	=	275
∧	336	>	276
–	337	"	277
Carriage Return	215	TAB	211
ALT MODE	375	Line Feed	212