DEC - 08 - G61B - D

PROGRAMMED BUFFERED DISPLAY 338 PROGRAMMING MANUAL

PDP-8

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

DEC - 08 - G61B - D

PROGRAMMED BUFFERED DISPLAY 338
PDP-8 PROGRAMMING MANUAL

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

CONTENTS

CONTENTS (continued)

PART 1
SYSTEM INTRODUCTION

The Type 338 Programmed Buffered Display is a precision incremental display system, consisting of a small scale, high speed computer and a display subsystem for control of the CRT. The computer used is the Digital Equipment PDP-8 (for Programmed Data Processor). It is a single address, fixed word length (twleve bits) machine. The complete cycle time for its random access magnetic core memory is 1.5 μsec. All arithmetic operations are performed in 2s complement notation.

This guide enables the experienced user to adapt the powerful capabilities of the Type 338 to his application. The contents assume that the reader is familiar with the operation of the PDP-8 and its instruction set. This information is readily available in the PDP-8 User Handbook, F-85.

1.1 FUNCTIONAL DESCRIPTION

A self-contained unit with built-in control and power provisions, the 338 Display may be interfaced to an existing system or it may stand alone as a powerful computer-driven display system. The CRT is a 16-inch tube with 9-3/8 inch by 9-3/8 inch usable display area. Magnetic deflection and focusing techniques result in uniform resolution over this area. Up to eight display CRTs may be remotely slaved to the 338 Display. All can receive identical information, or all can receive different information, or any combination can be established. The routing of information to the slave display is a function of display file instructions in the control state.

The display logic can be thought of as a special purpose computer which stores its instructions (display commands) in the memory of the PDP-8, and interacts with the computer through a series of instruction interrupts and data transfers. The display is an output device with respect to the computer for the following reasons:

a. The PDP-8 has a series of instructions which start, stop, and load and interrogate the registers of the display.

b. The PDP-8 can modify the data commands which are interpreted by the display because the commands are stored in the PDP-8 memory.

The commands are transferred to the display control via the PDP-8 single cycle data break system. The display file words are loaded into a table or block of successive memory locations; and the beginning location of this table is loaded into a special register called the display address counter (DAC). The output of the DAC is applied to the inputs of the memory address (MA) register. A data break is then initiated by either the display or the computer, and this address is read into the MA. The computer then goes through a break cycle in which it fetches the word from memory and places it into its memory buffer (MB) register from where it is transferred to the buffer register (DX) in the display. During this time, the display starts its operation and the DAC is incremented by one. The computer program counter (PC) is not incremented during the break cycle. At the end of the break cycle the PDP-8 continues its main program until the display requires another data break.

1.1.1 Display Parameters (Coordinate System)

The display screen which is 9-3/8 inches square, has 10 bits of resolution; in other words there are 1,024 points in the x and y directions or about a million points in all. The x and y position registers are 13 bits long, however, and therefore the screen represents only 1/64 the total addressable area (paper). The paper is broken up into 64 sectors corresponding to the upper 3 bits of x and y, with sector 0 defined as the lower left sector. Only information in sector 0 is intensified so that translation is accomplished by moving the paper in relation to sector 0. The lower left corner is point (0,0), and the coordinates increase to the right and up, and decrease to the left and down. An edge violation occurs when a line is drawn across the boundary of the paper. This is a warning that an overflow condition has just occurred in the x or y position register. A vertical edge flag indicates the y position register went from all 1s to all 0s, or from all 0s to all 1s. The horizontal edge flag indicates overflow in the x register. The overflow can be set to occur after the 10th, 11th, 12th or 13th bit in x and y. The virtual paper size can therefore be changed under program control.

1.1.2 Scale

The scale setting determines the number of positions each succeeding spot is moved before it is intensified. It effects both the size and appearance of lines or symbols drawn in the vector, vector continue, short vector, increment, or character modes. At scale setting 11_2, each point can be clearly distinguished. At scale setting 00_2, lines and symbols appear to be continuous. The point spacing is illustrated in the following table.

Scale	Point Spacing	Intensify
00_2	● ● ● ● ● ● ● ● ● ●	Every
01_2	● ○ ● ○ ● ○ ● ○ ● ○	2nd
10_2	● ○ ○ ○ ● ○ ○ ○ ● ○	4th
11_2	● ○ ○ ○ ○ ○ ○ ○ ● ○	8th

1.1.3 Intensity

There are eight intensity levels available on the display, ranging from 000_2, which is barely visible, to 111_2, which is very bright. Note that scale and intensity settings are interrelated. For example, if characters are drawn (with the character generator) at the lowest scale setting, and too high an intensity is used, they will be badly blurred. On the other hand, if many characters are to be displayed simultaneously or if the light pen is to be used, it is best to use as high an intensity level as possible.

1.1.4 State

The display logic is broken into two states, data state and control state. Control state commands are interpreted as instructions to the display logic to change parameters, jump, skip, etc. The data state commands are instructions to move the beam via the x and y position registers. When the display is initialized, the commands are accepted in control state until an "enter data state" command is given. The display returns to control state from data state by escaping.

In control state, the first three bits (op code) designate the operation to be performed by the remaining nine bits. Seven of the eight op codes are used:

- 0 Parameter
- 1 Mode
- 2 Jump
- 3 Pop
- 4 Conditional skip 1
- 5 Conditional skip 2
- 6 Miscellaneous (microprogrammed)
  - 0 Arithmetic compare 1
  - 1 Arithmetic compare 2
  - 2 Skip on flags
  - 3 Count
  - 4-7 Set slaves (optional)
- 7 Spare

The details are discussed in part 2 of this manual.

1.1.5 Mode

Data state words are accepted in one of seven formats according to the contents of the mode register. The data state modes available are:

0	Point
1	Increment
2	Vector
3	Vector continue
4	Short vector
5	Character (optional)
6	Graphplot
7	Spare

All modes are entered from control state by the "enter data state" command. Each mode, however, has its own way of escaping back to control state. The mode register is cleared by power clear and initialization of the display (IOT 165).

1.1.6 Subroutining

The display has control state commands which will modify the DAC. This enables unconditional display jumps (jump), jump to subroutine (push jump) and the return from subroutines (pop). The new address is specified by 15 bits allowing direct addressing of 32K of core. The jump and push jump commands are specified by two consecutive 12-bit words. Push jump stores the return address, mode, intensity, scale, and light pen on a push-down pointer list which resides in the first 4K of PDP-8 core. This information is automatically written into two locations in the format shown below:

Break Field			Light Pen	Scale		Mode			Intensity		
0	1	2	3	4	5	6	7	8	9	10	11

First Word

Low Order 12-Bits of Memory Address											
0	1	2	3	4	5	6	7	8	9	10	11

Second Word

The information is placed in the address indicated by the push-down pointer, (PDP) which is a 12-bit register in the display logic. When a push jump is executed, the PDP is incremented twice, adding a new entry to the PDP list. This allows multi-level and recursive subroutines in the display.

The pop command takes the last entry on the PDP list from core and gates it back to the proper registers. The display status, however, can be inhibited from being restored. The PDP is automatically decremented by two, making the PDP list a last in first out stack.

1.1.7 Light Pen

The light pen is an input device which generates a signal (flag) that can be sensed and interpreted by the computer. A light pen interruption stops the display, leaving the contents of all display registers intact, and signals the computer that an interruption has occurred. When this happens, the programmer can examine the contents of the display registers to determine the location (on the display) of the point of light that was sensed by the light pen and/or determine the memory location of the data word specifying that point. The light pen detects light in the range 4300 to 5600 angstroms.

1.1.8 Pushbuttons

The 338 is equipped with a bank of twelve pushbuttons. They are placed six in a row with a clear button to reset that group. The buttons in each group are interlocked, but two buttons in different groups can be pressed simultaneously. Pressing a button complements an associated flip-flop. For reference, the pushbutton is lit when its flip-flop is in the 1 state. The state of the pushbuttons can be sensed both by the display, using the control state skip instructions, and the PDP-8, which can read the state of the pushbuttons into the accumulator. The PDP-8 and display can also clear and set the pushbuttons. This enables three-way communication between the operator, display logic, and PDP-8. The buttons are labeled 0 through 11 and are packaged in a compact, portable box. The box is connected to the display by a 20-ft cable. There is also a special computer interrupt button on the box.

1.1.9 Flags

There are a number of special conditions that can arise in the display which require the attention of the PDP-8 processor. These conditions are indicated by display flags which can interrupt the computer and be sensed by IOT skip instructions. The flags are:

a. Internal stop
b. External stop
c. Edge
d. Light pen find
e. Push button hit
f. Manual interrupt

The flags can be separated into two groups; a–d are flags which stop the display; e–f are flags which do not stop the display. Group 1 flags are cleared in one of three ways: initialization of a display sequence (6165); resuming from the point the display stopped (6174_8 or 6164_8); and a pulse to clear the flag if the display is no longer needed (6161_8). Power clear clears all display flags.

1.1.9.1 Internal Stop - Internal stop is a control state "mode" command (1400_8). When the display stops, the DAC has already been incremented and points to the location after the stop code. The PDP-8 skips if the internal stop flag is on and IOT 171 is given. IOT 164 with the AC = 0 causes the display to resume from this point.

1.1.9.2 External Stop - The PDP-8 IOT 154, with AC bits 0 and 4 at 0, causes the display to stop. When the display stops, the external stop flag is set and interrupts the computer if the interrupt system is on. The time lapse between the IOT and the display stop is dependent upon the display command being executed when IOT 154 is given. The rule is that the entire command will be executed before the display is stopped. If the command is a two-word instruction like jump, pjmp, or vector, or point mode command, both words are pulled out of memory and executed before the display stops. This allows simple resuming of the sequence even if the display is used inthe interim. The external stop flag causes the PDP-8 to skip when IOT 151 is given (with AC bits 0 and 4 at 0). The display sequence can be resumed by giving IOT 174.

1.1.9.3 Edge - The edge is defined as the point at which the x or y position registers overflow. The initial conditions of the display can be set up so that all edges are ignored. In the normal case, the edge flag stops the display; when the edges are ignored, however, the display waits 35 μsec and then resumes automatically. The IOT 174 will restart the display from the edge if the display has stopped. IOT 152 causes the PDP-8 to skip if either the horizontal or vertical edge flag is on.

1.1.9.4 Light Pen Find - The light pen find flag always stops the display as soon as it senses light. This occurs about 6 μsec after the initial flash. The display logic therefore has time to execute several more points in the vector before it stops. This must be taken into consideration when reading back the x and y coordinates after the display has stopped. The PDP-8 can skip on the light pen flag using IOT 132. The vector or increment sequence can be resumed by giving IOT 174.

1.1.9.5 Pushbutton Hit Flag - If any of the twelve pushbuttons are hit, the pushbutton hit flag is raised. This flag will not stop the display, but can cause a computer interrupt. The flag is cleared by IOT 062, which reads back the status of the flag into the AC. IOT 171 reads the state of the twelve pushbuttons into the accumulator. There is no skip IOT for the pushbutton hit flag.

1.1.9.6 Manual Interrupt - The manual interrupt flag is set by the interrupt button on the pushbutton box. It has a light associated with the flag in the button. Whenever the flag is set, the button is lit. This flag will not stop the display. The flag can be cleared by IOT 172, which will also cause the program counter to skip if the flag is set.

Four of the display flags must be gated onto the interrupt line by the initial condition IOT 145. These flags are edge, light pen find, pushbutton, and internal stop. The external stop and manual interrupt flags always cause an interrupt if the interrupt system is on.

1.1.10 Timing

The display can take a break cycle a maximum of one out of three machine cycles. The effective cycle time of the display is 4.5 μsec if single cycle instructions are being executed.* All control state instructions except jump, push jump, and pop, are executed in one display cycle. The jump instruction takes two cycles because it is a two-word instruction. The push jump is a four-cycle instruction, two to get the instruction out, and two to push the two status words away on the push down list. The pop instruction is three cycles, one to obtain the instruction and two to pull the status words from the push down list.

In data state, the mode and the number of intensified points determine execution time. In the incremental modes, increment, vector, short vector, and vector continue, points are plotted at a rate of one every 1.1 μsec for intensified lines and 300 nsecs for nonintensified lines. Point and Graphplot mode words are given a 35-μsec delay to allow the beam to settle, whether the point is intensified or not. If points are plotted in the same general area, as in a continuous curve, the delay for settling is only 6 μsec. The small delay is given if, and only if, the two points plotted have the same high order 6 bits (of 13) in both x and y position. The time needed to fetch the information must then be added (one or two display cycles) to get the total execution time.

The VC38 Character Generator (optional) plots at about 37 μsec per character. This time is variable depending on the number of intensified points in the character. Control characters are executed in two display cycles except for carriage return which takes an additional 35 μsec.

To estimate the time to display a 6-bit character, add up the time required for drawing out the character, then add in 2-1/2 display cycles needed for the character dispatch.

*See PDP-8 Users Manual for details on the data break system.

PART 2
DATA FORMATS

Listed below are the data formats for the control and data state commands. The bit structure, a short general description, and a bit-by-bit explanation are given. The control state also has a list of common mnemonics.

Control state is broken into the seven op codes with the miscellaneous command broken down further to five microprogrammed commands. In data state, each of the seven modes is discussed separately.

2.1 CONTROL STATE COMMANDS

Control state commands are instructions to the display logic to change a register, such as scale, DAC, mode, etc. Some of the commands change more than one register, so an enable bit is provided for each function. If the enable bit is a 1, the associated register is reset to the appropriate value. If the enable bit is a 0, the associated register is unchanged independent of contents of the bit(s) following the enable bit. There is no restriction on the number of registers that can be changed with one instruction.

2.1.1 Parameter

Op Code: Parameter			Scale			Light Pen		Intensity			
0 0	1 0	2 0	3	4	5	6	7	8	9	10	11

Parameter command is an instruction to change the scale, light pen, and intensity registers. When the light pen reigster is a 1, a light pen hit stops the display. If the light pen register is a 0, all light pen hits are ignored.

Bit(s)	Interpretation
0-2	Op code: parameter
3	Enables scale change.
4,5	Determines one of four possible scale settings if bit 3 is a 1.
6	Enables light pen change.
7	Turns the light pen on when it is a 1, or off when it is 0, provided bit is a 1.
8	Enables intensity change.
9, 10,11	Determines one of eight possible intensity settings if bit 8 is a 1.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Code	Operation
LPOF	0040	Light pen off.
LPON	0060	Light pen on.
SC1	0400	Set scale to X1.
SC2	0500	Set scale to X2.
SC4	0600	Set scale to X4
SC8	0700	Set scale to X8.
INT*	0010	Set the intensity.

*INT only enables the change of intensity; a space then a number between 0 and 7 to indicate the desired intensity should follow. Any of the mnemonics of the same op code may be concatenated to form a compound command. A typical parameter command would be:

SC2 LPOF INT 4

which would have an octal value of 554.

2.1.2 Mode

Op Code: Mode			Stop Code	Clear Push-Button Flag	Mode				Clear Sector Bits	Clear Coord-inate Bits	Enter Data State
0	1	2	3	4	5	6	7	8	9	10	11
0	0	1									

The mode command has six separate functions: It can cause the display to stop and set the internal stop flag, clear push-button and manual interrupt flags, set the mode register, clear sector bits, clear coordinate bits, or enter data state. Only the mode change has an enable bit to prevent its being changed.

Bits	Interpretation
0,1,2	Op code: mode.
3	Stops the display and sets the internal stop flag when bit is 1.
4	Clears the push-button and manual interrupt flags when bit is a 1.
5	Enables mode change.
6,7,8	Determines one of eight possible mode settings if bit 5 is a 1.
9	Clears only the high order three bits in both the x and y position registers when bit is a 1.
10	Clears only the low order ten bits in the x and y position registers when bit is a 1.
11	When bit is 1, the next instruction is accepted as a data state word rather than control state. The display remains in data state until an escape is executed.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Code	Operation
EDS	1001	Enter data state.
CCB	1002	Clear coordinate bits.
CSB	1004	Clear sector bits.
POINT	1100	Set mode to 0.
INCR	1110	Set mode to 1.
VEC	1120	Set mode to 2.
VECON	1130	Set mode to 3.
SVEC	1140	Set mode to 4.
CHAR	1150	Set mode to 5.
GRAPH	1160	Set mode to 6.
CLDF	1200	Clear flag.
STOP	1400	Stop display.

2.1.3 Jump

Op Code: Jump			Scale			Light Pen		Push	Break Field		
0 0	1 1	2 0	3	4	5	6	7	8	9	10	11

First Word

Low Order						12 Bits of Address					
0	1	2	3	4	5	6	7	8	9	10	11

Second Word

The jump command is a 2-word (24-bit) instruction, 15 bits of which specify the new address. The 15 bits used are the last 3 bits in the first word, which specifies the memory field, and the entire second word, which specifies the address in one of the 4K memory banks. If bit 8 is a 1, the command causes two words to be entered on the end of the PDP list. For full explanation of the push jump command, see "Subroutining section." The jump command can also change the scale and light pen registers.

Bits	Interpretation
0,1,2	Op code: jump.
3	Enables scale change.
4,5	Determines one of four possible scale settings if bit 3 is a 1.
6	Enables light pen change.
7	Turns the light pen on when it is a 1, or off when it is a 0, provided bit 6 is a 1.
8	When bit is a 0, the command is a simple jump, the scale and light pen are changed, and the new 15-bit address is inserted in the DAC and the display continues from there. When bit is a 1, the command is a push jump. The old address and the status of the display are stored on the PDP list; then the new address and scale or light pen change is inserted.
9,10,11	Specifies the high order 3 (of 15) address bits for the jump or push jump command.
Second Word	
0,1,11	Specifies the low order 12 (of 15) address bits for the jump or push jump command.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Code	Operation
JUMP	2000	Jump to 15-bit address contained in last digit and the next word addressed.
PJMP	2010	Jump to subroutine addressed the same as JUMP.
LPOF	0040	Light pen off.
LPON	0060	Light pen on.
SC1	0400	Set scale to X1.
SC2	0500	Set scale to X2.
SC4	0600	Set scale to X4.
SC8	0700	Set scale to X8.

2.1.4 Pop

								Inhibit Restoring			
Op Code: Pop			Scale			Light Pen		Mode	Light Pen and Scale	Intensity	Enter Data State
0	1	2	3	4	5	6	7	8	9	10	11
0	1	1									

The pop command is the last word in a subroutine file. It causes the display to bring out the old address and old status from the PDP list. The status can be blocked from being reinserted by raising bits 8, 9, and/or 10. The new light pen and scale settings in the pop will always be inserted after the status is restored. The first instruction after the push jump can be a data state instruction by entering data state during the pop.

Bits	Interpretation
0,1,2	Op code: pop
3	Enables scale change.
4,5	Determines one of four possible scale settings if bit 3 is a 1.
6	Enables light pen change.
7	Turns the light pen on when it is a 1, or off when it is a 0, provided bit 6 is a 1.
8	The mode status from the PDP list will not be restored when bit is a 1.
9	The light pen and scale status from the PDP list will not be restored when bit is a 1.
10	The intensity status from the PDP list will not be restored when bit is a 1.
11	The display will be in data state when the word at the address taken from PDP list is executed.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Code	Operation
POP	3000	Exit from subroutine to next address after PJMP.
PEDS	3001	Pop and enter data state.
PNI	3002	Pop and inhibit restoring intensity.
PNLS	3004	Pop and inhibit restoring light pen and scale.
PNM	3010	Pop and inhibit restoring mode.
LPOF	0040	Light pen off.
LPON	0060	Light pen on.
SC1	0400	Set scale to X1.
SC2	0500	Set scale to X2.
SC4	0600	Set scale to X4.
SC8	0700	Set scale to X8

2.1.5 Conditional Skip (Bank 1)

Op Code: Conditional Skip (Bank 1)			Sense of Test	Clear Bits After Test	Complement Bits After Test	Selected Buttons 0-5 PB0	PB1	PB2	PB3	PB4	PB5
0 1	1 0	2 0	3	4	5	6	7	8	9	10	11

All display skip commands skip two display words if the condition tested fails. The display skips two instructions so that a JUMP or PJMP command (which are two words long) can be executed or not executed properly. The buttons to be tested should have 1s in the proper bits of the skip command. Using the clear and complement facilities, the buttons can be set in any desired configuration. The sense test bit determines whether the user is testing for 1s or 0s.

Bits	Interpretation
0,1,2	Op code: conditional skip (bank 1).
3	If bit is 0, the display skips two words if any of the indicated push buttons are 0. If bit is 1, the display skips two words if any of the indicated push buttons are 1.
4	Sets all the selected push buttons to the 0 state (light off) when it is a 1, independent of the outcome of the test.
5	Complements all the selected push buttons after the test when it is a 1, independent of the outcome of the test. Since the buttons are cleared before they are complemented, they can be set to the 1 state by having both bits 4 and 5 at 1.
6, ,11	Selected push buttons of bank 1; e.g., bit 6 = push button 0, bit 11 = push button 5.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Value	Operation
SKI	4000	Skip if any of the selected buttons are 0.
INV	0400	Invert sense of test (skip if any selected button is 1).
CLAT	0200	Clear buttons tested after test.
COAT	0100	Complement buttons tested after test.

2.1.6 Conditional Skip (Bank 2)

Op Code: Conditional Skip (Bank 2)			Sense of Test	Clear Bits After Test	Complements Bits After Test	Selected Buttons 6-11					
						PB6	PB7	PB8	PB9	PB10	PB11
0	1	2	3	4	5	6	7	8	9	10	11
1	0	1									

This command is identical to conditional skip (bank 1) except that it tests push buttons 6-11.

Bits	Interpretation
0,1,2	Op code: conditional skip (bank 2).
3	
4	Same as conditional skip (bank 2).
5	
6-11	Selected push buttons of bank 2; e.g., bit 6 = push button 6, bit 11 = push button 11.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Value	Operation
SK2	5000	Skip if any of the selected buttons are 0.
INV	0400	Invert sense of test (skip if any of the selected buttons are 1).
CLAT	0200	Clear buttons tested after test.
COAT	0100	Complement buttons tested after test.

2.1.7 Miscellaneous

2.1.7.1 Arithmetic Compare Push Buttons (Bank 1)

Op Code: Miscellaneous			Microprogrammed: Arithmetic Compare PB (0-5)			Push Buttons (0-5) PB0	PB1	PB2	PB3	PB4	PB5
0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	0						

Bits 6-11 of this command are compared to the contents of buttons 0-5 (bank 1). If all the bits and buttons match, the test succeeds and the display follows a normal sequence. If the test fails, the display skips two words.

2.1.7.2 Arithmetic Compare Push Buttons (Bank 2)

Op Code: Miscellaneous			Microprogrammed: Arithmetic Compare PB (6-11)			Push Buttons (6-11) PB6	PB7	PB8	PB9	PB10	PB11
0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	1						

Same instruction as above, except bits 6-11 are compared to push buttons 6-11.

2.1.7.3 Skip on Flags

Op Code: Miscellaneous			Microprogrammed: Skip on Flags			Skip Uncon-ditional	Skip if not Sector 0	Skip on Push-Button Hit Flag Bank 1	Bank 2	Spare	
0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	1	0						

All the commands will skip two words if the test fails. Sector 0 is defined as any point where the x and y position registers have all 0s in the high order 3 bits (of 13). The push button hit skip skips on the individual banks (1 or 2). Both flags are cleared by IOT 062 or by the display command CLDF (1200_8).

Bit(s)	Interpretation
0,1,2	Op code: miscellaneous.
3,4,5	Microprogrammed: skip on flags.
6	Do not execute the next two display words if bit is 1.
7	Do not execute the next two words unless the high order three bits of both the x and y position registers are 0; i.e., skip if the beam is not on the screen.
8	Skip if any push button 0-5 has been pushed.
9	Skip if any push button 6-11 has been pushed.
10,11	Spare.

2.1.7.4 Count

Op Code: Miscellaneous			Microprogrammed: Count			Count Scale		Count Intensity		Blink On	Off
0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	1	1						

The scale and intensity registers are also up-down counters. They cannot overflow; however, the scale register stays at 11_2 no matter how many count scale up commands are given. The blink facility allows the picture or any section of it to flash on and off at 1 cps (1/2 sec with the intensity on and 1/2 sec with the intensity off).

Bit(s)	Interpretation
0,1,2	Op code: miscellaneous.
3,4,5	Microprogrammed: count.
6	Enables count scale logic.
7	0: count scale up (unless at 11_2). 1: count scale down (unless at 00_2).
8	Enables count intensity logic.
9	0: count intensity up (unless at 111_2). 1: count intensity down (unless at 000_2).
10	Turn blink on, all intensified points will be gated through the blink logic.
11	Turn blink off.

2.1.7.5 Slave Logic (Optional)

Op Code: Miscellaneous			Micro-pro: Slaves	Group Number		Unit 0			Unit 1		
0	1	2	3	4	5	6	7	8	9	10	11
1	1	0	1								

The slave logic allows the display to control up to eight CRTs. The slave logic blanks or unblanks the intensity and light pen at each CRT for the display file being executed until the slave status is changed. IOT 072 and IOT 074 can read back the status of the eight slaves into the PDP-8.

Bit(s)	Interpretation
0,1,2	Op code: miscellaneous.
3	Specifies slave logic when it is a 1.
4,5	Choose one of four slave groups, with two units each, to be modified.
6	Enables change of unit 0 of slave group specified in bits 4,5.
7	Turns on light pen of unit 0 if bit is a 1; turns it off if bit is a 0.
8	Turns on the intensity of unit 0 if bit is a 1; turns it off if bit is a 0.
9	Enables change of unit 1 of slave group specified in bits 4,5.
10	Turns on light pen of unit 1 if bit is a 1; turns it off if bit is a 0.
11	Turns on the intensity of unit 1 if bit is a 1; turns it off if bit is a 0.

ASSOCIATED MNEMONICS AND VALUES

Mnemonic Symbol	Octal Value	Operation
SK3	6000	Arithmetically compare pushbuttons (0-5) with last two digits of instruction; skip if not equal.
SK4	6100	Same as SK3 but for buttons 6-11.
SKIP	6240	Unconditional skip (two locations).
SNSZ	6220	Skip if sector 0 flag is not up.
SPB1	6210	Skip if push button (0-5) flag is down.
SPB2	6204	Skip if push button (6-11) flag is down.
SCUP	6340	Count scale up.
SCDN	6360	Count scale down.
INTUP	6310	Count intensity up.
INTDN	6314	Count intensity down.
BKON	6302	Blink on.
BKOF	6301	Blink off.
SG0	6400	Set slave group 0.
SG1	6500	Set slave group 1.
SG2	6600	Set slave group 2.
SG3	6700	Set slave group 3.
SU0	0040	Turn light pen and intensity off on unit 0.
LP0	0060	Unit 0 light pen on.

ASSOCIATED MNEMONICS AND VALUES (continued)

Mnemonic Symbol	Octal Value	Operation
IT0	0050	Unit 0 intensity on.
SU1	0004	Turn light pen and intensity off on unit 1.
LP1	0006	Unit 1 light pen on.
IT1	0005	Unit 1 intensity on.

2.2 DATA STATE COMMANDS

All data state commands change the x and y position registers which are in turn connected through D to A converters to the CRT deflection system. The mode register determines the data state format used by the display. In control state only the mode command can change the mode. Seven of the eight possible modes are used. If the display tries to enter data state in the unused mode, (111_2), the display stops but no error flag is raised. Each of the seven modes has an escape mechanism to return to control state. Since most of the modes are different, each is described below.

Point, vector, and vector continue modes are two-word commands; a single command is specified by two consecutive locations in the display list. Both words must be brought out before execution, and therefore there are two input buffer registers. The register DX, which is used for all commands, receives its information directly from the data lines. If the command is two words long, the first input word is transferred to the DY register while the second input word is brought in to DX from memory. The only exception to this is data state increment mode words. In this case a single word command is executed from the DY register. The DX register is used for double buffering virtually eliminating the wait for input words. Short vector mode uses the DY register in order to appear as a normal vector. In other words, the delta Y portion of the command is transferred to the DY register.

2.2.1 Point Mode (Two Words)

Point	(000)										
Intensity	Inhibit	Y Position									
0	1	2	3	4	5	6	7	8	9	10	11

First Word (DY)

Point	(000)										
Escape	Inhibit	X Position									
0	1	2	3	4	5	6	7	8	9	10	11

Second Word (DX)

The basic action is to jam bits 2 through 11 of the first word (from DY) into the low order ten bits of the y position register, and the same bits in the second word into the x position register. The high order three bits in x and y remain unchanged. If bit 1 in either word is up, the contents of the associated position register are not changed during that command. This is useful if the user does not know the present beam position and wishes to change either x or y and leave the other at the same value. If bit 0 in the first y point word is a 1, the point specified is intensified when the beam reaches the proper position. If bit 0 in the second word (x point) is a 1, it indicates an escape and the next word is interpreted as a control state command. If the bit is a 0, the next word is interpreted as the first word of another point mode command. The scale setting has no effect in point mode.

Word	Bit	Interpretation
1	0	If bit is a 1, intensity given point.
	1	Inhibit changing y position register.
	2-11	New y coordinate (low order ten bits).
2	0	Escape to control state.
	1	Inhibit changing x position register.
	2-11	New x coordinate (low order ten bits).

2.2.2 Increment Mode

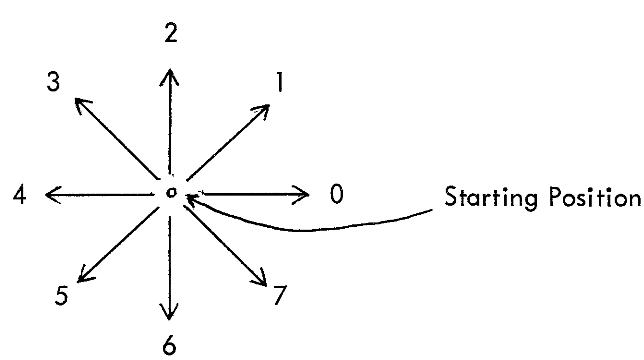
Increment	001										
Intensity	No. of Moves		Direction (0-7)			Intensity	No. of Moves		Direction (0-7)		
0	1	2	3	4	5	6	7	8	9	10	11

Increment mode is a mechanism for moving the beam a short distance in an efficient manner. The beam is moved from its previous position to a new position according to a 6-bit increment byte. Each byte is handled separately and executed independently of the other. Both bytes (first, 0-5, second, 6-11) are identical; therefore only the first will be discussed.

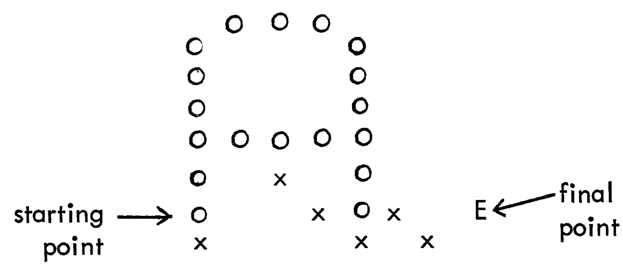
The byte is broken into three sections: first, to indicate whether the byte move is to be intensified (bit 0 (6)); second, to give the number of moves to be made (bits 1, 2 (7, 8)); and third, to specify the direction in which the move is to be made (bits 3, 4, 5 (9, 10, 11)). The beam is only intensified at discrete points according to the scale setting (see "Scale" section 1.), it is not run with the beam on during the motion. The increment can consist of one, two, or three moves per byte, with each move being one, two, four, or eight points apart on the raster depending on the scale. The two bits that control the number of moves are programmed as follows:

Bits 1, 2	Interpretation
00	Move the beam once in the indicated direction and escape.
01	Move the beam once; do not escape.
10	Move beam twice; do not escape.
11	Move beam three times; do not escape.

The three bits for direction indicate one of eight 45 degree directions:



The letter "A" is programmed in subroutine form as an illustration of the use of increment mode.



The "o" indicates an intensified point, the "x" a nonintensified point, and "E" the point at which the escape is given.

```
ALPHAA,   INCR    EDS    /octal (1111)
          1672           /byte 1 -↓ , 1 m, noninten. byte 2 -↑ , 3 m, inten.
          7251           /byte 1 -↑ , 3 m, inten. byte 2 - ↗ , 1 m, inten.
          6057           /byte 1 - → , 2 m, inten. byte 2 - ↘ , 1 m, inten.
          7674           /byte 1 -↓ , 3 m, inten. byte 2 - ←, 3 m, inten.
          3762           /byte 1 - ↘ , 3 m, noninten. byte 2 -↑ , 2 m, inten.
          2701           /byte 1 - ↘ , 2 m, noninten. byte 2 - ↗ , 1 m, noninten. and escape
          POP            /octal (3000)
```

Bit(s)	Interpretation
0	If bit is a 1, intensify the first byte.
1,2	Number of moves in the byte (00_2 ⇒ Move once and escape).
3,4,5	Direction in which moves are to be made.
6	If bit is a 1, intensify the second byte.
7,8	Same structure as byte 1 bits 1 and 2.
9,10,11	Same structure as byte 1 bits 9, 10, 11.

2.2.3 Vector Mode (Two Words)

Vector	(010)										
Intensify	$\pm$	10-Bit Delta Y									
0	1	2	3	4	5	6	7	8	9	10	11

First Word (DY)

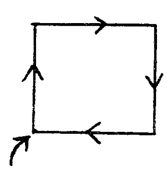
Vector (010)											
Escape	+ –	10-Bit Delta X									
0	1	2	3	4	5	6	7	8	9	10	11

Second Word (DX)

Vector Mode is used to draw long straight lines. The beam is moved from its present location in the direction and distance specified by the delta y and delta x. The deltas tell the beam how many moves to make in x and y before stopping. The number of raster points between each move is again dependent upon the contents of the scale register. At a scale of one, the 10-bit vector can take the beam from one end of the screen to the other. At a scale of eight, the beam can go to any point on the 13-bit "paper." Bit 1 in both words indicates the sign (direction) of the vector. A + is indicated by a 0 and is up for y and to the right for x. Bit 0 in the first word indicates whether the vector is to be intensified. Bit 0 in the second word is the escape to control state which indicates the end of a vector string.

The following display subroutine program places on the screen a 1-inch square. Since the box is drawn with vectors, it can be put anywhere on the screen (relocatable).

Direction of
Beam Movement



Starting and
final point

```
IBOX,   SC2 INT 5    /Control state octal (515)
        VEC EDS      /Control state octal (1121)
        4067         /Δy = +55.  intensify
        0000         /Δx =   0.
        4000         /Δy =   0.  intensify
        0067         /Δx = +55.
        6067         /Δy = -55.  intensify
        0000         /Δx =   0.
        4000         /Δy =   0.  intensify
        6067         /Δx = -55.  escape
        POP          /Control state octal (3000)
```

Word	Bits	Interpretation
1	0	If bit is a 1, intensify the vector.
	1	If bit is a 0, the sign of delta y given in bits 2-11 is positive; if a 1, it is negative.
	2-11	The 10-bit delta y.
2	0	If bit is a 1, the next word is interpreted in control state.
	1	If bit is a 0, the sign of delta x given in bits 2-11 is positive, if a 1, it is negative.
	2-11	The 10-bit delta x.

2.2.4 Vector Continue Mode (Two Words)

Vector Continue (011)											
Intensify	±	10-Bit Delta Y									
0	1	2	3	4	5	6	7	8	9	10	11
Escape		10-Bit Delta X									
0	1	2	3	4	5	6	7	8	9	10	11

Vector continue mode has the same format and action as vector mode, except the vector does not stop until it reaches the edge.

2.2.5 Short Vector

Short Vector (100)											
Intensify	±	Delta Y				Escape	±	Delta X			
0	1	2	3	4	5	6	7	8	9	10	11

Short vector mode is basically the same as vector mode except it is only one word long. In order to fit it into one word, the maximum number of increments has been reduced from 1,024 to 16 in x and y directions. Bits 0 and 1 of a short vector word correspond to bits 0 and 1 of the first word of a vector command. Bits 2, 3, 4, and 5 correspond to 8, 9, 10, and 11 of the first word; bits 6 and 7 to bits 0 and 1 of the second word; and bits 8, 9, 10, and 11 to bits 8, 9, 10, and 11 of the second word. In execution of a short vector mode command, the left half of the input word is transferred to DY and bits are rearranged to be in the same format as a vector mode command.

Bits	Interpretation
0	If bit is a 1, intensify the vector.
1	If bit is a 0, the sign of delta y in bits 2, 3, 4, and 5 is positive; if a 1, it is negative.
2,3,4,5	The 4-bit delta y.
6	If bit is a 1, the next word is interpreted in control state.
7	If bit is a 0, the sign of delta x given in bits 8, 9, 10, and 11 is positive; if a 1, it is negative.
8,9,10,11	The 4-bit delta x.

2.2.6 Character Mode (Optional)

Character (101)											
Character 1						Character 2					
0	1	2	3	4	5	6	7	8	9	10	11

Six-bit format

Ignored					Character						
0	1	2	3	4	5	6	7	8	9	10	11

Seven-bit format

The character generator can be run in two different formats: 6-bit format, packed two to a word but directly referencing only 64 characters, or 7-bit format, only one character per word but referencing 128 different characters. The character generator is simply an efficient dispatch system for instructions stored in core memory. The characters, therefore, are programmable as well as the dispatch table that calls them. The details of programming the character generator are given in appendix 1.

Special characters available in the character generator include set and count intensity and scale, set light pen, change case, change character set, change code size (6-bit versus 7-bit), carriage return (clear x register), and escape to control state.

2.2.7 Graphplot Mode

Graphplot (110)											
Escape	Set Y Set X	X or Y Coordinate									
0	1	2	3	4	5	6	7	8	9	10	11

Graphplot mode is a concise way of describing and displaying tabular data. The execution of a data command is as follows: First, the x or y coordinate is incremented once; then the other coordinate is set by bits 2-11 of the graphplot mode command. Bit 1 of the word specifies whether x is to be incremented and y plotted (bit 1-0), or y incremented and x plotted (bit 1-1). If bit 0 is a 1, the display escapes to control state.

The incremented axis moves one, two, four, or eight points depending on the scale setting. Since one axis is always incremented, the graph is easily translated in this direction by changing the starting location.

Bit(s)	Interpretation
0	If bit is a 1, escape to control state.
1	If bit is a 0, increment x coordinate and set the y coordinate with bits 2-11; if bit is a 1, increment y and set x.
2-11	Ten bits of coordinate information.

2.3 DISPLAY ORIENTED COMPUTER INSTRUCTIONS

The PDP-8 has a family of instructions * which it uses to communicate with I/O equipment. A group of these instructions is assigned to the display (IOTs 05, 06, 07, 13, 14, 15, 16, 17, 30). Given below are the display IOTs, their mnemonics, and their functions. They are broken into three classes: the first requests information from the display; the second sends information to the display; and the third class is the computer skip on display flags. The information transfer is done through the PDP-8 accumulator; so if the user expects information, he should clear the AC beforehand, and if sending information, he should have the data in the AC before giving the IOT.

2.3.1 Group 1. From the Display

2.3.1.1 RPDP 6051 Read Push Down Pointer - A 1s (inclusive OR) transfer from the push down pointer (12 bits) to the AC is done. The PDP should be pointing to the location in which status information will go on the next push jump (if it comes before a pop). Reading the PDP, subtracting the set value, and dividing by two gives the level of the subroutine.

2.3.1.2 RXP 6052 Read x Position Register - A 1s transfer from the x position register to the AC is done. Only the low order 12 (of 13) bits are transferred; the high order bit must be obtained from the RS2 instruction.

2.3.1.3 RYP 6054 Read y Position Register - Same as RXP, except the y position register is transferred.

2.3.1.4 RDAC 6061 Read Display Address Counter - The contents of the display address counter are transferred from the display to the AC. The DAC will be set at the next command to be executed by the display.

2.3.1.5 RS1 6062 Read Status 1 - Status 1 consists of the state of all display flags and the contents of the break field register. If the flag is up the associated bit is a 1.

*See PDP-8 User Handbook for details.

Bit(s)	Interpretation
0	Light pen hit flag.
1	Vertical edge flag. The y position register has overflowed.
2	Horizontal edge flag. The x position register has overflowed.
3	Internal stop flag.
4	Sector 0 flag. If bit is a 1, the display is in sector 0.
5	Control state flag. If bit is a 1, the display is in control state, if it is a 0 the display is in data state.
6	Manual interrupt flag.
7	Push-button hit flag.
8	Display interrupt flag. If the interrupt system is turned on and bit is a 1, the computer will interrupt. It is set by one of the six display flags being on and gated onto the interrupt line.
9,10,11	Contents of break field register. These three bits and the twelve bits from the RDAC instruction give the full 15-bit memory address.

2.3.1.6 RS2 6064 Read Status 2 - Status 2 consists of the contents of some of the major registers in the display; e.g., light pen scale, mode, and intensity. It also contains byte information and the high order bit of the x and y position registers. The byte flip-flop indicates whether the left half or right half byte in increment mode was being executed when the display stopped. It does not tell whether the right or left hand character is being executed; this information is obtained from the RCG (IOT 304) instruction. The low order twelve bits of the 13-bit x and y position register are obtained by giving RXP or RYP.

Bit(s)	Interpretation
0	Byte flip-flop. If bit is a 0, the left hand increment is being executed; if bit is a 1, the right hand byte is being executed.
1	Light pen enable. If bit is a 1, the light pen is enabled.
2	High order y position register bit.
3	High order x position register bit.
4,5	Scale
6,7,8	Mode
9,10,11	Intensity

2.3.1.7 RPB 6071 Read Push Buttons - The contents of the twelve push buttons (0-11) are transferred into the corresponding AC bits.

2.3.1.8 RSG1 6072 Read Slave Group 1 - On this instruction, the light pen, light pen hit, and intensity status forces slaves 0,1,2, and 3. The control state command "set slaves" sets the light pen and intensity status. If the slave option is not present, the IOT reads back 0s into the accumulator.

Slave 0			Slave 1			Slave 2			Slave 3		
Light Pen	Intensity	Light Pen Hit									
0	1	2	3	4	5	6	7	8	9	10	11

AC Format

Bit(s)	Interpretation
0	Light pen enable, slave 0.
1	Intensity status of slave 0.
2	Light pen hit, status slave 0.
3,4,5	Same format as above for slave 1.
6,7,8	Same format as above for slave 2.
9,10,11	Same format as above for slave 3.

2.3.1.9 RSG2 6074 Read Slave Group 2 - RSG2 has the same format as RSG1, except it reads status of slaves 4,5,6, and 7.

2.3.1.10 RCG 6304 Read Character Generator - RCG reads in the five character generator parameters: character generator active (CHACT), character byte (CB), case, code size (CHSZ), and starting address register (SAR). The CHACT bit indicates whether the character generator is in use; i.e., the display is in character mode and data state. The CB shows whether the left or right half character (6-bit format) is being executed. The case bit is used (6-bit format) as a seventh bit to allow referencing either the lower or upper set of 64 characters. The CHSZ indicates whether the 6-bit or 7-bit character format is to be used. The SAR is a 6-bit register that indicates the starting address of the character dispatch table (see appendix).

Char-acter	CB	Spare	Case	CHSZ	Spare	SAR					
0	1	2	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0	If bit is a 1, the character generator is active.
1	If bit is a 0, left character is being executed. If bit is a 1, right character is being executed.
2	Spare.
3	If bit is a 1, upper case is in use, characters 65-128.
4	If bit is a 0, the character generator is using 6-bit format; if bit is a 1, the CG is using 7-bit format.
5	Spare.
6-11	Contents of the 6-bit SAR.

2.3.2 Group 2. To the Display

2.3.2.1 SPDP 6135 Set the Push Down Pointer - The contents of the AC are transferred into the PDP register. Since the PDP is a 12-bit register, the PDP list must reside in the first 4K of memory.

2.3.2.2 SIC 6145 Set Initial Conditions - SIC sets up a number of status registers in the display. The instruction enables four display flags onto the interrupt line, sets the paper size to 10, 11, 12, or 13 bits in x and y and light pen conditions. There are three options in the event the display is resumed after a light pen hit. The light pen can be left on, it can be turned off completely, or it can be turned off until the completion of the present command, then automatically turned back on at the next data request. There is also a register that tells the display to ignore all edge flags; therefore when the position register overflows, the edge flag is inhibited and the display continues in a normal fashion. Another register overrides the intensification bit in data state, causing all beam movements to be intensified. This feature is used principally for diagnostic purposes.

Bit(s)	Interpretation
0	Enable edge flag interrupt.
1	Enable light pen flag interrupt.
2	If bit is a 0, do not disable light pen after the resume; if bit is a 1, bit 3 indicates when to reenable the light pen.
3	If bit is a 0, reenable light pen on the first data request after the display is resumed. If bit is a 1, the light pen hit is equivalent to a LPOF command.
4,5	Set Y dimension. 00: 9.375 inches (10 bits) 01: 18.75 inches (11 bits) 10: 37.5 inches (12 bits) 11: 75.0 inches (13 bits)
6,7	Set X dimension, same as Y.
8	Intensify all points.
9	Inhibit edge flags.
10	Enable interrupt on push-button hit.
11	Enable interrupt on internal stop flag

2.3.2.3 LBF 6155 Load Break Field - This instruction has two functions. First, it loads the break field register when initializing the display; second, it sets the push buttons. Both functions have enable bits so that one may be executed without the other. If neither enable bit is up, both IOT pulses have other meanings (STPD-6154 and SPSI-6151).

Break Field				Push Buttons							
0	1	2	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0	Enable change of break field.
1,2,3	New break field.
4	Enable change of push buttons.
5	If bit is a 0, set push buttons 0-5 according to AC bits 6-11; if bit is a 1, set push buttons 6-11 according to AC bits 6-11.
6-11	New push-button states.

2.3.2.4 SCG 6303 Set Character Generator - SCG sets the SAR case and CHSZ.

Spare			Case	CHSZ	Spare	SAR					
0	1	2	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0,1,2	Spare.
3	Set case 0-lower 64 1-upper 64
4	Set code size 0-6 bit character format 1-7 bit character format
5	Spare
6-11	Starting address register

2.3.2.5 INIT 6165 Initialize the Display - The AC is transferred to the low order twelve bits of the DAC, and the display is initialized by raising the break request flag. The display will run uninterrupted until a flag is raised.

The five IOTs given (2.3.2.1-2.3.2.5) comprise a standard startup procedure for the display. The following program is given as an example.

```
        /Display start-up routine for
        /338 with VC38 Character Generator.
Initial, CLA                    /Clear AC.
        TAD XPDP                /Starting address for PDP, into AC.
        SPDP                    /IOT to transfer AC to PDP.
        CLA                     /Clear AC.
        TAD XSIC                /Constant to set initial conditions.
        SIC                     /Transfer AC to initial condition register.
        CLA                     /Clear AC.
        TAD XLBF                /Constant for BF and/or PB.
        LBF                     /Transfer AC to BF.
        CLA                     /Clear AC.
        TAD XCG                 /Constant for character generator.
        SCG                     /IOT to transfer AC to CG.
        CLA                     /Clear AC.
        TAD XIN                 /12-bit starting address for display.
        INIT                    /IOT to transfer AC to DAC and initialize display.
        CLA                     /Clear AC.
        ION                     /Turn interrupt on.
        JMP .                   /Display is now running.
XPDP,   7000                    /Starting address of push down list.
XSIC,   2367                    /Enable light pen, push button, and internal stop
                                /flags, set paper size to 75" by 75". Leave light
                                /pen on after light pen hit, and enable all edge
                                /flags.
XLBF,   4000                    /The display program starts in memory core zero.
                                /Note the enable bit must be a 1 to change the
                                /break field register.
XCG,    0016                    /Set code size to 6-bit format, start in the lower
                                /case, and the CG dispatch table starts in location
                                /16000 (memory bank 1 location 6000).
XIN,    DISSTT                  /Symbolic address of the first location in the display
                                /file.
```

2.3.2.6 RES1 6174 Resume After Light Pen Hit, Edge, or External Stop Flag - This IOT tells the display to resume the sequence of instructions from the point at which it stopped. In the case of a light pen hit or edge flag, the display completes the vector it stopped on before continuing to the next. One of the above flags must be up when RES1 is given; otherwise, the instruction has no effect. RES1 clears the flag before the display is reinitialized. The contents of the AC have no effect during this instruction.

2.3.2.7 RES2 6164 Resume After Stop Code - RES2 restarts the display after an internal stop flag and clears the flag before resuming. The AC must be zero before RES2 is given.

2.3.2.8 CFD 6161 Clear Display Flags - CFD clears the four flags that stop the display. This command is given when the display is not to be used any longer, but has been used in this program. The power clear pulse (START key) also clears all display flags. All display flags can be cleared by giving three IOTs: CFD-6161 (internal and external stop, light pen hit, and edge); RS1-6062 (push button); and SPMI-6172 (manual interrupt).

2.3.2.9 STPD 6154 Stop Display (External) - STPD stops the display and sets the external stop flag (see "Display Flags") when the display has stopped. This is one of the microprogrammed IOTs and requires bits 0 and 4 of the AC to be 0 when the IOT is given.

2.3.3 Group 3. IOT Skip on Display Flags

2.3.3.1 SPLP 6132 Skip on Light Pen Hit Flag - Pertains only to the master scope's light pen. If the flag is up, the computer skips one instruction.

2.3.3.2 SPSP 6142 Skip on Slave Light Pen Hit Flag - If any of the eight slave light pen flags are up, the computer skips. The particular display can be found by giving the RSG1 and RSG2 IOTs and interrogating the AC.

2.3.3.3 SPES 6151 Skip on External Stop Flag - This is a microprogrammed instruction and requires bits 0 and 4 of the AC to be 0 when the instruction is given.

2.3.3.4 SPEF 6152 Skip on Edge Flag - SPEF causes a computer skip if either the horizontal or vertical edge has been violated. The edge violated can be found by giving the RS1 IOT. If the display runs off the corner of the paper, both the horizontal and vertical edge flags will be up.

2.3.3.5 SPSF 6171 Skip on Internal Stop Flag - The computer skips if the display has executed a stop code and SPSF is given.

2.3.3.6 SPMI 6172 Skip on Manual Interrupt - SPMI causes the computer to skip if the manual interrupt flag is on. It also clears the flag and the light in the push button if it is up.

APPENDIX 1
VC38 CHARACTER GENERATOR

The VC38 is a dispatch type generator, with both the dispatch table and the execution routines stored in core memory. The 6- or 7-bit character is used to index a special 15-bit register (CHAC) which contains the starting address of the dispatch table. The word from the referenced location is then used to index the CHAC to get to the location of the beginning of the variable length execution routine. At the end of the routine, an escape code is given which directs the CG (character generator) to accept the next character and restart the process. There are also special dispatch words (control characters) which do not cause a dispatch but rather are direct commands to the logic.

The beginning of the dispatch table is specified by the SAR (starting address register) which is 6-bits long. The SAR is gated to the upper six bits of the CHAC which in turn is gated onto the MA (memory address) bus. As an example, if the SAR contains 16_8, the dispatch table begins at location 6000_8 in core memory 1. The SAR is set and read by the PDP-8 via IOTs (SCG and RCG respectively).

The characters are interpreted in 6- or 7-bit format depending on the contents of the 1-bit code size register (CHSZ). If the register is a 1, the low order seven bits of the data word are gated into the low order seven bits of the CHAC, and the SAR is gated into the CHAC to produce the dispatch table address. Thus, if the SAR is 04_8 and the character is 116_8, the word in location 4116_8 contains the dispatch address.

In the 6-bit format, an identical process takes place except that the leftmost six bits are first gated onto the CHAC and the right six bits are put in a character save (CS) register. The CS register is gated onto the low order six bits of the CHAC when the first character is complete. The CB register is also set to a one, indicating execution of the left character.

Along with the six character bits, the CASE bit is gated into bit 8, allowing reference to 128 characters by a 6-bit character code.

CHAC (Six Bit Code Format)														
SAR						Set to Zero		Case	6 Bit Character Code					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

The dispatch word obtained from the location specified by the CHAC is interpreted in the same way whether the six or seven bit format was used.

In the dispatch word, bit position 0 indicates whether the dispatch table word is a control word (bit 0=1) or a dispatch word (bit 0=0). If it is a dispatch word, the character can be drawn in either increment or short vector mode. Bit 1 in the dispatch word indicates the mode. If the bit is 0, the data is accepted in increment mode; if it is 1, the data is accepted in short vector mode. The low order ten bits are used as the dispatch address and are gated onto the low order ten bits of the CHAC.

Dispatch Word											
	Mode	Address									
0 0	1	2	3	4	5	6	7	8	9	10	11

CHAC During Dispatch														
SAR					Over-Lap	Dispatch Address								
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Both the SAR and the dispatch address reference bit 5 in the CHAC. The bit is the inclusive OR of these two registers. In other words, if the SAR is odd (i.e., bit 5 is a 1), the dispatch address can only reference 512_{10} locations, whereas, if the SAR is even, the table can reference 1024_{10} locations.

A1.1 CONTROL CHARACTERS

If the dispatch table word has bit 0=1, a dispatch does not take place, but rather the word is accepted in one of three special formats indicated by bits 1 and 2. After execution of the control character, the next character is immediately fetched.

A1.1.1 Parameter Control

Parameter control format is specified by 00_2 in bits 1 and 2 of the dispatch table word. Bits 3-11 are then interpreted in the same format as the parameter mode word of control state.

Dispatch Table Word (Parameter Control)											
			Scale			Light Pen		Intensity			
0 1	1 0	2 0	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0	=1 Control character.
1,2	$=00_2$ Parameter control.
3	Enables scale change.
4,5	Determine one of four possible scale settings if bit 3 is a 1.
6	Enables light pen change.
7	Turns the light pen on when it is a 1, or off when it is 0, provided bit 5 is a 1.
8	Enables intensity change.
9,10,11	Determine one of eight possible intensity settings if bit 8 is a 1.

A1.1.2 Table Control

The table control character is specified by having 101_2 in bits 0-2 of the dispatch table word. This control character has the ability to change the CASE bit and the SAR.

			Case	Enable Bits 6-8	Enable Bits 9-11	SAR (0-2)			SAR (3-5)		
0 1	1 0	2 1	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0	=1 Control character.
1,2	=01 Table control.
3	Case bit is set by this bit for referencing, 128 characters using 6-bit code size.
4	Enable changing SAR bits 0-2 from bits 6-8 of control word.
5	Enable changing SAR bits 3-5 from bits 9-11 of control word.
6-8	Set SAR bits 0-2 if bit 4 is a 1.
9-11	Set SAR bits 3-5 if bit 5 is a 1.

A1.1.3 Miscellaneous Control

The miscellaneous control character is specified by 110_2 in bits 0-2 of the dispatch table word. In this format the code size can be changed, the intensity and scale can be counted up and down, and carriage return (clear low order ten bits of x position register), and escape to control state (end character string, leave character generator) can be executed.

Dispatch Table Word (Miscellaneous Control)											
			CHSZ		Carriage Return	Escape	Count Scale		Count Intensity		Spare
0 1	1 1	2 0	3	4	5	6	7	8	9	10	11

Bit(s)	Interpretation
0	=1 Control character.
1,2	=10 Miscellaneous control.
3	Enable change of code size.
4	Set code size to six bit, if bit is 0; set to seven bit, if bit is a 1; if bit 3 is a 1.
5	Clear low order ten bits of x position register (carriage return) if bit is a 1.
6	Escape to control state if bit is a 1.
7	Enable count scale logic.
8	0: count scale up 1: count scale down

Bit(s)	Interpretation
9	Enable count intensity logic
10	0: Count intensity up 1: Count intensity down
11	Spare

```
        /Program example of the use of the VC38 Character Generator.
Initial, CLA                              /Display start up
        TAD XPDP                          /See programming manual
        SPDP
        CLA
        TAD SIC
        SIC
        CLA
        TAD XLBF
        LBF
        CLA
        TAD XCG
        SCG
        CLA
        TAD XIN
        INIT
        CLA
        ION
        JMP ·
        XPDP,   7000
        XSIC,   2367
        XLBF,   4000
        XCG,    0006
        XIN,    DISSTT
                                          /Beginning of the display file
        DISSTT, POINT EDS CCB CBS         /Enter point mode
                1000                      /Set point to middle of
                5000                      /the screen and escape
                CHAR EDS                  /Enter character mode
```

```
          0001              /Call character (00) -set scale and intensity then character
                            /(01) -alpha numeric A.
          0203              /Carriage return (02) -line feed (03)
          0104              /Alpha numeric A (01) -escape to control state (04)
          JUMP              /Display jump to top of display file.
          DISSTT
*6000
D,        4555              /Parameter control:SC2 LPOF, INT5
          ALPA-D            /Dispatch in increment mode to alpha A.
          6100              /Miscellaneous control:  carriage return
          LINFD-D+2000      /Dispatch in short vector mode to line feed routine:
          6040              /Miscellaneous control:  escape to control state.
          .                 /The rest of the dispatch table
*D+200
ALPA,     1672              /Alphanumeric A in
          7251              /Increment mode:  see programming manual for details.
          6057
          7674
          3762
          2701
LINFD,    3140              /Short vector line feed - (9 10) points in the minus
                            /y direction
ALPB,     etc.              /The rest of the character routines
          .
          .
          .
          .
```

APPENDIX 2
PROGRAMMING EXAMPLES

```
        /Interrupt handler, the display is
        /placed first, however, a device with critical
        /timing should be before the 338, e.g., tape or drum.
          CLA
INTER,    SPLP                  /Skip on light pen flag
          SKP
          JMP LPR               /Jump to light pen routine
          SPSF                  /Skip on internal stop flag
          SKP
          JMP SFR               /Jump to stop flag routine
          SP MI                 /Skip on manual interrupt
          SKP
          JMP MIR               /Jump to manual interrupt routine
          SPEF                  /Skip on edge flag
          SKP
          JMP EFR               /Jump to edge flag routine
          SPES                  /Skip on external stop flag
          SKP
          JMP ESR               /Jump to external stop routine
          SPSP                  /Skip on slave light pen routine
          SKP
          JMP SLR               /Jump to slave light pen routine
          RS 1                  /Read status 1
          AND (20               /Test bit 7 for pushbutton flag
          SZA CLA
          JMP PBR               /Jump to pushbutton routine
                                /continue checking other I/O equipment
```

```
        /The following display file can be used
        /to control the scale, intensity, and
        /blink of a picture following the routine.
HEADR1, SK1 40 CLAT                  /Skip 2 locations if pushbutton 0 is a 0, clear after
                                     /test.
        SC UP                        /Count the scale up if pb0 was 1.
        0                            /Nop
        SK1 20 CLAT                  /Skip if pb1 is 0, clear after test.
        SCDN                         /Count the scale down if pb1 was 1.
        0                            /Nop
        SK1 10 CLAT                  /Skip if pb2 is 0, clear after test.
        INT UP                       /Count intensity up
        0                            /Nop
        SK1 4 CLAT                   /Skip if pb3 is 0, clear after test.
        INT DN                       /Count intensity down.
        0                            /Nop
        SK1 2 CLAT                   /Skip if pb4 is 0, clear after test.
        BKON                         /Turn the blink on.
        0                            /Nop
        SK1 1 CLAT                   /Skip if pb5 is 0, clear after test.
        BKOF                         /Turn the blink off.
        0                            /Nop
        /Place the rest of the display
        /file here.
```

```
        /The following routine is used to
        /translate a picture if the picture is
        /closed in x-y and contains no point
        /or graphplot mode instructions.
HEADR2,     SK2 40                  /Skip if pb6 is off.
            SVEC EDS                /Short vector enter data state.
            41                      /Δx=1; shift picture to the right.
            SK2 20                  /Skip if pb7 is off.
            SVEC EDS
            61                      /Δx=-1; shift picture to the left.
            SK2 10                  /Skip if pb8 is off.
            SVEC EDS
            140                     /Δy=1; shift picture up.
            SK2 4                   /Skip if pb9 is off.
            SVEC EDS
            2140                    /Δy=-1; shift picture down
            SK2 2 CLAT              /Skip if pb10 is off; clear after test.
            CCB CSB                 /Clear coordinate and sector bits.
            0                       /Nop
            SK2 1                   /Skip if pb11 is off.
            SK2 74 CLAT             /Clear pb's 6,7,8,9 if
            0                       /pb11 is on.  Causes single
            0                       /move each time one of the buttons is pushed.
            STOP                    /Stop the display and record the new starting
                                    /coordinates.
```

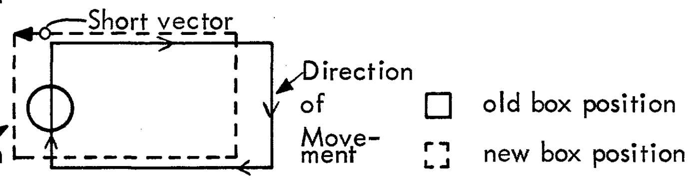
```
          /This routine produces a raster
          /with every eighth point on the
          /screen intensified.  The raster is
          /put up if pb5 is on.  The y overflow
          /must be set at greater than 10 bits.
RAST,       SK1 1 INV                     /Skip if pb5 is on
            JUMP                          /Jump over the raster
            MAST                          /Routine
            SC8 LPON INT 6 CCB CSB        /Set beam to lower left corner.
RAS,        VEC EDS
            4001                          /INTENSIFY Δy=1
            4177                          /ESCAPE Δx=177 (across the screen at scale 8).
            POINT EDS
            2000                          /Do not change y coordinate.
            4000                          /ESCAPE. Set x to 0.
            SNSZ                          /Have we run off the top of the screen yet?
            JUMP                          /No! Go back and generate
            RAS                           /Another line of the raster.
            LPOF                          /Shut light pen off.
            VEC EDS CCB CSB               /This vector
YVECI,      0                             /is changed when picture is
XVECI,      4000                          /shifted to get back to virtual starting point.

MAST,       JUMP                          /Continue on.
            PICT
```

```
          /Program to keep box under the
          /light pen, assuming startup IOT's
          /were given and standard INTERRUPT
          /system is some place in core.
           CLA                       /PDP-8 got to this location through interrupt handler
                                     /given above.
LPR,       RPB                       /Read in contents of 12 pushbuttons.
           AND (76                   /Keep contents of buttons 6-10 set 11 to 0.
           TAD (300                  /Add enable and bank bit.
           LBF                       /Give the IOT to clear pb11.
           CLA                       /IOT's do not clear AC.
           RES1                      /Resume display sequence after light pen hit.
           ION                       /Turn interrupt system on.
           JMP                       /Wait for next pen hit.

DISSTT,    SC2 LPON INT6
           VEC EDS CSB
           4050                      /Δy=40; intensify
           4000                      /Δx=0; escape
           SK2 1 INV CLAT COAT
           SVEC EDS
           61                        /Δy=0,Δx=-1; escape
           VEC EDS
           4000                      /Δy=0; intensify
           4050                      /Δx=40; escape
           SK2 1 INV CLAT COAT
           SVEC EDS
           140                       /Δy=1,Δx=0; escape
           VEC EDS
           6050                      /Δy=-40; intensify
           4000                      /Δx=0; escape
           SK2 1 INV CLAT COAT
           SVEC EDS
           41                        /Δy=0, Δx=1; escape
           VEC EDS
           4000                      /Δy=0; intensify
           6050                      /Δx=-40; escape
           SK2 1 INV CLAT COAT
           SVEC EDS
           2140                      /Δy=-1, Δx=0; escape
           JUMP
           DISSTT
```

/Program works as follows:
/Pushbutton 11 is sensed at the end of
/each side. If the button is on the
/next side is drawn, if the button is
/off a short invisible vector is executed
/in the proper direction to keep the box
/under the pen
/



/
/ light pen
/The button is turned back on
/then the next side is drawn.

APPENDIX 3
REFERENCE TABLES

(to be supplied)

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

PRINTED IN U.S.A. 10/66