GØ

# GENIE GRAPHICAL INPUT/OUTPUT SYSTEM

Gary D. Hornbuckle

University of California, Berkeley

Document No. 30.80.10

Issued February 28, 1966

Revised July 1, 1966

Office of Secretary of Defense

Advanced Research Projects Agency

Washington 25, D.C.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

## 1.0 Introduction

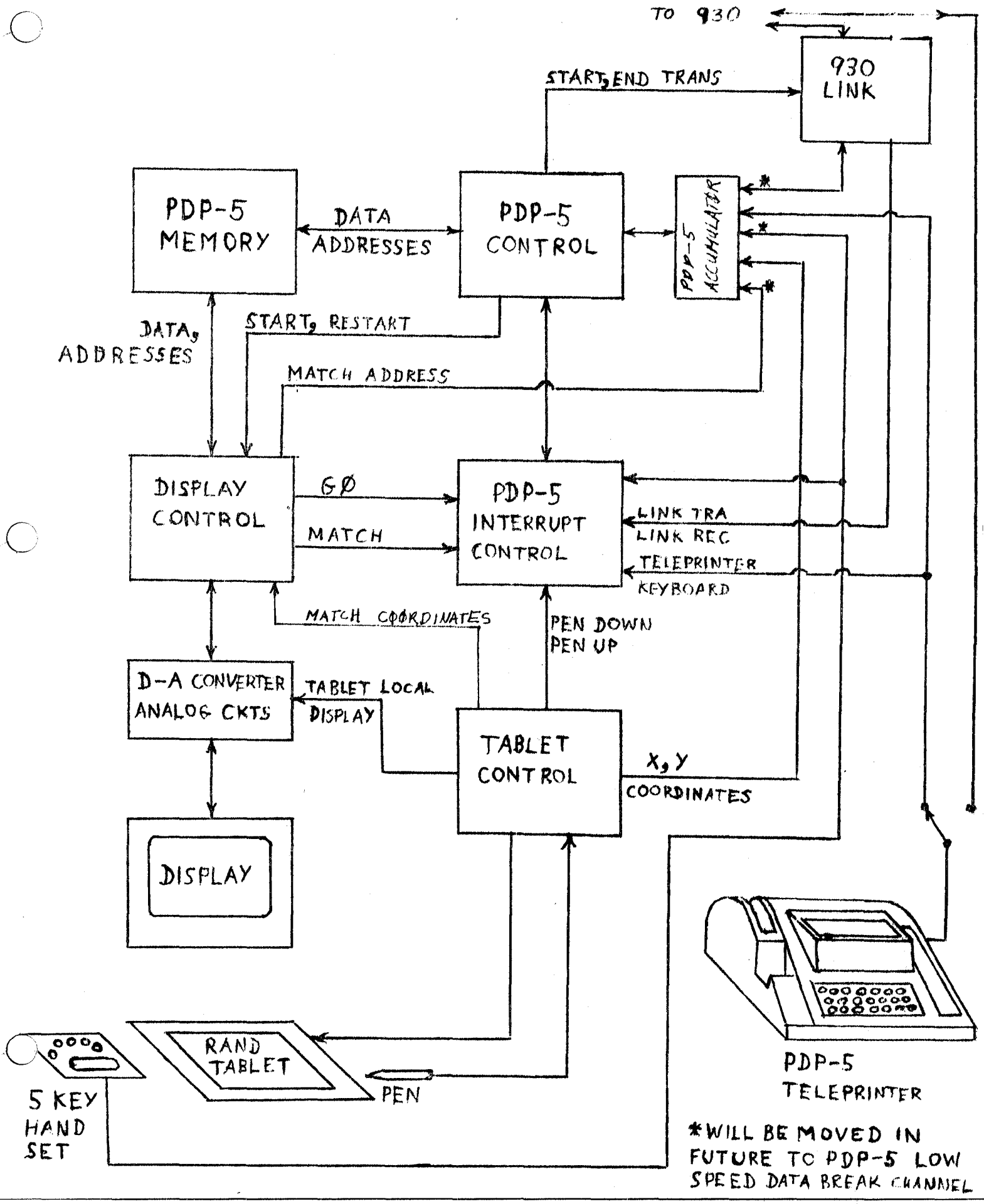
GØ is an integrated graphical input-output facility (see Fig. 1) to the ARPA SDS 930 time sharing system which provides rapid response for highly interactive man-machine studies. Included is a remotely located PDP-5 with a 12 x 12 inch CRT display console and a RAND tablet. Software is currently provided to aid in writing and debugging machine-language programs. These include INT (PDP-5 Interrupt Processing Monitor), ARPAL (PDP-5 Symbolic assembler and debugger) and CARP (for converting ARPAS assembled PDP-5 programs).

The display is updated directly from the PDP-5 4K memory; approximately 2000 characters (from an alphabet of 128), 2000 inches of lines, or 8000 short vectors (memory limitation) are available at 30 frames per second. The RAND tablet provides smoothed 10-bit X, Y coordinates, in a 1 to 1 relationship to the display coordinate system, every 5 ms.

Information is transmitted to or from the 930 at a 50KC bit rate. The PDP-5/930 Link converts parallel words from the computer to serial for transmission which is in one direction only at a time (half duplex).

The PDP-5 is standard except for the addition of hardware for rapid handling of I/Ø through interrupts, IØT instructions peculiar to the devices attached, and a semi-automatic loader. A hardware Program Counter may soon be added to speed up the machine by one cycle per instruction.

FIGURE 1: GØ HARDWARE



# 2.0 Initialization

## 2.1 Initialization - Hardware

Naturally the first requirement is that Power be turned on. This is accomplished by separate switches for the PDP-5 and Display console; both have internal power cycling which is 60 second up and 10 second down for the Display and a 10 second up and down for the PDP-5. The Display should never be cycled with the PDP-5 running, as its program will be disturbed, and the Display Intensity control should be fully counterclockwise.

The PDP-5 will run normally with the Display off -- the inverse is obviously not true.

When the Display is fully cycled up, the green GO light comes on. This button should then be pressed.

## 2.2 Initialization - Software

### 2.2.1 PDP-5 Bootstrap

If no program is in the PDP-5, one of the two bootstraps is loaded from the special switch box by:

1) turn switch to position 1
2) Hit LOAD ADDRESS on PDP-5
3) Turn to position 2
4) Hit DEPOSIT on PDP-5
5) Repeat steps 3 and 4 for all remaining switch positions
6) Repeat steps 1 and 2
7) Set switch to position 0
8) Hit START on PDP-5

The top bootstrap switch is for paper tape in RIM or BIN format*; the bottom for the 930-LINK. If the time sharing system is up,

---

* These are DEC standard formats -- see Appendix A.

the LINK bootstrap requires no user interaction through a 930-teletype console**. The 930 will transmit a standard system program called GØ to the PDP-5 which will provide the user with the currently available GØ system options through the Display console.

2.2.2 GØ System Program

When the GØ system has been fully loaded into the PDP-5, a display will appear (provided the GØ button has been pressed and the intensity turned up sufficiently) with several light-button options actuated by the RAND tablet pen. These options will vary as new software is added; the light button functions will be obvious from the display.

If it is ever necessary to halt the PDP-5 while GØ is in control, its starting address is $200_8$. However, in all anticipated systems, pressing the GØ button will reinitialize the current system in control, which may in turn provide the user with options to bring into control a higher-level program (the highest level being GØ). It should never be necessary to HALT the PDP-5, once it has been started.

Obviously it is not necessary to operate under GØ software control; this document should provide sufficient information for all types of use of the GØ hardware.

---

**This may be true in the future; however, currently one must ENTER the time sharing system in the usual manner via the PDP-5 teletype connected as a 930 teletype and then GØ TØ 'GØ' after the Link bootstrap is started.

## 3.0 User Created PDP-5 Programs

It is hoped that someday a sufficient set of "standard" systems will exist so that few users will find it necessary to write PDP-5 programs. Until that time, however, PDP-5 code may be created with the aid of either the 930-symbolic assembler, ARPAS, or the PDP-5 DDT-like program, ARPAL*.

To assemble with ARPAS, one appends, on the beginning of his symbolic file, a set of special op-code definitions** and assembles in the usual manner. The binary is then read by CARP***, which converts the 24-bit ARPAS produced code to 12-bit PDP-5 words which can then be punched onto paper tape or sent directly over the PDP-5/930 Link.

Of special interest are the display data macros contained in ASP:

| | |
|---|---|
| COORD | for line mode coordinates |
| SYMBOL | for symbol mode |
| VECTOR | for vector mode |
| POSYM | for position symbol mode |

These macros, fully described in the ASP manual, greatly simplify the data generation.

ARPAL assembles directly from the PDP-5 teletype, creating a symbol table and binary which can be saved on paper tape or (someday) be added to the user's 930 file. CARP and ARPAS punched paper tapes are compatible.

The symbolic op-codes mentioned in this document are those used with ARPAS and ARPAL.

*Document No. 30.40.20
** See ASP Manual, Document No. 30.60.51
*** Document No. 30.60.50

## 4.0 GØ Display Programs

The display program, or list, is a connected, but not necessarily consecutive, list of display commands mixed with display data such as line end-points or characters. The list resides in the PDP-5 memory and commands and data in the list are executed by the display control. In no way, other than the memory sharing, is the display list explicitly connected with PDP-5 programs. In addition to changing the display list, the PDP-5 influences the display only by start and restart pulses following GO and Match interrupts respectively.

### 4.1 Starting Display (GO Button)

When the GØ button is depressed, cell 7 is automatically taken to contain the location of the first display list command. Cell 7 is not read, however, until a display start instruction is executed, IØT 212.

Once started, the display will continue to read and interpret list elements until halted by a Match interrupt or another GØ button; cell 7 need not remain a list element.

### 4.2 Display Commands

Display commands are distinguished from display data by having their high-order bit true. The four basic commands are of two types, change mode or change address. The change address commands allow non-sequential lists and provide the necessary loop control for refreshing the display. The addressing scheme is identical to the PDP-5's. The following are display change address commands:

| Symbolic | Bit Pattern* |
|---|---|
| JMS - Jump Store, similar to PDP-5 JMS | 4XXX |
| JMP - Immediate Unconditional Transfer | 5XXX |

*Bit patterns are given in octal or binary in parentheses. Bits indicated by X are arbitrary.

The effective address of JMP is the location of the next list element, which is read immediately. The display list must close on itself, or loop. A Frame is defined as the loop, the Frame time as the real-time necessary for the display to execute the loop. Frame times come in 17ms quanta since a variable delay is introduced at each occurrence of FRAM command (described below).

The display command JMS is identical to the PDP-5 JMS. The location of the JMS+1, in the display list, is stored in the PDP-5 memory at the effective address of the JMS, and the list continues from there.

## 4.3 Display Modes

Of the above commands, none affect the mode, of which there are four. A mode change command is indicated by bit pattern 7XXX. A mode is left only by entering a new mode. Bit 5 of all four enter mode commands is for enabling (if true) or disabling (if false) the match interrupt for selected portions of the display list.

### 4.3.1 Line Mode

The enter-line-mode command is implied by any of the following:

| Symbolic | Bit Pattern |
|---|---|
| LINE -- solid line | 7(1OX)(X11)X |
| DOT -- single dot at end-point | 7(1OX)(X01)X |
| DASH -- dashed line | 7(1OX)(X10)X |
| JUMP -- blanked line | 7(1OX)(X00)X |

The list elements following the enter-line-mode word are taken sequentially as 10-bit X, Y coordinates for the end-point of a line, of the indicated type, where the start-point is the end-point of the previous line. The binary-coded coordinates are in bits 1-10; the coordinate 0,0 is lower-left corner and 1777,1777 is upper right corner of the display. Coordinate

1777 is 3776 in display data word.

The frame delay is caused by FRAM

FRAM -- frame delay ---- 7(10X)(1XX)X

There should be at least one FRAM per display buffer. The usual case is that the first command of a list will be JUMP FRAM followed by coordinates.

Bit 1 of the Y-coordinate, if true, is an overriding blank, causing a JUMP without changing the mode. Bit 1 of the X-coordinate, if true, causes the line to blink.

The intensity of lines is controllable to three levels by

"normal" -- normal - 7(10X)X(00X)

DIM -- dim - 7(10X)X(01X)

BRI -- bright - 7(10X)X(10X)

OFF -- off - 7(10X)X(11X)

Examples

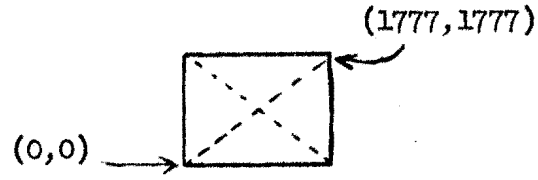
The display list:

JUMP,0,0

LINE,3776,0,3776,3776,0,3776,0,0

BRI DASH,3776,3776,0,3777,3776,0

causes the following where the dashed lines are bright, others are normal intensity:




4.3.2 Symbol Mode

The enter-symbol-mode command is implied by

SYMB -- symbol mode 7(01X)0X

SM -- set primary margin 7(01X)1X

| | | |
|---|---|---|
| RM | -- return primary margin | 7(01X)2X |
| RM1 | -- return primary margin and line feed 16 units | 7(01X)3X |
| RSM | -- return secondary margin | 7(01X)4X |
| SSM | -- set secondary margin | 7(01X)5X |
| RM2 | -- return primary margin and line feed 32 units | 7(01X)6X |
| RM3 | -- return primary margin and line feed 64 units | 7(01X)7X |

In addition to the margin commands, the enter-symbol-mode command will cause the 3-bit static-offset register to be set. The static offset is added to the 2-bit transient offset given in the symbol data word.

| | | |
|---|---|---|
| "normal" | -- no-op | 7(01X)X0 |
| UP1 | -- +16 units up | 7(01X)X1 |
| UP2 | -- +32 units up | 7(01X)X2 |
| UP3 | -- +48 units up | 7(01X)X3 |
| --- | no-op | 7(01X)X4 |
| DWN3 | -- -48 units down | 7(01X)X5 |
| DWN2 | -- -32 units down | 7(01X)X6 |
| DWN1 | -- -16 units down | 7(01X)X7 |

The sequential words following any of the above contain the 6-bit character code in bits 1-6* with 7th bit, case shift, in bit 10. The following modifiers are for intensity and offset (subscripting, superscripting), and blink.

| | | |
|---|---|---|
| BRI | -- bright symbol | 0XX(1XX) |
| BLZ | -- blink symbol | 0XX(XX1) |
| SUP | -- 8 units + offset, small size | 0X(X01)X |

---

*See Appendix B for Display Symbol Codes

SMAL -- no offset, small size OXX(X10)X

SUB -- 8 units - offset, small size OXX(X11)X

"normal" - no offset, normal size OXX(X00)X

The offset is in the Y direction; the small and normal sizes allow 128 or 85 symbols per line, respectively. Each character has a unique "origin" about which it is drawn; most origins are near the lower-left corner.

Each symbol is drawn at a position determined by the line-drawing X, Y registers. The X register is incremented following each symbol according to the size. The first symbol in a string is positioned by drawing a line to the desired coordinate. The X-register can be set to one of two particular previously saved values called the primary and secondary margins as given above. Three different Y-register decrements are possible with RM1, RM2 and RM3. As a special case, the hardware automatically executes an RM2 whenever the X-register overflows.

Symbols can be drawn at 12 usec intervals, but current, hopefully temporary, conditions limit the speed to about 15 usec.

Example:

```
SSM,"- -"
RSM UP1,SUP"∂X"
RSM UP1,SMAL"- -"
RSM UP1,SUB"∂t"
RSM DWN1,SUP"∂y"
RSM DWN1,SMAL"- -"
RSM DWN1,SUB"∂t"
">e"
SUP"jwt"
"+X"
SUP"2π"
```

$$\frac{\frac{\partial x}{\partial t}}{\frac{\partial y}{\partial t}} > e^{jwt} + X^{2\pi}$$

Total = $30_{10}$ list elements

(In ARPAL the quoted symbols are entered via > or < ).

### 4.3.3 Symbol Positioning Mode

Symbols may be positioned at arbitrary coordinates by a combination line and symbol mode. Lines of the types described in Line-Mode (4.3.1) are drawn between symbol positions; symbols in the format described in 4.3.2 are drawn at the ends of the lines.

The enter-mode command is specified by the Enter-Line-mode commands plus SYMB.

Enter Symbol Positioning mode . . . 7(11X)XX

Line type and brightness is specified exactly as in the line mode. The data following the enter-mode command is in triplets: X-coordinate, Y-coordinate, Symbol, in that order. The symbols and coordinates are specified exactly as in the Symbol and Line Modes, including the forced jump, bit 1 of the Y-coordinate.

Example:

The following list will draw lines connecting coordinates (400,600), (420,630), (400,650) and marked by small bright circles at the three coordinates:

..., LINE SYMB, 1000,1400,4,1040,1560,4,1000,1520,4,...

### 4.3.4 Vector Mode

Script-like information is most effectively reproduced with the vector mode, in which one, variable length short line segment can be drawn with each list element.

VEC enter mode 7(00X)XX

The enter-mode command specifies the intensity and blink for all following vectors by:

| | | |
|---|---|---|
| "normal" | normal | 7(00X)X(00X) |
| DIM | dim | 7(00X)X(01X) |

| | | |
|---|---|---|
| BRI | bright | 7(00X)X(10X) |
| OFF | off | 7(00X)X(11X) |
| BLZ | blink | 7(00X)X(XX1) |

Bits 7, 8 in the enter vector mode command are for transient offset as described in 4.3.2 for symbols

| | | |
|---|---|---|
| "normal" | no offset | 7(00X)(X00) |
| SUP | +8 units up | 7(00X)(X01) |
| SMAL | no offset | 7(00X)(X10) |
| SUB | -8 units down | 7(00X)(X11) |

Data for the vector mode consists of one line segment specification per 12-bit word. Bits 1-3 specify $\Delta X$ and 4-6, $\Delta Y$. ($\Delta X$, $\Delta Y$) gives the direction and unit-size. The repeat field, bits 7-10, gives the segment size (unit-size times repeat +1). Bit 11 on causes a blanked segment (BLZ).

$\Delta X/\Delta Y$

000 - no-op

001 - +1 increment integrator

010 - +2 increment integrator

011 - no-op

100 - reset integrator

101 - reset and increment (x)/decrement (y) register by 2 units

110 - -2 decrement integrator

111 - -1 decrement integrator

The vectors are generated by circuitry different from the line-mode; the line-mode registers generally serve to set up the position for the vectors which are accumulated head-on-tail until the integrators are reset. Since the vector integrators are in effect at all times, one could use the vector mode to position the line-mode, i.e., a kind of relative line-mode.

The integrators are automatically reset only at the end of each frame; changing modes does not cause a reset.

Example:

The following lists will generate a line from coordinates (1000,1200) to (1012,1206) and reset both integrators.

... JUMP,1000,1200,VEC,1054, ...

or

... JUMP,1000,1200,VEC,1046,1046,...

## 5.0 RAND Tablet

The tablet is a high resolution (100 lines/inch) input device capable of generating digital 2-dimension positional information as direct computer input.

By bringing the "pen" tip near the tablet surface and pressing lightly until the micro-switch (built into the tip) closes and writing in a natural manner, the user causes 10-bit X and 10-bit Y coordinates to be input to the PDP-5 at a 5 ms rate through program interrupts. Finally, when the pen is lifted, a second tablet interrupt, called Pen-up, acts as an "end-of-stroke" signal to the program. The sequence of X, Y-coordinates read while the pen was down is called a "stroke". The geometry of the stroke may cause activation of so-called "light-buttons" or may provide more complex graphical input.

### 5.1 Tablet Local Display

The X, Y-coordinates are fed to the display deflection circuits directly as well as to the PDP-5. At the end of each frame the coordinates are displayed as a dot. This dot provides the necessary feedback for positioning the pen since the tablet is on a horizontal surface distinct from the vertical surface of the CRT output. The local display requires no software intervention other than the FRAM instruction which defines the end-of-frame.

### 5.2 Tablet "Match"

The tablet may be used in a second distinct manner for pointing at displayed objects. This feature, called Match, uses the pen-coordinates and displayed coordinates as inputs to a comparator to detect and cause interrupts whenever both the X and Y inputs

match within approximately ±1 millimeter on the display. By reading the Display Address Register (DAR) which will contain the location of the next display list element, one can determine what was being displayed at the time of the interrupt (as opposed to the usual positional input). This feature is sometimes referred to as the Light-pen simulation capability of the tablet.

Since the display data fields are variable word-length the following is useful:

| Mode | (DAR-1) at Match Points To | Compared On |
|---|---|---|
| Symbol | symbol | symbol |
| Vector | vector word | left or right byte (cannot distinguish) |
| Line | y-coordinate | line or dot depending on line type |
| Position Symbol | (a) line y-coordinate<br>(b) symbol | same as line mode<br>same as symbol mode |

Moreover, the match interrupts will only occur if

(a) Interrupts are enabled

(b) Match interrupts are armed

(c) The most recent display change-mode command had bit 11 true

(d) No previous matches have occurred in the current frame

(e) Both X and Y coordinates match and the display is unblanked.

Also, matches are obviously disabled during the local display of the pen coordinates. Matches may occur regardless of the state of the pen switch.

At the time of the interrupt the display is temporarily halted until the DAR is read. The instruction which reads the DAR into the PDP-5 AC allows the display to continue.

## 5.3 PDP-5/Tablet Instructions

When the pen-switch is closed, at 5-ms intervals a flag is set which causes a PDP-5 interrupt if "Pen-Down" is armed and enabled. Following each interrupt the coordinates or Display Address Register may be read by:

IØT 111 .... "OR" X-coordinate with $AC_1$ - $AC_{10}$, reset Pen Down flag

IØT 112 .... "OR" Y-coordinate with $AC_1$ - $AC_{10}$, reset Pen Down flag

IØT 114 .... "OR" ones complement of Display Address Register with AC and restart display.

IØT 202 .... Reset Pen Up flag

One may execute the IØT's at any time; however, at 5 ms intervals the tablet registers holding the coordinates are cleared and updated during which time (approximately 20 $\mu$s) the coordinates are invalid (a similar situation exists with the Display Address Register). There is currently no way for the PDP-5 to detect this "invalid" time other than implicitly since the registers are updated just prior to the interrupt. One reason one may want to read the coordinates directly is that they are "random" 10-bit numbers when the pen tip is not near the tablet surface (the distribution of the coordinates or individual bits could easily be determined).

Lifting the Pen (opening the micro-switch) causes the Pen-up interrupt; no other PDP-5 instructions are available (or necessary) for detecting this condition.

The interrupt system transfer-vector locations and arm/disarm bits for the pen interrupts are:

Pen-down .... cell $74_8$ .... bit=$AC_8$

Pen-up .... cell $73_8$ .... bit=$AC_7$

Match .... cell $77_8$ .... bit=$AC_9$

## 5.4 INT Provisions for Tablet

The interrupt processing system INT may be used for buffered input of tablet information by calling BIØ (Block I/O) with:

| Interrupt | File Number (Octal) |
|---|---|
| Pen-down . . . . . | 14 |
| Pen-up . . . . . | 20 |
| Match . . . . . | 10 |

The buffer for Pen-down interrupts is for the coordinate data $X_1$, $Y_1$, $X_2$, $Y_2$, ..., $X_n$, $Y_n$ where the buffer starting address is the location of $X_1$ and the ending address the location of $Y_n$. The Pen-up buffer will contain the Pen-down buffer pointer at the time of the Pen-up interrupt (i.e., the location of the last Y-coordinate of the stroke). This is necessary since the Pen-down buffer may be only partially filled at the time of the Pen-up interrupt. Also, by having separate files for Pen-down and Pen-up it is easy to write user programs to ignore one or the other type interrupt, etc.

The buffer for Match interrupts will contain the location of the cell causing the match as given in part 5.2, i.e., DAR-1 at the time of the match. It is anticipated that random matches may occur (on enabled data) when the pen tip is not close to the tablet surface. Therefore, the program should insist upon several identical matches (say one buffer full) on the same data before accepting the match as valid. Further experimentation is necessary before the exact properties of the match feature are known.

## 6.0 Display Keyboard

The 5-key display keyboard is designed to operate with the left-hand leaving the right-hand free to use the RAND Tablet pen. When any key is depressed or released, the interrupt flag is set, at which time the five key-positions may be read into 5 bits of the PDP-5 accumulator. The simple programming algorithm of OR'ing all characters read between reads with all bits off will allow character input without the user being concerned about relative timing in depressing or releasing the separate switches in inputting a 5-bit code. However, because of the simple logic used it is possible (but very improbable) to get two successive interrupts with code =0*, although this need not cause any problems with the above algorithm.

The instruction which reads the 5-bits into the AC is:

> IØT 222 ... ØR Display Keyboard with $AC_0$ through $AC_4$ (thumb position is $AC_4$), and clear flag.

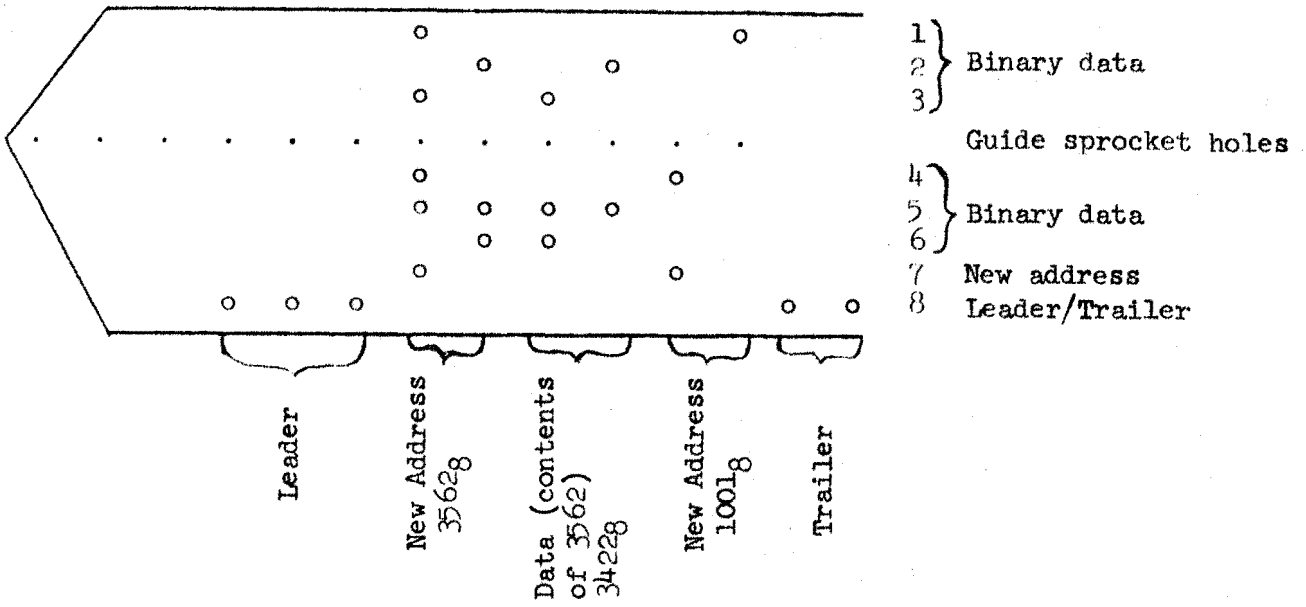
The PDP-5 Interrupt monitor, INT, has provision for block input of the Display Keyboard characters. The file number is $34_8$.

---

*This occurs because code=0 may be gated into the AC, which also resets the flag, just at the instant the last key is lifting which will cause the flag to set. The choice of having too many interrupts was chosen over the alternative of having too few.

# APPENDIX A: RIM and BIN Paper Tape Formats

Digital Equipment Corporation (DEC) standard RIM and BIN format paper tape is as follows:




The distinction between RIM and BIN is that any number of Data words may follow an Address word, each filling successive locations with BIN tape while with RIM tape each Data word must be preceded by an Address word. Also, BIN tape often has a check-sum word at the end of the tape. The Bootstrap-loader switch will read either format, but will ignore any check-sums.

# APPENDIX B: DISPLAY SYMBOL CODES

| Octal Code | Symbol | Octal Code | Symbol |
|---|---|---|---|
| 100 | $\theta$ (lower case theta) | 000 | (blank) |
| 101 | $\lambda$ (lower case lambda) | 001 | ! (exclamation point) |
| 102 | $\mu$ (lower case mu) | 002 | " (right double quotation mark) |
| 103 | $\xi$ (lower case xi) | 003 | # (pound sign) |
| 104 | $\pi$ (lower case pi) | 004 | $ (dollar sign) |
| 105 | $\rho$ (lower case rho) | 005 | % (percent) |
| 106 | $\sigma$ (lower case sigma) | 006 | & (ampersand) |
| 107 | $\tau$ (lower case tau) | 007 | ' (prime) |
| 110 | $\psi$ (lower case psi) | 010 | ( (left parenthesis) |
| 111 | $\partial$ (partial differential) | 011 | ) (right parenthesis) |
| 112 | $\Delta$ (increment) | 012 | * (asterisk) |
| 113 | $\sqcap$ (logical AND) | 013 | + (plus) |
| 114 | $\sqsupset$ (left implication) | 014 | , (comma) |
| 115 | $\sqsubset$ (right implication) | 015 | - (minus) |
| 116 | $\sqcup$ (logical OR) | 016 | . (period) |
| 117 | $\neq$ (not equal) | 017 | / (diagonal) |
| 120 | $\leqslant$ (less or equal) | 020 | 0 (zero) |
| 121 | $\geqslant$ (greater or equal) | 021 | 1 (one) |
| 122 | $\sim$ (similar) | 022 | 2 (two) |
| 123 | $\surd$ (radical sign) | 023 | 3 (three) |
| 124 | $\sum$ (summation) | 024 | 4 (four) |
| 125 | $\int$ (integral) | 025 | 5 (five) |
| 126 | $\prod$ (product) | 026 | 6 (six) |
| 127 | (horizontal) | 027 | 7 (seven) |
| 130 | $\cdot$ (multiply dot) | 030 | 8 (eight) |
| 131 | $\rightarrow$ (right arrow) | 031 | 9 (nine) |
| 132 | $\downarrow$ (down arrow) | 032 | : (colon) |
| 133 | $\triangle$ (centered triangle) | 033 | ; (semicolon) |
| 134 | $\triangleright$ (centered triangle) | 034 | < (less than) |
| 135 | $\triangleleft$ (centered triangle) | 035 | = (equal) |
| 136 | $\triangledown$ (centered triangle) | 036 | > (greater than) |
| 137 | $\times$ (centered cross) | 037 | ? (question mark) |

# APPENDIX B: DISPLAY SYMBOL CODES

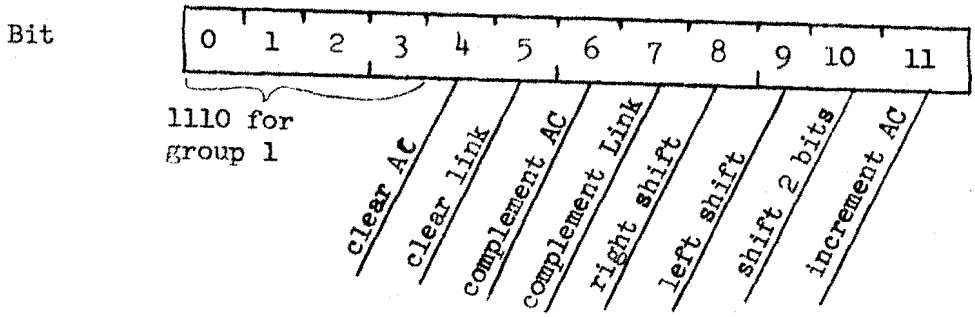
| Octal Code | Symbol | Octal Code | Symbol |
|---|---|---|---|
| 140 | ⊙ (centered circle) | 040 | @ (at) |
| 141 | a (lower case a) | 041 | A (upper case A) |
| 142 | b (lower case bee) | 042 | B (upper case Bee) |
| 143 | c (lower case cee) | 043 | C (upper case Cee) |
| 144 | d (lower case dee) | 044 | D (upper case Dee) |
| 145 | e (lower case e) | 045 | E (upper case E) |
| 146 | f (lower case ef) | 046 | F (upper case Ef) |
| 147 | g (lower case gee) | 047 | G (upper case Gee) |
| 150 | h (lower case aitch) | 050 | H (upper case Aitch) |
| 151 | i (lower case i) | 051 | I (upper case I) |
| 152 | j (lower case jay) | 052 | J (upper case Jay) |
| 153 | k (lower case kay) | 053 | K (upper case Kay) |
| 154 | l (lower case el) | 054 | L (upper case El) |
| 155 | m (lower case em) | 055 | M (upper case Em) |
| 156 | n (lower case en) | 056 | N (upper case En) |
| 157 | ø (lower case ø) | 057 | Ø (upper case Ø) |
| 160 | p (lower case pee) | 060 | P (upper case Pee) |
| 161 | q (lower case cue) | 061 | Q (upper case Cue) |
| 162 | r (lower case ar) | 062 | R (upper case Ar) |
| 163 | s (lower case ess) | 063 | S (upper case Ess) |
| 164 | t (lower case tee) | 064 | T (upper case Tee) |
| 165 | u (lower case u) | 065 | U (upper case U) |
| 166 | v (lower case vee) | 066 | V (upper case Vee) |
| 167 | w (lower case double-u) | 067 | W (upper case Double-u) |
| 170 | x (lower case ex) | 070 | X (upper case Ex) |
| 171 | y (lower case wye) | 071 | Y (upper case Wye) |
| 172 | z (lower case zee) | 072 | Z (upper case Zee) |
| 173 | α (lower case alpha) | 073 | [ (left bracket) |
| 174 | β (lower case beta) | 074 | \| (vertical) |
| 175 | δ (lower case delta) | 075 | ] (right bracket) |
| 176 | ε (lower case epsilon) | 076 | ↑ (up arrow) |
| 177 | ζ (lower case zeta) | 077 | ← (left arrow) |

# APPENDIX C: PDP-5 INSTRUCTIONS

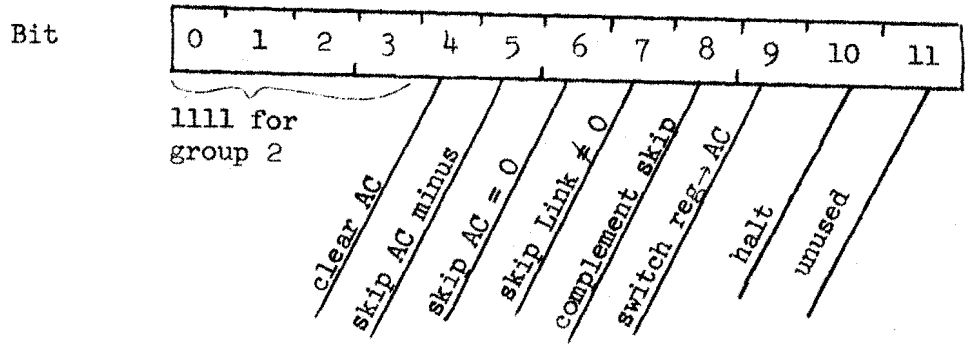
| Mneumonic | Code | Operation | See Footnote |
|---|---|---|---|
| AND | 0000 | logical AND | 8 |
| TAD | 1000 | 2's complement add | 8 |
| ISZ | 2000 | index and skip if zero | 8 |
| DCA | 3000 | deposit and clear AC | 8 |
| JMS | 4000 | jump to subroutine | 8 |
| JMP | 5000 | jump | |
| IØT | 6000 | in-out transfer | |
| ØPR | 7000 | operate | |
| CLA | 7200 | clear AC | 1 |
| CLL | 7100 | clear Link | 1 |
| CLL CLA | 7300 | clear AC and Link | 1 |
| STL | 7120 | set Link = 1 | 1 |
| CMA | 7040 | 1's complement AC | 1 |
| CMA CLA | 7240 | set AC = -1 | 1 |
| CML | 7020 | complement Link | 1 |
| IAC | 7001 | index AC | 1 |
| IAC CLA | 7201 | set AC = +1 | 1 |
| CIA | 7041 | 2's complement AC | 1 |
| RAR | 7010 | ring shift AC and Link one right | |
| RAR CLL | 7110 | logical shift right one | |
| RTR | 7012 | ringshift AC and Link two right | |
| RTR CLL | 7112 | clear Link then RTR | |
| RAL | 7004 | ring shift AC and Link one left | |
| RAL CLL | 7104 | logical shift left one | |
| RTL | 7006 | ring shift AC and Link two left | |
| RTL CLL | 7106 | clear Link then RTL | |
| GLK | 7204 | get Link (Link to AC-11) | |
| SMA | 7500 | skip on AC <0 | 2 |
| SMA SZA | 7540 | skip on AC ≤0 | 2 |
| SZA | 7440 | skip on AC =0 | 2 |
| SNA | 7450 | skip on AC ≠0 | 2 |
| SPA | 7510 | skip on AC ≥0 | 2 |
| SPA SNA | 7550 | skip on AC >0 | 2,3 |
| SNL | 7420 | skip on Link ≠0 | 2 |

| Mneumonic | Code | Operation | See Footnote |
|---|---|---|---|
| SNL SZA | 7460 | Skip on Link ≠0 OR AC=0 | 2 |
| SNL SMA | 7520 | Skip on Link ≠0 OR AC <0 | 2 |
| SNL SMA SZA | 7560 | Skip on Link ≠0 OR AC ≤0 | 2 |
| SZL | 7430 | Skip on Link =0 | 2 |
| SZL SPA | 7530 | Skip on Link =0 and AC ≥0 | 2 |
| SZL SNA | 7470 | Skip on Link =0 and AC ≠0 | 2 |
| SZL SNA SPA | 7570 | Skip on Link =0 and AC >0 | 2 |
| SKP | 7410 | Unconditional skip | 2 |
| OSR | 7404 | AC ∪ switch register → AC | |
| LAS | 7604 | switch register → AC | |
| HLT | 7402 | halt | |
| IØN | 6001 | enable all interrupts | |
| IØF | 6002 | disable all interrupts | |
| ARM | 6101 | arm interrupts | 4,5 |
| DARM | 6102 | disarm interrupts | 4,5 |
| IRC | 6104 | ICW→AC | 5,6 |
| TXR | 6111 | TX ∪ AC→AC, reset Pen Down flag | 5,6 |
| TYR | 6112 | TY ∪ AC→ AC, reset Pen Down flag | 5,6 |
| -- | 6113 | TX ∪ TY ∪ AC→ AC, reset Pen Down flag | 5,6 |
| DAR | 6114 | $\overline{\text{DAR}}$ ∪ AC→ AC, restart display | 5,6 |
| LRS | 6201 | 930-Link receive, skip (LB∪AC→ AC) | 7 |
| TPU | 6202 | Reset Pen-up flag | |
| LTE | 6204 | End 930 transmission | 7 |
| -- | 6205 | 930-Link Receive, skip, end transmission | 7 |
| LTS | 6211 | 930-Link, Transmit, skip (AC→LB, clear AC) | 7 |
| DGØ | 6212 | Reset 930-Link, Pen-up, Pen Down flags and start Display | 7 |
| LTR | 6214 | 930-Link transmit request | 7 |
| -- | 6215 | Trans request, Transmit skip, clear AC | 7 |
| -- | 6216 | Reset Link, enter Transmit request | 7 |
| DKR | 6222 | Display keyboard ∪ AC→ AC, reset flag | |
| KSF | 6031 | skip if keyboard/reader flag=1 | |
| KCC | 6032 | clear AC and keyboard/reader flag | |
| KRS | 6034 | keyboard buffer ∪AC→ AC | |
| KRB | 6036 | KCC and KRS | |
| TSF | 6041 | skip if teleprinter/punch flag =1 | |
| TCF | 6042 | clear teleprinter/punch flag | |
| TPC | 6044 | AC→teleprinter/punch buffer, select and print | |
| TLS | 6046 | TCF and TPC | |

GROUP - 1 Micro Instruction Bit Decoding




GROUP - 2 Micro Instruction Bit Decoding



Footnotes

1. Link commands can be combined with AC commands.
2. CLA can be used with any of these. AC is cleared after skip conditions are tested.
3. Also known as SSP.
4. Particular interrupts are armed/disarmed depending upon particular AC bits.
5. See Document 30.60.15 part 6.0 PDP-5 Interrupt System, Programmer's Guide.
6. TX and TY are RAND Tablet registers.

   DAR is Display Address Register

   ICW is Interrupt Counter Word.
7. See Document 20.50.20, PDP-5/930 Communications Link, Programmer's Guide.
8. AND, TAD, ISZ, DCA are 18 msec instructions. JMS is 24 usec. ADD 6 usec for each indirect reference. All other instructions are 12 usec.

# APPENDIX D

## GO/IDI Display Symbol Table

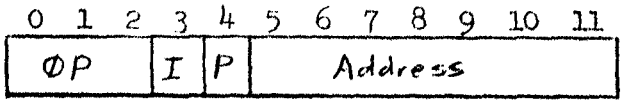
| Mneumonic | Octal Code | Function |
|---|---|---|
| **General Commands** | | |
| JMS | 4000 | subroutine jump* |
| JMP | 5000 | jump |
| HALT | 6000 | halt* |
| **Change Mode Commands** | | |
| VEC | 7000 | vector (increment) mode |
| SYMB | 7200 | symbol mode |
| SM | 7210 | save margin |
| RM | 7220 | return margin |
| RM1 | 7230 | RM, line feed 16 units |
| RSM | 7240 | return secondary margin* |
| SSM | 7250 | save secondary margin* |
| RM2 | 7260 | RM, line feed 32 units* |
| RM3 | 7270 | RM, line feed 64 units* |
| JUMP | 7400 | position |
| DOT | 7410 | position and dot |
| DASH | 7420 | dashed line |
| LINE | 7430 | solid line |
| FRAM | 7440 | frame delay |
| SCH | 7600 | strung symbol mode (IDI)<br>position symbol mode (GO) |
| **Intensity Control** | | |
| BLZ | 0001 | enable blink or set Z bit |
| DIM | 0002 | dim (or case shift for GO display*) |
| BRI | 0004 | bright |
| OFF | 0006 | off |
| **Transient Offset Control** | | |
| SUP | 0010 | small, superscript |
| SMAL | 0020 | small, no offset |
| SUB | 0030 | small, subscript |

| | | |
|---|---|---|
| **Match Enable** | | |
| MAT | 7100B | enable tablet or light pen match |
| **Static Offset**** | | |
| UP3 | 7203 | +24 unit y-offset* |
| UP2 | 7202 | +16 " " |
| UP1 | 7201 | +8 " " |
| DWN1 | 7207 | -8 " " |
| DWN2 | 7206 | -16 " " |
| DWN3 | 7205 | -24 " " |

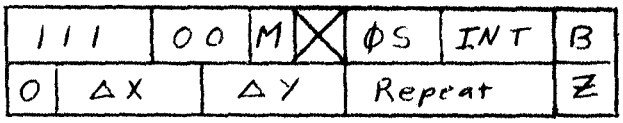
* - currently doesn't exist in IDI display.

** - doesn't exist in ARPAL symbol table, but does in CARP.

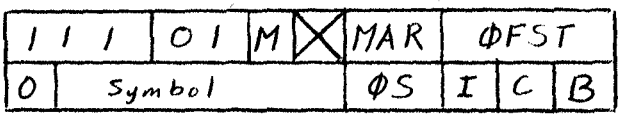
## Bit Assignments -- GO Display



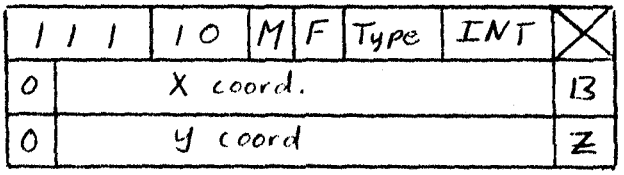

JMS, JMP, HALT (doesn't decode I,P,address)



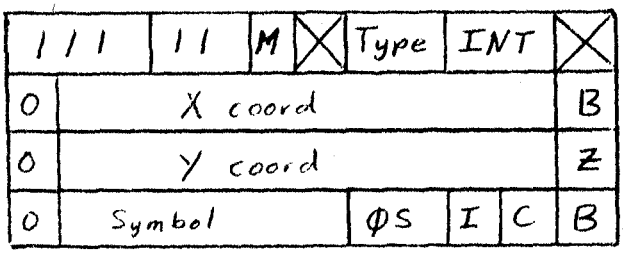

Vector Mode




Symbol Mode




Line Mode




Position Symbol Mode

| | 0 | 1 |
|---|---|---|
| M | disable | enable tablet match (MAT) |
| B | disable | enable blink (BLZ) |
| Z | unblank | blank - don't change INT (BLZ) |
| F | nop | frame sync. (FRAM) |
| C | ASCII | special case |
| I | normal | bright symbols |

OFST = 000 - no-op
001 - +8 units (UP1)
010 - +16 units (UP2)
011 - +24 units (UP3)
100 - no-op
111 - -8 units (DWN1)
110 - -16 units (DWN2)
101 - -24 units (DWN3)

Type = 00 - blank (JUMP)
01 - dot (DOT)
10 - dash (DASH)
11 - solid (LINE)

INT = 00 - normal
01 - dim (DIM)
10 - bright (BRI)
11 - off (OFF)

ØS = 00 - normal, no offset
01 - small, superscript (SUP)
10 - small, subscript (SUB)
11 - small, no offset (SMAL)

ΔX, ΔY = 000 - no-op
001 - +1 increment
010 - +2 increment
011 - no-op
100 - reset integrators
101 - reset and inc(dec)X(y) reg's.
110 - -2 increment
111 - -1 increment

MAR = 000 - no-op
001 - save margin (SM)
010 - return margin (RM)
011 - return margin, line feed 16 units (RM1)
100 - return secondary margin (RSM)
101 - save secondary margin (SSM)
110 - RM, line feed 32 units (RM2)
111 - RM, line feed 64 units (RM3)