

VS60

VS60 VIS DSPLY TST
CZVSDC0

AH-9496C-MC

COPYRIGHT 76-80

FICHE 1 OF 1

JAN 1980

digital

MADE IN USA



IDENTIFICATION

B 1

SEQ 0001

PRODUCT CODE: AC-9494C-MC
PRODUCT NAME: CZVSDCO VS60 VIS DSPLY TST
DATE: MAY 1979
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1976, 1979
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

0.0 TABLE OF CONTENTS

| | |
|------|--------------------------------------|
| 1.0 | ABSTRACT |
| 2.0 | REQUIREMENTS |
| 2.1 | EQUIPMENT |
| 2.2 | STORAGE |
| 3.0 | LOADING PROCEDURE |
| 4.0 | STARTING PROCEDURE |
| 5.0 | TEST PATTERN CONTROL |
| 5.1 | SWITCH REGISTER CONTROL |
| 5.2 | KEYBOARD CHARACTER CONTROL |
| 6.0 | ERROR REPORTING |
| 7.0 | MISCELLANEOUS |
| 7.1 | VS60 BUS/VECTOR ADDRESS MODIFICATION |
| 7.2 | XXDP/APT NOTES |
| 7.3 | POWER FAIL |
| 7.4 | SINGLE VS60 TESTING |
| 8.0 | EXECUTION TIME |
| 9.0 | PROGRAM TEST DESCRIPTIONS |
| 10.0 | LISTING |

1.0 ABSTRACT

***** REV B *****
 DYNAMIC EXTERNAL STOP FRAME (W/27) IS DEPENDANT ON
 HARDWAREECO'S (M7058 #5, M7054 #4 AND VT48 #7).
 ***** REV C *****
 FIXES KEYBOARD SELECTION HACK AND EXPANDS THE
 DYNAMIC OFFSET FRAME.

THE PROGRAM PROVIDES THE OPERATOR WITH TWENTY TWO VISUAL FRAMES
 TO VERIFY THE OPERATION OF THE VS60 DISPLAY SYSTEM.
 NORMALLY EACH FRAME WILL CYCLE FOR ABOUT 5 SECONDS BEFORE
 ADVANCING TO THE NEXT FRAME. EACH VISUAL FRAME CAN BE SELECTED
 VIA SWITCH REGISTER OR KEYBOARD SELECTION.

2.0 REQUIREMENTS

2.1 EQUIPMENT

- A. PDP-11 COMPUTER WITH AT LEAST 12K OF MEMORY.
- B. I/O TERMINAL (I.E. ASR33 TTY OR LK40).
- C. VS-60 DISPLAY SYSTEM.
- D. ADDITIONAL VS-60 DISPLAY CONSOLE (OPTIONAL).

2.2 STORAGE

THE PROGRAM OCCUPIES THE LOWER 8K OF MEMORY BUT
 REQUIRES 12K (BUFFER SPACE) TO RUN.

3.0 LOADING PROCEDURE

NORMAL PROCEDURE FOR LOADING A BINARY PROGRAM INTO MEMORY SHOULD
 BE FOLLOWED.

4.0 STARTING PROCEDURE

LOAD ADDRESS 200 AND START TO INITIALIZE THE SYSTEM
 AND BEGIN TESTING.

5.1 SWITCH REGISTER CONTROL

| SWITCH | FUNCTION |
|--------|-----------------------------------|
| SW14=1 | LOOP ON CURRENT TEST |
| SW09=1 | STOP SUB-PICTURE MOTION |
| SW08=1 | LOOP ON TEST IN SWR<4:0> |
| SW07=1 | ENABLE KEYBOARD CONTROL (REF.5.2) |

5.2 KEYBOARD CONTROL

SEQ 0004

STARTING THE TEST WITH SR7=1 WILL ENABLE KEYBOARD CONTROL. KEYBOARD CONTROL IS AN AUXILIARY METHOD OF SELECTING THE TEST FRAME, LOOP ON A TEST FRAME, OR STOP-START FRAME MOTION. THE SWITCH REGISTER BITS OVERRIDE THE KEYBOARD CONTROL. THE DIRECTORY FRAME PROVIDES THE OPERATOR WITH THE KEYBOARD LETTER AND SWITCH REGISTER VALUE FOR EACH TEST PATTERN. TO SELECT A TEST PATTERN, SIMPLY DEPRESS THE TEST LETTER ON THE CONSOLE KEYBOARD. DEPRESS THE 'RUB-OUT' KEY TO LOOP ON THE CURRENT TEST PATTERN. DEPRESS THE 'CR' KEY TO STOP MOTION. UNDEFINED TEST LETTERS WILL DISPLAY THE DIRECTORY FRAME. ALL OTHERS WILL HAVE NO EFFECT OTHER THAN TO RESUME PICTURE MOTION.

6.0 ERROR REPORTING

THE PROGRAM ONLY DISPLAYS VISUAL ERRORS AND DOES NOT REPORT ANY LOGIC ERRORS.

7.0 MISCELLANEOUS

7.1 VS60 BUS/VECTOR/PRIORITY ADDRESS MODIFICATION

MODIFY LOCATION 1242 (\$VECT1) IF BASE VECTOR ADDRESS IS NOT 100320.
MODIFY LOCATION 1246 (\$BASE) IF BASE BUS ADDRESS IS NOT 172000.

NOTE: A RESTART IS REQUIRED AFTER THE ABOVE ADDRESS MODIFICATION.

7.2 XXDP/APT NOTES

THE VISUAL TEST IS CHAINABLE UNDER XXDP IF 16K OR GREATER MEMORY IS AVAILABLE. THE VISUAL TEST INCLUDES THE 'APT' SOFTWARE HOOKS, HOWEVER THEY HAVE NOT BEEN TESTED.

7.3 POWER FAIL

A POWER FAILURE WILL CAUSE THE PROGRAM TO BE RESTARTED.

7.4 SINGLE VS60 TESTING

THE VISUAL TEST DOES NOT TEST MULTIPLE VS60'S.
THE VISUAL TEST WILL UTILIZE THE SECOND CONSOLE IF CONNECTED.
THE 'A' AND 'U' FRAMES ARE USED TO VERIFY PROPER OPERATION BETWEEN THE TWO DISPLAY CONSOLES.

8.0 EXECUTION TIME

SEQ 0005

EXECUTION TIME IS APPROX. FOUR MIN. AN 'END OF PASS' IS INDICATED BY A RETURN TO THE DIRECTORY FRAME. NO 'END OF PASS' MESSAGE IS TYPED.

9.0 PROGRAM TEST DESCRIPTIONSA = 01 Directory Frame

The sub-picture supplies the operator with a List of the Different Visual frames for his inspection.

This frame also includes a list of switch register values and keyboard control letters to select the visual frames. When a non-valid switch register value or keyboard key has been selected, the directory frame will be displayed. IF THE SECOND CONSOLE IS CONNECTED, THE OPERATOR SHOULD VERIFY THE 'THIS IS CONSOLE 0' MESSAGE ON CONSOLE #0 AND THE 'THIS IS CONSOLE 1' MESSAGE ON CONSOLE #1.

The frame is displayed by doing the following:

1. Point to x = 0 y = 1500
2. Enable console 1 intensity
3. Enter 'character' mode and display inline text.
4. Display 'STOP'
5. Display 'JUMP ABSOLUTE' to the start of the frame.

B = 02 Astigmatism and Settling Time Frame

The frame will display points at individual bits at each x and y position register.

A floating one pattern used on each register followed by an accumulation pattern.

Bit 9 of x pos. Bit 9 of y pos.
 Bit 8 of x pos. Bit 9 of y pos.
 etc. etc.

Bits 9 and 8 of x pos. Bits 9 and 8 of y pos.
 etc. etc.

C = 03 Short Term Drift Frame

The frame will display five points. The points will be displayed in each corner and the center of the screen. Each point actually consists of four 'Display Point' instructions.

The point is generated by:

1. Positioning the x and y DAC at a coordinate.
2. Intensifying the coordinate ONCE.
3. Do not intensify the point again for five (5) milliseconds.
4. Repeat 2 and 3 three more times.
5. If all the coordinates have not been displayed, update the coordinate and rePEAT 1 thru 4.

The C.P.U. cycle time is a factor in the 5 msec. delay routine. The current delay value (location 'DELAY') is valid for a PDP-11/40 CPU type.

D = 04 Minor Axis Gain, Offset and Phase Frame

The frame consists of three square boxes with diagonal bisecting lines. The largest box encompasses the whole main screen viewing area. The second box, whose size is 100., is displayed in the right center area. The third box, whose size is 10., is below the second box. The boxes are drawn counter clockwise from the lower left corner. Upon completion the procedure is reversed and drawn clockwise from the lower left corner. When drawing the clockwise box the 'Negative' polarity bit is set to enable adjustment of the 'Offset' pot. Each box, upon completion, is segmented by a diagonal line from lower left to upper right and lower right to upper left corner. The frame also draws the same type box in the 'menu' area. Because the 'menu' is narrower than the high, the result is an rectangle in the menu area. In the lower center area, a series of four vectors 200 units long, are drawn from a common point. In the left center quadrant, ten vectors are drawn using 'SHORT VECTOR' mode. Each of the vectors have a

H 1

length of eight units. After drawing the vector a 'RELATIVE POINT' is displayed two units away from the end of the vector. The 'Y' coordinate is updated by two units and the 'SHORT VECTOR' and 'RELATIVE POINT' sequence is repeated. The visual result is a vertical 'DOT-DASH' line. Included in the left quadrant is the Intensity Delay sub-picture. Eight vectors are drawn away from a 'COMMON POINT' offset by ONE unit. The result will appear to be a square formed by the starting points of the vectors. Each vector has a length of 40 units.

| VECTOR # | ORIGINATES AT X | ORIGINATES AT Y |
|----------|-----------------|-----------------|
| 1 | 0354 | 1003 |
| 2 | 0354 | 1004 |
| 3 | 0353 | 1004 |
| 4 | 0352 | 1004 |
| 5 | 0352 | 1003 |
| 6 | 0352 | 1002 |
| 7 | 0353 | 1002 |
| 8 | 0354 | 1002 |

E = 05 Major Axis Offset and Vector Start Frame

The frame includes the minor axis gain frame plus two additional patterns. The first is used to adjust the vector starting point. The second pattern to adjust the major axis offset. The first pattern is drawn, in the upper quadrant, with the following manNEr:

| Vector # | Direction |
|----------|--|
| 1 | Positive Vertical Reference Vector. |
| 2 | Positive Horizontal Vector starting on VECTOR #1 |
| 3 | Positive Horizontal Vector starting 1 unit RIGHT OF VECTOR #1 |
| 4 | Negative Horizontal Vector starting on VECTOR #1 |
| 5 | Negative Horizontal Vector starting 1 unit LEFT OF VECTOR #1 |
| 6 | Positive Horizontal Vector start at the BOTTOM OF VECTOR #1 |
| 7 | Negative Vertical Vector starting at the bottom OF VECTOR #1 |
| 8 | Negative Horizontal Vector starting at the BOTTOM OF VECTOR #1 |

The second pattern draws, from a common point (x=1000, y=400), four pairs of vectors. The first of each pair is drawn with the 'y' axis being the major value. With the second using the 'x' as the major. THE THIRD PATTERN CONSISTS OF 10 PAIRS OF SHORT LENGTH VECTORS AND RELATIVE POINT'S DRAWN IN THE LEFT CENTER AREA. AN 8 UNIT SHORT VERTICAL VECTOR IS DRAWN FOLLOWED BY A ONE UNIT RELATIVE POINT.

The pattern appears to be a series of horizontal lines being intersected by a diagonal line from upper left to lower right.

The picture is drawn by:

1. Draw an outer reference box
2. Starting from maximum, draw an increasing negative length vector from an increasing 'y' origin.
3. Starting from minimum x, draw a decreasing length vector from an increasing 'y' origin.
4. Starting in the upper left edge, intensify a point at the intersection of #2 to #3 vector.
5. Starting in the upper left edge, intensify a decending vector that is over #4.
6. From center screen, using 'BASIC' Vectors draw two intersecting 'x' and 'y' lines.

G = 07 Pincushion Frame

Using the 'LONG' Vector instruction, display a 'CROSS HATCH' visual pattern. The frame can be used to detect distortion in Vectors. From a distance of three feet, all vectors should appear straight with no vector curvature.

H = 10 Octagons AND CIRCLES Frame

The purpose of the frame is to verify the endpoint matching of vectors. FIVE octagons are drawn from the center of the screen. The outer most octagon is drawn by using the 'ABSOLUTE VECTOR' instruction from the point $x = 530$ $y = 10$.

| Vector # | from | x-y | to | x-y |
|----------|------|-----------|----|-----------|
| Vector 1 | from | 530-10 | to | 1250-10 |
| Vector 2 | from | 1250-10 | to | 1770-530 |
| Vector 3 | from | 1770-530 | to | 1770-1250 |
| Vector 4 | from | 1770-1250 | to | 1250-1770 |
| Vector 5 | from | 1250-1770 | to | 530-1770 |
| Vector 6 | from | 530-1770 | to | 10-1250 |
| Vector 7 | from | 10-1250 | to | 10-530 |
| Vector 8 | from | 10-530 | to | 530-10 |

The FOUR concentric octagons are drawn by using the 'LONG VECTOR' display instruction. The sizes are 377, 177, 77, 7 respectively. Two more octagons with a size of 17 units are drawn at $x = 300$ $y = 1000$ and $x = 1500$ $y = 1000$. These two are drawn using the 'BASIC SHORT' vector display instruction. THREE CONCENTRIC CIRCLES ARE DRAWN USING ABSOLUTE VECTOR MODE. EACH CIRCLE CONSISTS OF 45 ABSOLUTE VECTORS. THE THREE CIRCLES HAVE A RADIUS OF 64., 128., AND 256. RESPECTIVELY.

I = 11 Scissoring and Vector Scaling Frame

The frame starts out by displaying a reference box around edge of the screen.

A VECTOR IS DRAWN FROM AN 'ON-SCREEN' POSITION TO AN 'OFF-SCREEN' POSITION. Another vector is drawn from the end of the previous vector back into the viewing area. This is repeated four times on each screen edge. The vectors should all terminate WITH NO bending or distortion. After all edges have been intersected, draw a large diamond that intersects each edge. The diamond and the vectors crossing the edges are the standard vector length. To verify that vector scale operates properly, draw a square in the center of the screen. By changing the value of the 'Vector Scale' register the box should increase in size. the vector scale is changed with the resulting picture being sixteen scaled boxes in the center of the screen.

J = 12 X and Y Dynamic Offset Frame

IN THIS FRAME, A 1000 UNIT BOX IS DRAWN IN THE CENTER OF THE SCREEN. USING THE DISPLAY 'OFFSET' INSTRUCTION, THE BOX IS MADE TO SLIDE ACROSS THE SCREEN TO THE RIGHT, THEN TO THE LEFT, TOP, AND BOTTOM EDGES.

AFTER THESE FOUR MOTIONS THE BOX IS RETURNED TO CENTER AND MOVED DIAGONNALLY TO THE UPPER RIGHT, AND LOWER LEFT BY SETTING THE OFFSET REGISTERS DIRECTLY FROM THE CPU (EXTERNAL TO THE DISPLAY FILE).

THE RANGE OF OFFSET USED IS 0 TO 1400 (POS AND NEG) IN BOTH CASES.

K = 13 Character Scale Frame

The frame function is to verify that character scale does change the size. To verify character scale, six characters, (the letters A, B, F, O, T and X) are displayed. each character starts with the largest to the smallest size on a common base line. A horizontal reference is drawn along the base of the characters.

L = 14 Character Quality and Character Rotate Frame

In this frame the message "The quick brown fox jumped over the lazy dogs" is displayed over the entire screen. By displaying the full screen of characters, the quality and distortion of the characters may be checked. Also included in the frame are rotated CHARACTERS. The rotated characters are displayed in the menu area.

M = 15 Character Set, Superscript, Subscript and Italic Frame

The frame displays all the displayable characters, special, italic, superscript and subscript. The first line consists of upper case letter (codes 100-137) and italic uppercase letters. The second line contains lower case letters (codes 140-177) and italic lower case letters. The third line contains numbers and punctuation (codes 40-77) and italic numbers and punctuation. The fourth line contains the special characters and italic special characters. These four lines are repeated in the lower half of the screen. Near the center of the screen a horizontal reference line is displayed.

The largest character scale is enabled and the letter 'E' is displayed. This should appear on the base reference line. The code "super-script on" is enabled, followed by another 'E'.

The procedure is repeated three times with the result being four letter's of 'E' with each having a reduced size and an ascending y position. To verify the "superscript-off" function, the code "super-script off" followed by an ASCII 'E' is sent. The procedure is repeated three times with the character increasing in size and decending in the y position.

The last 'E' should be on the base reference line. The same procedure is repeated using the "subscript-on" and "subscript-off" codes except the characters should first decend with reducing in size followed by ascending and increasing in character size.

N = 16 Sync Speed and Character Terminate Frame

The patterns serves two FUNCTIONS. The first is to test character terminate. A diamond is displayed in the center of the screen with a message about the 'SYNC' speed. The message is terminated by the value of '177' (a full dot matrix character).

The code #177 is loaded into the character terminate register and character terminate (character string escape) function is enabled.

The diamond is displayed using the 'BASIC Vector' instruction. The message is displayed by entering 'character' mode and doing a 'display JSR' to the ASCII string. The text should be displayed and a 'display POP and RESTORE' should occur after the code #177 is displayed. If 'character terminate' fails to cause a 'POP', a DIFFERENT message will be displayed reporting THE FACT.

The second purpose is to verify a visual change in the picture intensity when using NO SYNC, 40 cps sync and 30 cps sync.

The displayed message will indicate the different sync speeds.

When no sync is enabled the frame will appear bright and will have no flicker. When a sync speed of 40 is enabled, the frame will become dim. Upon selection of a sync speed of 30, the frame should appear to flicker. In each case, the frame appears different for each sync speed.

O = 17 Dash Lines and Blink Frame

This is a frame dedicated to the different line types and the ability to generate a blinking element. The type of line followed by two vectors of the same line type are displayed. The first is without blink enabled and the second is displayed with blink enabled. Visually the type of line is displayed followed by a non blinking line of the type followed by a blinking line of the type. This frame also used a 'Display jump relative to loop' on the frame.

P = 20 Vector Length (Spray) Frame

The frame consists of 'ABSOLUTE' vectors drawn from point 00 to another x,y point and a return vector to point 0,0. The first vector is drawn from point 0,0 to the maximum x and a y position of 1. Then a INVISIBLE vector to 0,0 is drawn. The third vector is drawn from point 0,0 to the maximum x and a y position of 3. This is repeated until the maximum y position has been displayed(45 DEG.). At that point the sequence is reversed IN that the x is the adjusted end point. The vector is drawn from point 0,0 to a value of x and the maximum value of y. A reference x and y vector is drawn at the right and top edge of the main screen. Each vector should terminate on the reference line. Even spacing should exist between the end of each vector. EVERY OTHER VECTOR WILL BE DISPLAYED.

Q = 21 Horizontal Phosphor Frame

In this frame, a reference box around the main screen perimeter is displayed. A band of intensified vectors are drawn to enable the operator to inspect phosphor surface. The band uses the 'BASIC Vector' instruction by going the full value of y (path 2), delta x of 2 units (path 0), negative full value of y (path 6) and a delta x of 2 units. This is repeated 50 times. The origin point of the band is updated via the 'Point' instruction. The number of times the band is displayed before moving to the next position is controlled by the number loaded into the 'TEMPA'.

R = 22 Vertical Phosphor Frame

In this frame, a reference box around the main screen and menu perimeter is displayed. A band of intensified vectors are drawn thru the main screen and the menu screen to enable the operator to inspect the phosphor surface. The band uses 'BASIC Vector' instruction by going the full value of x (path 0), delta y of 2 units (path 2), negative full value of x (path 4), and a delta y of 2 units. This is repeated 50 times. THE PROCESS IS THEN REPEATED AGAIN IN THE MENU AREA EXCEPT USING THE MAXIMUM X MENU LENGTH (177).

The origin point of the band is updated via the 'Point' instruction. The number of times the band is displayed before moving to the next position is controlled by the number loaded into the 'TEMPA'.

S = 23 Short Vector and Relative Point Frame

SEQ 0013

With this frame the operator can verify the correct selection of Relative point and short vectors. Four octagons are drawn in the four quadrants of the screen. Each octagon consists of an outer octagon drawn using the 'short vector' instruction. Within each major octagon should be eight points at the intersecting vectors OF THE MAJOR OCTAGON. The 'Relative point' instruction is used to display these points. A THIRD OCTAGON IS DISPLAYED USING THE 'SHORT VECTOR' INSTRUCTION.

T = 24 GRAPHPLOT INCREMENT REGISTER TEST USING GRAPHPLOT X AND GRAPHPLOT Y

THE GRAPHPLOT INCREMENT REGISTER IS VERIFIED WITH A 'SINE WAVE' PATTERN. TWO CYCLES OF A SINE WAVE ARE DISPLAYED IN GRAPHPLOT Y AND GRAPHPLOT Y MODES. THE AMOUNT OF INCREMENT BETWEEN POINTS IS A FUNCTION OF THE GRAPHPLOT INCREMENT REGISTER. AT THE END OF THE DISPLAY FILE IS A 'DISPLAY STOP'. UPON DETECTING THE DSTOP, A COUNTER IS DECREMENTED. UPON EXHAUSTION OF THE COUNTER, THE GRAPHPLOT INCREMENT REGISTER IS CHANGED. THE RESULT IS THE SINE WAVES WILL APPEAR TO EXPAND TO THE RIGHT, FOR GRAPHPLOT Y, AND TO THE TOP, FOR GRAPHPLOT X. ONLY THE LOWER THREE BITS OF THE INCREMENT REGISTER ARE VERIFIED WITH THIS PATTERN.

U = 25 Intensity Level and Lightpen Frame

The frame provides the operator with a method to visually check the eight different intensity levels. Points, Vectors and Characters are drawn using the different intensity levels. The frame also includes handling of 'Light-pen' flags and 'Light-Pen switches'. An octagon is displayed in the upper right corner. Inside the octagon contain the X and Y axis values for the last 'Light-Pen Hit'. The state of the 'Light-Pen switch' is also displayed within the octagon. In the lower right area a matrix of dots is used for a static test of the 'Light-Pen field of View'. The intensified dots are spaced four units apart. When the dots are detected by the 'Light-Pen', the dot which a hit has occurred on will not be displayed. Below the dot matrix is an octal readout reporting the hit count total. The center of the frame is bisected by a Horizontal Reference Line (Y=700). Nine vertical reference lines are drawn at 200 unit increments. The vertical lines are drawn below the Horizontal Reference Line are used to verify correct 'X' pen hit position. The lower left section contains vertical spacing test. Three parallel vectors are drawn with decreasing vertical spacing between the lines. The lower center area consists of a Variable Line Length Test. Twenty horizontal lines with increasing X length are drawn from a common X position. Both sections are used to test light pen selectivity. IF THE SECOND CONSOLE IS CONNECTED, VERIFY INDEPENDANT OPERATION OF THE X/Y AND PEN SWITCH READOUT FOR EACH CONSOLE. THE 'FIELD OF VIEW' AND THE 'HIT-COUNT' ARE THE ONLY DEPENDANT ELEMENTS.

***** TYPE <V> OR SWR = 426 TO RUN THIS FRAME *****

THE FRAME PROVIDES A KEYBOARD TO VS60 SCREEN CHARACTER LOOP TO VERIFY PROPER OPERATION OF THE CONSOLE KEYBOARD. A MAXIMUM OF 1024 CHARACTERS CAN BE DISPLAYED BY THIS LOOP. THE OPERATOR MAY ESCAPE THE LOOP, BY DEPRESSING THE "CTRL" AND "C" KEYS, TO RETURN TO THE DIRECTORY FRAME. UPON DETECTION OF A KEYBOARD CHARACTER, THE CHARACTER'S OCTAL VALUE AND THE CHARACTER ARE DISPLAYED ON THE SCREEN.

THE "SHIFT-OUT" CODE CAN BE ENTERED BY THE OPERATOR, HOWEVER THE PROGRAM WILL NOT USE ANY KEYBOARD CODES GREATER THAN 37 OCTAL. UPON ENTERING A "SHIFT-OUT" MODE, THE CHARACTER DISPLAYED FROM THE CURRENT CHARACTER POSITION TO THE END OF THE LINE WILL APPEAR TO BE AN UPSIDE DOWN "Y" CHARACTER. IN THE "SHIFT-OUT" MODE, THE CHARACTER DISPLAYED HAS THE VALUE OF ZERO.

W = 27 DYNAMIC EXTERNAL STOP FRAME

***** THIS FRAME VERIFIES VS60 ECO'S VT48 #7, *****
***** M7054 #4 AND M7058 #5 HAVE BEEN INSTALLED *****

THIS FRAME VERIFIES PROPER OPERATION OF THE EXTERNAL DISPLAY STOP LOGIC. A FRAME CONTAINING MOST OF THE VS60 INSTRUCTIONS IS DISPLAYED. WHILE THE VS60 IS DISPLAYING THE FRAME, THE -11 CPU IS RANDOMLY GENERATION A EXTERNAL DISPLAY STOP SIGNAL (EDSS) TO THE VS60. AFTER AN "EDSS" HAS BEEN SENT, THE -11 WILL VERIFY THE DISPLAY PROGRAM COUNTER REGISTER TO BE WITHIN AN EXPECTED RANGE. THE GENERATION OF AN "EDSS" SHOULD CAUSE AN EXTERNAL STOP INTERRUPT. UPON DETECTING AN "EDSS" INTERRUPT, A COUNTER IS DECREMENTED. IF THE COUNTER DOES NOT GO TO 0, THE PROGRAM WILL ISSUE A "RESUME" TO THE VS60. IF THE COUNTER BECOMES 0, THE PROGRAM WILL GO TO THE "END OF PASS" AND RESTART THE PROGRAM. SEVEN DIFFERENT ERROR CONDITIONS WILL BE VISUALLY REPORTED WITH THIS SUB-TEST:

| <u>ERROR #</u> | <u>REASON</u> |
|----------------|---|
| 0 | NO EXTERNAL STOP INTERRUPT |
| 1 | UNEXPECTED INTERRUPT TO VECTOR +4 |
| 2 | UNEXPECTED INTERRUPT TO VECTOR +10 |
| 3 | UNEXPECTED INTERRUPT TO VECTOR +14 |
| 4 | D.P.C. OUT OF RANGE (TOO LOW) |
| 5 | D.P.C. OUT OF RANGE (TOO HIGH) |
| 6 | EXTERNAL STOP INTERRUPT BUT NO EXTERNAL |

10.0 PROGRAM LISTING


```
(1) ;*'SWITCH REGISTER' SWITCH DEFINITIONS
(1) 100000 SW15= 100000
(1) 040000 SW14= 40000
(1) 020000 SW13= 20000
(1) 010000 SW12= 10000
(1) 004000 SW11= 4000
(1) 002000 SW10= 2000
(1) 001000 SW09= 1000
(1) 000400 SW08= 400
(1) 000200 SW07= 200
(1) 000100 SW06= 100
(1) 000040 SW05= 40
(1) 000020 SW04= 20
(1) 000010 SW03= 10
(1) 000004 SW02= 4
(1) 000002 SW01= 2
(1) 000001 SW00= 1
(1) .EQUIV SW09,SW9
(1) .EQUIV SW08,SW8
(1) .EQUIV SW07,SW7
(1) .EQUIV SW06,SW6
(1) .EQUIV SW05,SW5
(1) .EQUIV SW04,SW4
(1) .EQUIV SW03,SW3
(1) .EQUIV SW02,SW2
(1) .EQUIV SW01,SW1
(1) .EQUIV SW00,SW0
```

```
(1) ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1) 100000 BIT15= 100000
(1) 040000 BIT14= 40000
(1) 020000 BIT13= 20000
(1) 010000 BIT12= 10000
(1) 004000 BIT11= 4000
(1) 002000 BIT10= 2000
(1) 001000 BIT09= 1000
(1) 000400 BIT08= 400
(1) 000200 BIT07= 200
(1) 000100 BIT06= 100
(1) 000040 BIT05= 40
(1) 000020 BIT04= 20
(1) 000010 BIT03= 10
(1) 000004 BIT02= 4
(1) 000002 BIT01= 2
(1) 000001 BIT00= 1
(1) .EQUIV BIT09,BIT9
(1) .EQUIV BIT08,BIT8
(1) .EQUIV BIT07,BIT7
(1) .EQUIV BIT06,BIT6
(1) .EQUIV BIT05,BIT5
(1) .EQUIV BIT04,BIT4
(1) .EQUIV BIT03,BIT3
(1) .EQUIV BIT02,BIT2
(1) .EQUIV BIT01,BIT1
(1) .EQUIV BIT00,BIT0
```

```
(1)      ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)      ERRVEC= 4      ;;TIME OUT AND OTHER ERRORS
(1)      RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
(1)      TBITVEC=14     ;;'T' BIT
(1)      TRTVEC= 14     ;;TRACE TRAP
(1)      BPTVEC= 14     ;;BREAKPOINT TRAP (BPT)
(1)      IOTVEC= 20     ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)      PWRVEC= 24     ;;POWER FAIL
(1)      EMTVEC= 30     ;;EMULATOR TRAP (EMT) **ERROR**
(1)      TRAPVEC=34     ;;'TRAP' TRAP
(1)      TKVEC= 60      ;;TTY KEYBOARD VECTOR
(1)      TPVEC= 64      ;;TTY PRINTER VECTOR
(1)      PIRQVEC=240    ;;PROGRAM INTERRUPT REQUEST VECTOR
19
20      ABASE= 172000   ; DISPLAY PC ADDRESS.
21      AVECT1= 100320 ; 1ST OF 4 DISPLAY VECTORS.
22      APRIOR= 200
23
24      .SBTTL OPERATIONAL SWITCH SETTINGS
(1)      :*
(1)      :*      SWITCH      USE
(1)      :*      -----
(1)      :*      14      LOOP ON TEST
(1)      :*      9      STOP SUB-PICTURE MOTION
(1)      :*      8      LOOP ON TEST IN SWR<7:0>
25      .SBTTL TRAP CATCHER
(1)      .=0
(1)      000000
(1)      ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
(1)      ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1)      ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1)      .=174
(1)      000174 000000  DISPREG: .WORD 0      ;;SOFTWARE DISPLAY REGISTER
(1)      000176 000000  SWREG:   .WORD 0      ;;SOFTWARE SWITCH REGISTER
(1)      .SBTTL STARTING ADDRESS(ES)
(1)      000200 000137 001336 JMP @#BEGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
```

27
(1)
(2)
(1)
(1) 000046 000204
(1) 000046 000046
(1) 000052 006664
(1) 000052 000052
(1) 000052 000000
(1) 000052 000204
(1) 000052 001000
28
29
(1)
(2)
(1)
(2)
(1) 001000
(1) 000024 000024
(1) 000024 000200
(1) 000044 000044
(1) 000044 001000
(1) 001000
(1) 001000 000000
(1) 001002 001172
(1) 001004 000020
(1) 001006 000300
(1) 001010 000000
(1) 001012 000032

```
.SBTTL ACT11 HOOKS
:*****
:HOOKS REQUIRED BY ACT11
  $SVPC=.          ;SAVE PC
  .=46
  $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
  .=52
  .WORD 0         ;;2)SET LOC.52 TO ZERO
  .=$SVPC        ;; RESTORE PC
  .=1000

.SBTTL APT PARAMETER BLOCK
:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
  .$X=.          ;;SAVE CURRENT LOCATION
  .=24          ;;SET POWER FAIL TO POINT TO START OF PROGRAM
  200          ;;FOR APT START UP
  .=44          ;;POINT TO APT INDIRECT ADDRESS PNTR.
  $APTHDR      ;;POINT TO APT HEADER BLOCK
  .=$X         ;;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.

$APTHD:
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM:  .WORD 20    ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 300   ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 0     ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```



```
(2) 001206 000000 $MSGAD: .WORD   AMMSGAD  ;;MESSAGE ADDRESS
(2) 001210 000000 $MSGLG: .WORD   AMMSGLG  ;;MESSAGE LENGTH
(2) 001212          $ETABLE:          ;;APT ENVIRONMENT TABLE
(2) 001212          $ENV: .BYTE   AENV      ;;ENVIRONMENT BYTE
(2) 001213          $ENVM: .BYTE   AENVM     ;;ENVIRONMENT MODE BITS
(2) 001214 000000 $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
(2) 001216 000000 $USWR: .WORD   AUSWR    ;;USER SWITCHES
(2) 001220 000000 $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
(2)          : *          BITS 15-11=CPU TYPE
(2)          : *          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2)          : *          11/70=06,PDQ=07,Q=10
(2)          : *          BIT 10=REAL TIME CLOCK
(2)          : *          BIT 9=FLOATING POINT PROCESSOR
(2)          : *          BIT 8=MEMORY MANAGEMENT
(2) 001222          $MAMS1: .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S. BYTE
(2) 001223          $MTYP1: .BYTE   AMTYP1   ;;MEM. TYPE,BLK#1
(2)          : *          MEM.TYPE BYTE -- (HIGH BYTE)
(2)          : *          900 NSEC CORE=001
(2)          : *          300 NSEC BIPOLAR=002
(2)          : *          500 NSEC MOS=003
(2) 001224 000000 $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
(2)          : *          MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
(2) 001226          $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
(2) 001227          $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
(2) 001230 000000 $MADR2: .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
(2) 001232          $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
(2) 001233          $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
(2) 001234 000000 $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
(2) 001236          $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
(2) 001237          $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
(2) 001240 000000 $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
(2) 001242 100320 $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2) 001244 000000 $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2) 001246 172000 $BASE: .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2) 001250 000000 $DEV: .WORD   ADEV      ;;DEVICE MAP
(2) 001252 000000 $CDW1: .WORD   ACDW1    ;;CONTROLLER DESCRIPTION WORD#1
(2) 001254 000000 $CDW2: .WORD   ACDW2    ;;CONTROLLER DESCRIPTION WORD#2
(2) 001256          $ETEND:
(2)          .MEXIT
```

```
(1) .SBTTL ERROR POINTER TABLE
(1)
(1) ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1) ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1) ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1) ;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
(1) ;*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1) ;* EM ;:POINTS TO THE ERROR MESSAGE
(1) ;* DH ;:POINTS TO THE DATA HEADER
(1) ;* DT ;:POINTS TO THE DATA
(1) ;* DF ;:POINTS TO THE DATA FORMAT
(1)
(1) 001256 $ERRTB:
(1) 31 ;NO ERRORS ARE TYPED OUT
(1) 32
(1) 34
(1) 35 .SBTTL OPERATOR VARIABLE LOCATIONS
(1) 36
(1) 37 001256 000020 DELAY: BIT4 ;CPU DELAY FACTOR (5MS FOR 11/40 CPU)
(1) 38 001260 000060 TKBVCT: 60 ;CONSOLE KEYBOARD VECTOR
(1) 39 001262 000062 TKBVT1: 62
(1) 40
(1) 41 .SBTTL VS-60 ADDRESSES AND INTERRUPT VECTORS
(1) 42
(1) 43 001264 172000 DPC: 172000 ;DISPLAY PROGRAM COUNTER
(1) 44 001266 172002 DSR: 172002 ;DISPLAY STATUS REGISTER
(1) 45 001270 172004 XPOS: 172004 ;DISPLAY X AXIS REGISTER
(1) 46 001272 172006 YPOS: 172006 ;DISPLAY Y AXIS REGISTER
(1) 47 001274 172010 DSREL: 172010 ;DISPLAY RELOCATE REGISTER
(1) 48 001276 172012 DSR1: 172012 ;DISPLAY EXT. STOP ADDRESS
(1) 49 001300 172014 XDOff: 172014 ;DISPLAY X DYNAMIC OFFSET REGISTER
(1) 50 001302 172016 YDOFF: 172016 ;DISPLAY Y DYNAMIC OFFSET REGISTER
(1) 51 001304 172020 ;
(1) 52 001306 172022 VSCONS: 172022 ;DISPLAY CONSOLE STATUS REGISTER
(1) 53 001310 172024 ;
(1) 54 001312 172026 ;
(1) 55 001314 172030 VSTERM: 172030 ;DISPLAY CHARACTER TERMINATE REGISTER
(1) 56
(1) 57 001316 000320 DDONE: 320 ;DISPLAY INTERRUPT VECTOR FOR STOP
(1) 58 001320 000322 DDONE1: 322 ;
(1) 59 001322 000324 LPVCT: 324 ;DISPLAY INTERRUPT VECTOR FOR LIGHT-PEN
(1) 60 001324 000326 LPVCT1: 326 ;
(1) 61 001326 000330 TIMEVT: 330 ;DISPLAY INTERRUPT VECTOR FOR TIME-OUT OR SHIFT-OUT
(1) 62 001330 000332 TMEVT1: 332 ;
(1) 63 001332 000334 NAMEVT: 334 ;DISPLAY NAME MATCH VECTOR
(1) 64 001334 000336 NAMEV1: 336 ;
(1) 65
(1) 66 .SBTTL INITIAL PROGRAM STARTUP ROUTINE
```

```

68 001336 000005 BEGIN: RESET
69 .SBTTL INITIALIZE THE COMMON TAGS
(1) ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
(1) 001340 012706 001100 MOV #CMTAG,R6 ;;FIRST LOCATION TO BE CLEARED
(1) 001344 005026 CLR (R6)+ ;;CLEAR MEMORY LOCATION
(1) 001346 022706 001140 CMP #SWR,R6 ;;DONE?
(1) 001352 001374 BNE -6 ;;LOOP BACK IF NO
(1) 001354 012706 001100 MOV #STACK,SP ;;SETUP THE STACK POINTER
(1) ;;INITIALIZE A FEW VECTORS
(1) 001360 012737 025776 000020 MOV #SCOPE,@IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
(1) 001366 012737 000340 000022 MOV #340,@IOTVEC+2 ;;LEVEL 7
(1) 001374 013737 006650 006642 MOV $ENDCT,$EOPCT ;;SETUP END-OF-PROGRAM COUNTER
(1) 001402 012737 001402 001106 MOV #,$LPADR ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
(2) ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
(2) ;;EQUAL TO A '-1', SETUP FOR A SOFTWARE SWITCH REGISTER.
(2) 001410 013746 000004 MOV @ERRVEC,-(SP) ;;SAVE ERROR VECTOR
(2) 001414 012737 001450 000004 MOV #64$,@ERRVEC ;;SET UP ERROR VECTOR
(2) 001422 012737 177570 001140 MOV #DSWR,SWR ;;SETUP FOR A HARDWARE SWICH REGISTER
(2) 001430 012737 177570 001142 MOV #DDISP,DISPLAY ;;AND A HARDWARE DISPLAY REGISTER
(2) 001436 022777 177777 177474 CMP #-1,@SWR ;;TRY TO REFERENCE HARDWARE SWR
(2) 001444 001012 BNE 66$ ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
(2) ;;AND THE HARDWARE SWR IS NOT = -1
(2) 001446 000403 BR 65$ ;;BRANCH IF NO TIMEOUT
(2) 001450 012716 001456 64$: MOV #65$,(SP) ;;SET UP FOR TRAP RETURN
(2) 001454 000002 RTI
(2) 001456 012737 000176 001140 65$: MOV #SWREG,SWR ;;POINT TO SOFTWARE SWR
(2) 001464 012737 000174 001142 MOV #DISPREG,DISPLAY
(2) 001472 012637 000004 66$: MOV (SP)+,@ERRVEC ;;RESTORE ERROR VECTOR
(1)
(2) 001476 005037 001200 CLR $PASS ;;CLEAR PASS COUNT
(2) 001502 132737 000200 001213 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
(2) 001510 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
(2) 001512 012737 001214 001140 MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
(2) 001520 67$:
70 001520 012700 001264 RESTAT: MOV #DPC,R0 ;GET POINTER
71 001524 013701 001246 MOV $BASE,R1 ;GET SUPPLIED ADDRESS
72 001530 010120 1$: MOV R1,(R0)+ ;UPDATE
73 001532 062701 000002 ADD #2,R1 ;THE
74 001536 022700 001316 CMP #DDONE,R0 ;ADDRESSES
75 001542 001372 BNE 1$ ;UNTIL DONE
76 001544 012700 001316 MOV #DDONE,R0 ;GET POINTER
77 001550 013701 001242 MOV $VECT1,R1 ;GET SUPPLIED VECTOR
78 001554 042701 160000 BIC #160000,R1 ;CLEAR PSW BITS
79 001560 010120 2$: MOV R1,(R0)+ ;UPDATE
80 001562 062701 000002 ADD #2,R1 ;THE VECTORS
81 001566 022700 001336 CMP #DDONE+20,R0
82 001572 001372 BNE 2$
83 001574 005037 010012 CLR SWITCH ;HOUSEKEEP
84 001600 005037 007664 CLR HOLD ;
85 001604 005004 CLR R4 ;
86 001606 005037 007666 CLR TSAVE ;
87 001612 013777 001262 177440 MOV TKBVT1,@TKBVCT ;RESET KRB VECTOR
88 001620 005077 177436 CLR @TKBVT1
89 001624 005037 002246 CLR KRBD
90 001630 105777 177304 TSTB @SWR ;TEST FOR 'KRB' CONTROL
91 001634 001410 BEQ 3$ ;BR IF NOT

```

```

92 001636 005137 002246          COM      KRB0          ;SET 'KRB' CONTROL
93 001642 012777 002030 177410    MOV      #RETB,@TKBVCT ;SET UP 'KRB' INT
94 001650 012777 000340 177404    MOV      #340,@TKBVT1
95 001656 004737 001666          3$:      JSR      PC,FIXVCT ;LOAD BUS VECTORS
96 001662 000137 002250          JMP      TST1        ;START TESTING
97 001666 012777 007734 177422    FIXVCT: MOV      #STOPI,@DDONE ;SET UP VS-60 DONE VECTOR
98 001674 113700 001243          MOVVB   $VECT1+1,R0   ;GET BR LEVEL
99 001700 042700 177400          BIC     #177400,R0   ;MASK OFF OTHER BITS
100 001704 010077 177410          MOV     R0,@DDONE1
101 001710 013777 001324 177404    MOV     LPVCT1,@LPVCT ;RESET LIGHT-PEN VECTOR
102 001716 005077 177402          CLR    @LPVCT1
103 001722 013777 001330 177376    MOV     TMEVT1,@TIMEVT ;RESET TIME-OUT/SHIFT OUT VECTOR
104 001730 005077 177374          CLR    @TMEVT1
105 001734 013777 001334 177370    MOV     NAMEV1,@NAMEVT ;RESET NAME MATCH VECTOR
106 001742 005077 177366          CLR    @NAMEV1
107 001746 012737 030060 022314    MOV     #30060,DLT14A ;INIT X READOUT VALUE FOR CONSOLE #0
108 001754 012737 030060 022316    MOV     #30060,DLT14A+2
109 001762 012737 030060 022326    MOV     #30060,DLT14B ;INIT Y READOUT
110 001770 012737 030060 022330    MOV     #30060,DLT14B+2
111 001776 012737 030060 022364    MOV     #30060,DLT14C ;RESET READOUT VALUE FOR CONSLE #1
112 002004 012737 030060 022366    MOV     #30060,DLT14C+2
113 002012 012737 030060 022376    MOV     #30060,DLT14D
114 002020 012737 030060 022400    MOV     #30060,DLT14D+2
115 002026 000207          RTS     PC          ;EXIT
116          .SBTTL  KEYBOARD SERVICE ROUTINE
117 002030 117737 177112 007666    RETB:  MOVVB  @TKB,TSAVE ;READ THE CHARACTER
118 002036 042737 177600 007666    BIC     #177600,TSAVE ;MASK TO 7 BITS
119 002044 022737 000003 007666    CMP     #3,TSAVE     ;TEST FOR 'CTRL C'
120 002052 001005          BNE     7$          ;BR IF NOT
121 002054 005777 177206          TST    @DSR
122 002060 100375          BPL     .-4        ; WAIT FOR DISPLAY STOP...
123 002062 000137 001336          JMP     BEGIN      ;...AND RE-START.
124 002066 022737 000015 007666    7$:    CMP     #15,TSAVE ;TEST FOR 'CR'
125 002074 001454          BEQ     5$          ;BR IF
126 002076 005037 010012          CLR    SWITCH     ;CLEAR 'SWITCH'
127 002102 162737 000101 007666    SUB     #101,TSAVE ;MAKE 0-77
128 002110 100443          BMI     4$          ;<A
129 002112 022737 000032 007666    CMP     #32,TSAVE
130 002120 100427          BMI     3$          ;>Z
131 002122 013704 007666          MOV     TSAVE,R4
132 002126 110437 001102          MOVVB  R4,$STNM   ;LOAD TEST #
133 002132 006304          ASL    R4
134 002134 005037 010012          CLR    SWITCH
135 002140 005037 007664          CLR    HOLD
136 002144 005777 177116          TST    @DSR
137 002150 100375          BPL     .-4        ; WAIT FOR DISPLAY STOP.
138 002152 000005          RESET
139 002154 004737 001666          JSR    PC,FIXVCT   ;RESET DISPLAY VECTORS
140 002160 005764 026162          TST    DISPTC(R4) ;TEST IF VALID
141 002164 001001          BNE     2$
142 002166 005004          CLR    R4
143 002170 012706 001100          2$:    MOV     #STACK,SP ; RESET STACK.
144 002174 000174 026162          JMP    @DISPTC(R4) ;EXIT TO THAT TEST SELECTED
145
146 002200 022737 000076 007666    3$:    CMP     #76,TSAVE
147 002206 001013          BNE     6$

```



```

148 002210 012737 177777 007664      MOV    #-1,HOLD      ; RUBOUT
149 002216 000002                    RTI      ;EXIT
150 002220 005037 007664      4$:    CLR    HOLD
151 002224 000002                    RTI
152 002226 012737 177777 010012  5$:    MOV    #-1,SWITCH ;SET 'STOP MOTION' FLAG
153 002234 000002                    RTI
154 002236 162737 000040 007666  6$:    SUB    #40,TSAVE  ;CONVERT LC TO UC
155 002244 000721                    BR      1$
156 002246 000000      KRBD:  0
157
158      .SBTTL  VS-60 INSTRUCTION SET
159
160      100000      CHAR=100000      ;DISPLAY IN CHARACTER MODE
161      104000      SHORTV=104000    ;          SHORT VECTOR
162      110000      LONGV=110000     ;          LONG VECTOR MODE
163      114000      POINT=114000    ;          POINT MODE
164      120000      GRAPHX=120000   ;          GRAPHPLOT X MODE
165      124000      GRAPHY=124000   ;          GRAPHPLOT Y MODE
166      120000      BASICV=GRAPHX    ;          BASIC VECTOR MODE
167      130000      RELATP=130000   ;          RELATIVE POINT MODE
168      134000      BASICS=RELATP!4000 ;          BASIC SHORT VECTOR MODE
169      144000      ABSVCT=144000   ;          ABSOLUTE VECTOR MODE
170
171      010000      OFFST0=10000     ;          ENABLE OFFSET OF 0
172      012000      OFFST1=12000     ;          ENABLE OFFSET OF 1
173      014000      OFFST2=14000     ;          ENABLE OFFSET OF 2
174      016000      OFFST3=16000     ;          ENABLE OFFSET OF 3
175
176      002000      INT0=2000        ;ENABLE INTENSITY LEVEL 0
177      002200      INT1=2200        ;          1
178      002400      INT2=2400        ;          2
179      002600      INT3=2600        ;          3
180      003000      INT4=3000        ;          4
181      003200      INT5=3200        ;          5
182      003400      INT6=3400        ;          6
183      003600      INT7=3600        ;LEVEL 7
184
185      000100      LPOFF=100         ;DISABLE LIGHT-PEN INTERRUPT
186      000140      LPON=140
187      000020      BLKOFF=20
188      000030      BLKON=30         ;DISABLE BLINK
189                                     ;ENABLE BLINK
190      000004      LINE0=4           ;ENABLE LINE TYPE 0
191      000005      LINE1=5           ;ENABLE LINE TYPE 1
192      000006      LINE2=6           ;ENABLE LINE TYPE 2
193      000007      LINE3=7           ;ENABLE LINE TYPE 3
194
195      002000      PATH0=2000        ;DIRECTION 0
196      006000      PATH1=6000        ;DIRECTION 1
197      012000      PATH2=12000       ;DIRECTION 2
198      016000      PATH3=16000       ;DIRECTION 3
199      022000      PATH4=22000       ;          4
200      026000      PATH5=26000       ;          5
201      032000      PATH6=32000       ;          6
202      036000      PATH7=36000       ;          7
203

```

| | | | |
|-----|--------|-------------------------|---|
| 204 | 160000 | DJMP=160000 | :DISPLAY ABSOLUTE JUMP |
| 205 | 161000 | DJMPR=DJMP!BIT9 | :DISPLAY RELATIVE JUMP |
| 206 | 162000 | DJSR=DJMP!BIT10 | :DISPLAY JSR ABSOLUTE |
| 207 | 163000 | DJSRR=DJSR!BIT9 | :DISPLAY JSR RELATIVE |
| 208 | | | |
| 209 | 164000 | DNOP=164000 | |
| 210 | 166000 | DPOP=DNOP!BIT10 | :POP AND RESTORE |
| 211 | 165000 | DPOPNR=DNOP!BIT9 | :POP AND NO RESTORE |
| 212 | 164000 | CONSLO=DNOP | :CONSOLE 0 |
| 213 | 164400 | CONSL1=DNOP!BIT8 | :CONSOLE 1 |
| 214 | | | |
| 215 | 170000 | STATSA=170000 | |
| 216 | 173400 | DSTOP=173400 | |
| 217 | 170002 | DMENU0=STATSA!BIT1 | :DISABLE MENU |
| 218 | 170003 | DMENU1=DMENU0!BIT0 | |
| 219 | | | |
| 220 | 000200 | LPLITE=200 | |
| 221 | 000300 | LPDARK=300 | |
| 222 | 000040 | ITAL0=40 | :DISABLE ITALIC CHARACTERS |
| 223 | 000060 | ITAL1=60 | |
| 224 | 000004 | SYNC30=4 | :ENABLE SYNC OF 30 CPS |
| 225 | 000010 | SYNC40=10 | :ENABLE SYNC OF 40 CPS |
| 226 | | | |
| 227 | 174000 | STATSB=174000 | |
| 228 | | | |
| 229 | 000100 | INCR=100 | :ENABLE 'GRAPHLOT INCREMENT REG. CHANGE'' |
| 230 | | | |
| 231 | 154000 | STATSC=154000 | |
| 232 | 155000 | CHRRTO=STATSC!BIT9 | :DISABLE CHAR ROTATE |
| 233 | 155400 | CHRRT1=CHRRTO!BIT8 | |
| 234 | | | |
| 235 | 154200 | CHARS0=STATSC!BIT7 | :LOAD CHARACTER SCALE TO 1/2 |
| 236 | 154240 | CHARS1=CHARS0!BIT5 | : 1 |
| 237 | 154300 | CHARS2=CHARS0!BIT6 | : 1 1/2 |
| 238 | 154340 | CHARS3=CHARS0!BIT6!BIT5 | : 2 |
| 239 | | | |
| 240 | 154020 | VCTR00=STATSC!BIT4 | :LOAD VECTOR SCALE REGISTER |
| 241 | | | |
| 242 | 176000 | STATE=STATSB!BIT10 | |
| 243 | | | |
| 244 | 176002 | STRNG0=STATE!BIT1 | :DISABLE CHARACTER STRING TERMINATE |
| 245 | 176003 | STRNG1=STRNG0!BIT0 | |
| 246 | | | |
| 247 | 176040 | EDGE0=STATE!BIT5 | :DISABLE EDGE INTERRUPT |
| 248 | 176060 | EDGE1=EDGE0!BIT4 | |
| 249 | 150000 | DNAME=150000 | :LOAD DISPLAY NAME REGISTER |
| 250 | | | |
| 251 | | :MORE EQUATES | |
| 252 | | | |
| 253 | 040000 | INTX=BIT14 | :INTENSIFY |
| 254 | 000177 | MAXMUX=177 | :MAX. MENU X WIDTH |
| 255 | 001777 | MAXX=1777 | :MAX. X AXIS LENGTH |
| 256 | 001777 | MAXY=1777 | :MAX. Y AXIS LENGTH |
| 257 | 000777 | HALFX=MAXX/2 | :HALF OF MAXIMUM LENGTH |
| 258 | 020000 | MINUSX=20000 | :NEGATIVE SIGN BIT |
| 259 | 020000 | MINUSY=20000 | :NEGATIVE SIGN BIT |

| | | | | |
|-----|--------|------------|--|--|
| 260 | 000100 | MINSUY=100 | | :NEGATIVE SIGN BIT <SHORT VECTOR MODE> |
| 261 | 000021 | SUPON=21 | | :SUPER-SCRIPT ENABLE |
| 262 | 000023 | SUPOFF=23 | | :SUPER-SCRIPT DISABLE |
| 263 | 000022 | SUBON=22 | | :SUB-SCRIPT ENABLE |
| 264 | 000024 | SUBOFF=24 | | :SUB-SCRIPT DISABLE |

| | | | | | |
|-----|--|--------|------|--------|-------------|
| 270 | | .SBTTL | | | |
| 271 | | .SBTTL | TEST | LETTER | DESCRIPTION |
| 272 | | .SBTTL | ---- | ----- | ----- |
| 273 | | .SBTTL | | | |

275
(3) :*****
: *TEST 1 A DIRECTORY FRAME
(3) :*****
(2) 002250 000004 TST1: SCOPE

| | | | | | |
|-----|--------|--------|--------|------------------|--------------------------|
| 276 | | | | | |
| 277 | 002252 | 004537 | 007676 | JSR R5,DSPLA | :EXIT TO DISPLAY A FRAME |
| 278 | 002256 | 001000 | | 1000 | |
| 279 | 002260 | 010176 | | FRME0 | :USING THE DIR. FRAME |

282
(3) :*****
: *TEST 2 B ASTIGMATISM AND SETTLING TIME
(3) :*****

| | | | | | |
|-----|--------|--------|--------|------------------|-----------------------------------|
| 283 | 002262 | 000004 | | TST2: SCOPE | |
| 284 | 002264 | 004537 | 007676 | JSR R5,DSPLA | :DISPLAY DATA LOCATED IN 'BUFFER' |
| 285 | 002270 | 020000 | | 20000 | |
| 286 | 002272 | 012230 | | TABB | |

287

```
289 (3)
(3)
(2) 002274 000004
290 002276 012737 000100 007672
291 002304 012700 010074
292
293 002310 012037 017722
294 002314 012037 017724
295
296 002320 100441
297 002322 004537 007676
298 002326 000001
299 002330 017716
300
301 002332 052737 040000 017722
302
303 002340 004537 007676
304 002344 000001
305 002346 017716
306 002350 004537 002434
307 002354 000005
308 002356 004537 007676
309 002362 000001
310 002364 017716
311 002366 004537 002434
312 002372 000005
313
314 002374 004537 007676
315 002400 000001
316 002402 017716
317 002404 004537 002434
318 002410 000005
319
320 002412 004537 007676
321 002416 000001
322 002420 017716
323 002422 000732
324
325 002424 005337 007672 3$: DEC TEMPA ;FINISHED EXECUTION?
326 002430 001325 BNE 1$
327 002432 000416 BR TST4 ;;BR OVER SUBROUTINE
328
329 002434 012537 002466 SECDLY: MOV (R5)+,11$ ;LOAD TOTAL DELAY COUNT
330 002440 013737 001256 002464 2$: MOV DELAY,10$ ;LOAD 1 MS.
331 002446 005337 002464 1$: DEC 10$ ;DELAY
332 002452 001375 BNE 1$
333 002454 005337 002466 DEC 11$ ;DEC MSEC COUNT
334 002460 100367 BPL 2$
335 002462 000205 RTS R5 ;EXIT
336 002464 000000 10$: 0
337 002466 000000 11$: 0
338
```

```
340 ::*****
(3) :*TEST 4 D MINOR AXIS GAIN, OFFSET AND PHASE ADJUSTMENT
(3) :*****
(2) 002470 000004 TST4: SCOPE
341 002472 012737 000010 007672 MOV #BIT3,TEMPA ;LOAD EXECUTION COUNT
342 002500 004537 007676 JSR R5,DSPLA ;DISPLAY SUB-PICTURE
343 002504 001000 1000
344 002506 012472 FRME2
345
346 002510 005337 007672 DEC TEMPA ;FINISHED EXECUTION ?
347 002514 001371 BNE 1$ ;BR IF NOT
348
349 ::*****
(3) :*TEST 5 E MAJOR AXIS OFFSET AND VECTOR START POINT ADJUSTMENT
(3) :*****
(2) 002516 000004 TST5: SCOPE
350 002520 012737 001000 007672 MOV #BIT9,TEMPA
351 002526 004537 007676 JSR R5,DSPLA ;DISPLAY OFFSET SUB-PICTURE
352 002532 000010 10
353 002534 012472 FRME2
354 002536 004537 007676 JSR R5,DSPLA ;DISPLAY VECTOR START SUB-PICTURE
355 002542 000010 10
356 002544 020360 VSTRT
357
358 002546 005337 007672 DEC TEMPA ;FINISHED EXECUTION LOOP?
359 002552 001365 BNE 1$ ;BR IF NOT DONE.
360
361 ::*****
(3) :*TEST 6 F VECTOR LENGTH GAIN, CONVERGENCE AND VECTOR LINEARITY
(3) :*****
(2) 002554 000004 TST6: SCOPE
362 ;GENERATE THE SCREEN PICTURE BUFFER FIRST
363 002556 012700 026246 MOV #BUFFER,R0 ;LOAD DISPLAY PICTURE POINTER
364 002562 012720 164700 MOV #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1
365 002566 012701 012556 MOV #PICST0,R1 ;LOAD 'BOX POINTER'
366 002572 012120 1$: MOV (R1)+,(R0)+ ;GET DATA INTO BUFFER
367 002574 022701 012606 CMP #PICST1,R1 ;TEST FOR END
368 002600 001374 BNE 1$ ;BR IF NOT
369
370 002602 012737 001777 017736 MOV #MAXX,PICVTA ;LOAD STARTING X POSITION
371 002610 012737 000040 017740 MOV #40,PICVTB ;LOAD STARTING Y POSITION
372 002616 012737 060040 017744 MOV #INTX!MINUSX+40,PICVTC ;LOAD INTENSIFY, MINUS DIR AND VALUE
373
374 002624 012737 000037 007672 MOV #37,TEMPA ;LOAD A COUNTER
375 002632 012701 017734 3$: MOV #PICVTL,R1 ;LOAD SUB-PICTURE POINTER
376 002636 012120 2$: MOV (R1)+,(R0)+ ;LOAD DATA
377 002640 022701 017750 CMP #PICVTE,R1 ;TEST FOR END
378 002644 001374 BNE 2$ ;BR IF NOT
379 002646 005337 007672 DEC TEMPA
380 002652 001407 BEQ 4$ ;BR IF DONE THIS SIDE
381 002654 062737 000040 017740 ADD #40,PICVTB ;ADJUST STARTING Y POSITION
382 002662 062737 000040 017744 ADD #40,PICVTC ;ADJUST VECTOR LENGTH
383 002670 000760 BR 3$
384 002672 012737 000000 017736 4$: MOV #0,PICVTA ;LOAD STARTING X POSITION
385 002700 012737 000040 017740 MOV #40,PICVTB ;LOAD STARTING Y POSITION
386 002706 012737 041740 017744 MOV #INTX+1740,PICVTC ;LOAD INTENSIFY AND DELTA VALUE
```

```

387
388 002714 012737 000037 007672      MOV      #37,TEMPA      ;LOAD A COUNTER
389 002722 012701 017734      5$: MOV      #PICVTL,R1      ;LOAD SUB-PICTURE POINTER
390 002726 012120      6$: MOV      (R1)+,(R0)+      ;LOAD DATA
391 002730 022701 017750      CMP      #PICVTE,R1      ;TEST FOR END
392 002734 001374      BNE      6$              ;BR IF NOT
393 002736 005337 007672      DEC      TEMPA          ;TEST IF DONE
394 002742 001407      BEQ      7$              ;BR IF SUB-PICTURE
395 002744 062737 000040 017740      ADD      #40,PICVTB      ;ADJUST STARTING Y POSITION
396 002752 162737 000040 017744      SUB      #40,PICVTC      ;ADJUST VECTOR LENGTH
397 002760 000760      BR       5$              ;BR BACK
398 002762 012720 114000      7$: MOV      #POINT,(R0)+      ;LOAD POINT INST
399 002766 012701 040000      MOV      #INTX,R1        ;LOAD STARTING X POSITION
400 002772 012702 001777      MOV      #MAXY,R2        ;LOAD STARTING Y POSITION
401 002776 010120      8$: MOV      R1,(R0)+      ;LOAD X POSITION
402 003000 010220      MOV      R2,(R0)+      ;LOAD Y POSITION
403 003002 062701 000040      ADD      #40,R1          ;ADJUST X
404 003006 162702 000040      SUB      #40,R2          ;ADJUST Y
405 003012 100371      BPL      8$              ;BR IF NOT DONE
406 003014 012720 114000      MOV      #POINT,(R0)+
407 003020 012720 000000      MOV      #0,(R0)+
408 003024 012720 001777      MOV      #MAXY,(R0)+      ;LOAD POINT IN UPPER LEFT CORN
409 003030 012720 110000      MOV      #LONGV,(R0)+      ;LOAD DECENDING DIAG. LINE
410 003034 012720 041777      MOV      #INTX!MAXX,(R0)+
411 003040 012720 021777      MOV      #MINUSX!MAXY,(R0)+
412      ;DRAW BASIC VECTOR SECTION
413 003044 012720 114000      MOV      #POINT,(R0)+
414 003050 012720 001000      MOV      #1000,(R0)+
415 003054 012720 001000      MOV      #1000,(R0)+
416 003060 012720 120000      MOV      #BASICV,(R0)+      ;LOAD BASIC VECTOR
417 003064 012720 042777      MOV      #INTX!PATH0!HALFX,(R0)+      ;DISPLAY BASIC VECTOR
418 003070 012720 062777      MOV      #INTX!PATH4!HALFX,(R0)+
419 003074 012720 052777      MOV      #INTX!PATH2!HALFX,(R0)+
420 003100 012720 072777      MOV      #INTX!PATH6!HALFX,(R0)+
421 003104 012720 062777      MOV      #INTX!PATH4!HALFX,(R0)+
422 003110 012720 042777      MOV      #INTX!PATH0!HALFX,(R0)+
423 003114 012720 072777      MOV      #INTX!PATH6!HALFX,(R0)+
424 003120 012720 052777      MOV      #INTX!PATH2!HALFX,(R0)+
425 003124 012720 173400      MOV      #DSTOP,(R0)+
426 003130 012720 160000      MOV      #DJMP,(R0)+
427 003134 012720 026246      MOV      #BUFFER,(R0)+
428
429      ;THE PICTURE HAS NOW BEEN COMPLETED
430 003140 012737 003146 001106      20$: MOV      #20$,SLPADR      ;RESET LOOP ADDRESS
431 003146 004537 007676      JSR      R5,DSPLA        ;EXIT TO DISPLAY ROUTINE
432 003152 002000      2000
433 003154 026246      BUFFER                  ;USING BUFFER STORAGE
434
435

```

```

437
(3)
(3)
(2) 003156 000004
438 003160 012700 026246
439 003164 012720 164700
440 003170 004737 003340
441 003174 012701 000020
442 003200 012720 040000
443 003204 012720 001777
444 003210 012720 000100
445 003214 012720 021777
446 003220 005301
447 003222 001366
448 003224 012720 020001
449 003230 012720 000000
450 003234 012720 040000
451 003240 012720 001777
452 003244 004737 003340
453 003250 012701 000020
454 003254 012720 041777
455 003260 012720 000000
456 003264 012720 021777
457 003270 012720 000100
458 003274 005301
459 003276 001366
460 003300 012720 000000
461 003304 012720 020001
462 003310 012720 041777
463 003314 012720 000000
464 003320 012720 173400
465 003324 012720 160000
466 003330 012710 026246
467 003334 000137 003362
468
469 003340 012720 117000
470 003344 012720 000000
471 003350 012720 000000
472 003354 012720 110000
473 003360 000207
474
475 003362 004537 007676
476 003366 004000
477 003370 026246
478

*****
*TEST 7      G      PINCUSHION FRAME
*****
TST7:  SCOPE
      MOV      #BUFFER,R0      ;LOAD START ADDRESS
      MOV      #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOL #1
      JSR      PC,3$           ;LOAD 0,0 ORGIN
      MOV      #20,R1          ;SETUP COUNT
1$:    MOV      #INTX,(R0)+     ;LOAD INT LINE
      MOV      #MAXY,(R0)+     ; MAX Y
      MOV      #100,(R0)+     ;LOAD DELTA X
      MOV      #MINUSX+MAXY,(R0)+ ;LOAD - MAX Y
      DEC      R1              ;FINISHED ?
      BNE     1$              ;BR IF NOT
      MOV      #MINUSX+1,(R0)+ ;GO BACK 1 UNIT
      MOV      #0,(R0)+
      MOV      #INTX,(R0)+
      MOV      #MAXY,(R0)+     ;PLOT LAST LINE
      JSR      PC,3$           ;SET ORGIN
      MOV      #20,R1          ;SETUP COUNT
2$:    MOV      #INTX+MAXX,(R0)+ ;LOAD DELTA X MAX
      MOV      #0,(R0)+       ;LOAD DELTA Y = 0
      MOV      #MINUSX+MAXX,(R0)+ ;RETRACE
      MOV      #100,(R0)+     ;LOAD DELTA Y OF 100
      DEC      R1              ;FINISHED ?
      BNE     2$              ;BR IF NOT
      MOV      #0,(R0)+
      MOV      #MINUSX+1,(R0)+
      MOV      #INTX+MAXX,(R0)+ ;PLOT LAST LINE
      MOV      #0,(R0)+
      MOV      #DSTOP,(R0)+    ;LOAD STOP
      MOV      #DJMP,(R0)+    ;LOAD JUMP
      MOV      #BUFFER,(R0)
      JMP     4$
4$:    MOV      #POINT!INT4,(R0)+ ;LOAD POINT
      MOV      #0,(R0)+       ; AT X
      MOV      #0,(R0)+       ; AT Y
      MOV      #LONGV,(R0)+   ;LONG VECTOR
      RTS     PC               ;EXIT
4$:    JSR      R5,DSPLA      ;EXIT TO DISPLAY FRAME
      4000
      BUFFER                  ;USING THE CROSS HATCH PATTERN
  
```

480
(3)
(3)
(2) 003372 000004
481 003374 004537 007676
482 003400 006000
483 003402 013522
484
485
(3)
(3)
(2) 003404 000004
486 003406 012737 000400 007674
487 003414 012737 154037 020320
488 003422 012737 000020 007672
489
490 003430 004537 007676
491 003434 000001
492 003436 017756
493
494 003440 005337 020320
495 003444 005337 007672
496 003450 001367
497
498 003452 005337 007674
499 003456 001356
500

: *TEST 10 H OCTAGONS AND CIRCLES

TST10: SCOPE
JSR R5,DSPLA ;DISPLAY TEST
6000
FRME3 ;FRAME # 3

: *TEST 11 I SCISSORING AND VECTOR SCALING

TST11: SCOPE
MOV #BIT8,TEMPB ;LOAD EXECUTION COUNTER
1\$: MOV #VCTR00!17,PICSCA ;RELOAD VECTOR SCALE LENGTH TO 17
MOV #20,TEMPA ;LOAD SCALE COUNTER
2\$: JSR R5,DSPLA ;EXIT TO DISPLAY ROUTINE
1
PICSCS ;USING PRESET PICTURE DATA
DEC PICSCA ;REDUCE VECTOR SCALE
DEC TEMPA ;FINISHED ALL SCALES?
BNE 2\$;BR IF NOT
DEC TEMPB ;FINISHED EXECUTION COUNT
1\$ BNE 1\$;BR IF NOT


```
502 :*****  
(3) :*TEST 12 J OFFSET X AND OFFSET Y POSITION  
(3) :*****  
(2) 003460 000004 TST12: SCOPE  
503 :DISPLAY A SQUARE IN THE CENTER SCREEN, THEN  
504 :MOVE THE BOX TO THE RIGHT  
505 003462 012737 010000 015222 MOV #BIT12,OFFT1 ;LOAD BASIC X OFFSET VALUE  
506 003470 012737 010000 015224 MOV #BIT12,OFFT2 ;LOAD BASIC Y OFFSET VALUE  
507 003476 004537 007676 1$: JSR R5,DSPLA ;DISPLAY THAT FRAME  
508 003502 000100 100  
509 003504 015220 OFFTST  
510 003506 005737 010012 TST SWITCH ;TEST IF HOLD HERE  
511 003512 001371 BNE 1$ ;BR IF YES  
512 003514 062737 000001 015222 ADD #1,OFFT1 ;UPDATE THE X OFFSET  
513 003522 022737 011400 015222 CMP #BIT12!1400,OFFT1 ;TEST IF MORE TO MOVE  
514 003530 001362 BNE 1$ ;BR IF NOT  
515  
516 :MOVE THE BOX TO THE LEFT  
517 003532 012737 030000 015222 MOV #BIT12!MINUSX,OFFT1 ;LOAD THE BASIC X OFFSET  
518 003540 012737 010000 015224 MOV #BIT12,OFFT2 ;LOAD THE BASIC Y OFFSET  
519 003546 004537 007676 2$: JSR R5,DSPLA ;DISPLAY THE FRAME  
520 003552 000100 100  
521 003554 015220 OFFTST  
522 003556 005737 010012 TST SWITCH ;TEST IF HOLD HERE  
523 003562 001371 BNE 2$ ;BR IF HOLD  
524 003564 062737 000001 015222 ADD #1,OFFT1 ;UPDATE THE X OFFSET  
525 003572 022737 031400 015222 CMP #BIT12!MINUSX!1400,OFFT1 ;TEST IF MORE  
526 003600 001362 BNE 2$ ;BR IF NOT  
527  
528 :MOVE THE BOX UP  
529 003602 012737 010000 015222 MOV #BIT12,OFFT1 ;LOAD BASIC X OFFSET  
530 003610 012737 010000 015224 MOV #BIT12,OFFT2 ;LOAD BASIC Y OFFSET  
531 003616 004537 007676 3$: JSR R5,DSPLA ;DISPLAY THAT FRAME  
532 003622 000100 100  
533 003624 015220 OFFTST  
534 003626 005737 010012 TST SWITCH ;TEST IF HOLD HERE  
535 003632 001371 BNE 3$ ;BR IF YES  
536 003634 062737 000001 015224 ADD #1,OFFT2 ;UPDATE Y OFFSET  
537 003642 022737 011400 015224 CMP #BIT12!1400,OFFT2 ;TEST IF MORE  
538 003650 001362 BNE 3$ ;BR IF NOT  
539  
540 :MOVE THE BOX DOWN  
541 003652 012737 030000 015224 MOV #BIT12!MINUSY,OFFT2 ;LOAD THE BASIC Y OFFSET  
542 003660 012737 010000 015222 MOV #BIT12,OFFT1 ;LOAD THE BASIC X OFFSET  
543 003666 004537 007676 4$: JSR R5,DSPLA ;DISPLAY THAT FRAME  
544 003672 000100 100  
545 003674 015220 OFFTST  
546 003676 005737 010012 TST SWITCH ;TEST IF HOLD HERE  
547 003702 001371 BNE 4$ ;BR IF YES  
548 003704 062737 000001 015224 ADD #1,OFFT2 ;UPDATE Y OFFSET  
549 003712 022737 031400 015224 CMP #BIT12!MINUSX!1400,OFFT2 ;TEST IF MORE  
550 003720 001362 BNE 4$ ;BR IF NOT
```

```

552
553
554
555
556
557
558
559
560
561
562
563
564 003722
565 003722 005037 015222
566 003726 005037 015224
567 003732 005077 175342
568 003736 005077 175340
569 003742 012777 004036 175346
570 003750 004537 007676
571 003754 000001 015220
572 003760 005000
573 003762 010077 175312
574 003766 010077 175310
575 003772 012737 000100 010006
576 004000 004537 007716
577 004004 005737 010012
578 004010 001364
579
580 004012 005200
581 004014 020027 021400
582 004020 001414
583 004022 020027 001400
584 004026 001355
585 004030 012700 020000
586 004034 000752
587
588 004036 010077 175236
589 004042 010077 175234
590 004046 000137 007734
591
592 004052 005077 175222
593 004056 005077 175220
594 004062 012777 007734 175226
  
```

```

:
: NOW DO IT ALL AGAIN SETTING THE OFFSET REGISTERS DIRECTLY.
:
: MOVE THE BOX DIAGONALLY TO THE UPPER RIGHT (POS, POS)
: THEN TO THE LOWER LEFT (NEG, NEG).
:
: *** NOTE: WHEN DEALING WITH THE OFFSET REGISTERS DIRECTLY,
: IT APPEARS THAT THE NEG DIRECTION (BIT13) GETS CLEARED ON A
: DISPLAY "START" FUNCTION. RESUME SEEMS TO BE OK, HOWEVER TO
: BE SAFE, "START" WITH + OFFSET, AND RELOAD REGISTERS PRIOR
: TO EVERY "RESUME". ??? H - A - C - K ???
  
```

```

TST12A:
CLR OFFT1 ; ZERO OFFSET INSTRUCTIONS...
CLR OFFT2 ; ...IN THE DISPLAY FILE.
CLR @XDOFF ; CLEAR OFFSET REGISTERS.
CLR @YDOFF
MOV #2$,@DDONE ; CHANGE STOP VECTOR.
JSR R5,DSPLA ; XCT DISPLAY...
1, OFFTST ; ...ONCE TO INIT THE OFFSETS.
CLR R0 ; VARIABLE OFFSET VALUE => R0.
1$: MOV R0,@XDOFF ; SET 1ST/NEXT OFFSET.
MOV R0,@YDOFF
MOV #100,COUNT ; SET CYCLE COUNT...
JSR R5,RESUME ; ...AND RESUME DISPLAY.
TST SWITCH
BNE 1$ ; BR IF STOP MOTION IS SET.

INC R0 ; BUMP OFFSET VALUE.
CMP R0,#BIT13!1400 ; ALL DONE ???
BEQ 3$ ; EXIT IF SO.
CMP R0,#1400 ; HALF DONE ???
BNE 1$ ; NOT YET, LOOP.
MOV #BIT13,R0 ; YES, START NEG HALF.
BR 1$

2$: MOV R0,@XDOFF ; *** ON STOP INTERRUPT...
MOV R0,@YDOFF ; ...RELOAD OFFSETS ***
JMP STOPI ; CONTINUE NORMAL STOP SEQ.

3$: CLR @XDOFF ; CLEAR OFFSET REGISTERS.
CLR @YDOFF
MOV #STOPI,@DDONE ; RESET STOP VECTOR.
  
```

596
(3)
(3)
(2) 004070 000004
597 004072 012737 000200 007672
598
599 004100 004537 007676
600 004104 000060
601 004106 020666
602
603 004110 005337 007672
604 004114 001371
605
606
(3)
(3)
(2) 004116 000004
607 004120 012700 026246
608 004124 012720 164700
609 004130 012720 114000
610 004134 005020
611 004136 012720 001763
612 004142 012701 000015
613 004146 012720 162000
614 004152 012720 021424
615 004156 005301
616 004160 001372
617 004162 012720 160000
618 004166 012720 021356
619
620 004172 004537 007676
621 004176 000200
622 004200 026246
623

```
::*****  
:*TEST 13      K      CHARACTER SCALE FRAME  
:******  
TST13: SCOPE  
      MOV      #BIT7,TEMPA      ;LOAD EXECUTION COUNTER  
1$:   JSR      R5,DSPLA      ;DISPLAY SUBPICTURE DATA  
      60  
      CHAQU  
      DEC      TEMPA      ;FINISHED EXECUTION?  
      BNE     1$      ;BR IF NOT  
  
:******  
:*TEST 14      L      CHARACTER QUALITY AND CHARACTER ROTATE IN THE MENU  
:******  
TST14: SCOPE  
      MOV      #BUFFER,R0      ;LOAD DESTINATION POINTER  
      MOV      #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1  
      MOV      #POINT,(R0)+    ;LOAD 'DPOINT'  
      CLR      (R0)+          ;LOAD X AXIS  
      MOV      #MAXY-14,(R0)+  ;LOAD Y  
      MOV      #15,R1         ;LOAD COUNT  
1$:   MOV      #DJSR,(R0)+    ;LOAD 'DJSR' TO BUFFER SPACE  
      MOV      #CHARQA,(R0)+  ;LOAD TARGET ADDRESS  
      DEC      R1             ;FINISHED ?  
      BNE     1$             ;BR UNTIL DONE  
      MOV      #DJMP,(R0)+    ;RETURN ADDRESS TO THE ROTATED CHAR SUB-PIC  
      MOV      #ROTCHR,(R0)+  
      JSR      R5,DSPLA      ;EXECUTE DISPLAY FILE  
      200  
      BUFFER      ;STARTING AT 'BUFFER' ADDRESS
```

625
(3)
(3)
(2) 004202 000004
626
627 004204 012700 026246
628 004210 012720 155000
629 004214 012720 164700
630 004220 012720 114000
631 004224 005020
632 004226 012720 001700
633 004232 012720 100000
634 004236 012737 000100 004436
635 004244 004737 004374
636 004250 012737 000140 004436
637 004256 004737 004374
638 004262 012737 000040 004436
639 004270 004737 004374
640 004274 012720 170040
641 004300 004737 004340
642 004304 004737 004500
643 004310 012720 170060
644 004314 004737 004340
645 004320 012720 173400
646 004324 012720 160000
647 004330 012720 026246
648 004334 000137 004516
649
650 004340 112720 000016
651 004344 005002
652 004346 110220
653 004350 005202
654 004352 022702 000017
655 004356 001774
656 004360 022702 000040
657 004364 001370
658 004366 012720 020017
659 004372 000207
660
661 004374 012720 170040
662 004400 013702 004436
663 004404 004737 004462
664 004410 004737 004500
665 004414 012720 170060
666 004420 013702 004436
667 004424 004737 004462
668 004430 004737 004440
669 004434 000207
670
671 004436 000000

```
::*****  
:*TEST 15 M CHARACTER SET, SUPERSCRIPT, SUBSCRIPT AND ITALICS  
:*****  
TST15: SCOPE  
:SET UP THE BUFFER FOR THIS TEST  
MOV #BUFFER,R0  
MOV #CHRRTO,(R0)+ ;DISABLE CHAR. ROTATE  
MOV #CONSL1!BIT7!BIT6,(R0)+ ;ENABLE CONSOLE #1  
MOV #POINT,(R0)+ ;LOAD POINT MPDE  
CLR (R0)+  
MOV #MAXY-77,(R0)+  
MOV #CHAR,(R0)+  
MOV #100,STCHAR ;LOAD INITIAL CHAR.  
JSR PC,LOADBF  
MOV #140,STCHAR ;LOAD INITIAL IC CHAR  
JSR PC,LOADBF ;LOAD LINE  
MOV #40,STCHAR ;LOAD NUMBERS AND PUNCT  
JSR PC,LOADBF ;LOAD LINE  
MOV #STATSA!ITAL0,(R0)+ ;LOAD NORMAL FONT  
JSR PC,LOADSP ;LOAD SPECIAL CHARS  
JSR PC,SPACE ;INSERT SPACES  
MOV #STATSA!ITAL1,(R0)+ ;LOAD ITALICS FONT  
JSR PC,LOADSP ;LOAD SPIECAL  
MOV #DSTOP,(R0)+ ;LOAD DSTOP  
MOV #DJMP,(R0)+  
MOV #BUFFER,(R0)+  
JMP FILE4A  
  
LOADSP: MOV #16,(R0)+ ;LOAD 'SHIFT-OUT' CHARACTER  
CLR R2 ;SET INITIAL SHIFT OUT CHAR  
1$: MOV R2,(R0)+ ;LOAD CHAR  
2$: INC R2  
CMP #17,R2 ;TEST FOR 'SHIFT-IN' (SI)  
BEQ 2$ ;BR IF SI '17'  
CMP #40,R2 ;FINISHED ?  
BNE 1$ ;BR IF NOT  
MOV #20017,(R0)+ ;LOAD SHIFT-IN SPACE  
RTS PC ;EXIT  
  
LOADBF: MOV #STATSA!ITAL0,(R0)+ ;LOAD NORMAL FONT  
MOV STCHAR,R2 ;GET STARTING CHAR  
JSR PC,FILLIT ;LOAD THE CHARACTERS  
JSR PC,SPACE ;INSERT SPACES  
MOV #STATSA!ITAL1,(R0)+ ;LOAD ITALICS FONT  
MOV STCHAR,R2 ;GET STARTING CHARACTER  
JSR PC,FILLIT ;LOAD THE CHARACTERS  
JSR PC,ACRLF ;INSERT CR-LF  
RTS PC ;EXIT  
  
STCHAR: 0
```

673
674
675
676 004440 112720 000015
677 004444 112720 000012
678 004450 112720 000012
679 004454 112720 000012
680 004460 000207
681
682
683
684 004462 012703 000040
685 004466 110220
686 004470 005202
687 004472 005303
688 004474 001374
689 004476 000207
690
691
692
693 004500 012703 000010
694 004504 112720 000040
695 004510 005303
696 004512 001374
697 004514 000207
698

;LOAD CR-LF'S TO VERTICALLY SPACE THE STRINGS

ACRLF: MOVB #15,(R0)+
 MOVB #12,(R0)+
 MOVB #12,(R0)+
 MOVB #12,(R0)+
 RTS PC ;EXIT

;FILL IN WITH AN INCREMENTING CHARACTERS

FILLIT: MOV #40,R3
1\$: MOVB R2,(R0)+
 INC R2
 DEC R3
 BNE 1\$
 RTS PC

;LOAD "SPACE" CHAR TO SEPERATE CHAR STRINGS

SPACE: MOV #10,R3
1\$: MOVB #40,(R0)+ ;LOAD A SPACE
 DEC R3
 BNE 1\$;BR IF NOT DONE
 RTS PC ;EXIT

```

700
701
702 004516 012737 000600 007672 FILE4A: MOV #600,TEMPA ;LOAD A COUNTER
703 004524 012737 155000 026246 MOV #CHRRTO,BUFFER ;DISABLE ROTATE
704 004532 012737 001700 026256 1$: MOV #MAXY-77,BUFFER+10 ;LOAD STARTING Y POINT
705 004540 004537 007676 JSR R5,DSPLA ;DISPLAY IN UPPER HALF OF SCREEN
706 004544 000001 1
707 004546 026246 BUFFER
708
709 004550 012737 000400 026256 MOV #400,BUFFER+10 ;LOAD STARTING Y POINT IN THE LOWER HALF
710 004556 004537 007676 JSR R5,DSPLA ;DISPLAY IN LOWER HALF OF SCREEN
711 004562 000001 1
712 004564 026246 BUFFER
713
714 004566 004537 007676 JSR R5,DSPLA ;DISPLAY SUPER AND SUBSCRIPT IN THE MIDDLE
715 004572 000001 1
716 004574 015256 SUPPIC
717
718 004576 005337 007672 DEC TEMPA ;FINISHED ?
719 004602 001353 BNE 1$ ;BR IF NOT
720 004604 005737 010012 TST SWITCH ;TEST IF 'FREEZE'
721 004610 001342 BNE FILE4A ;BR IF YES
722
723 ;NOW DISPLAY THE CHARACTER SET FRAME ROTATED
724
725 004612 012737 000600 007672 2$: MOV #600,TEMPA ;LOAD DELAY COUNTER FOR THIS HALF
726 004620 012737 155400 026246 MOV #CHRRT1,BUFFER ;ENABLE CHAR. ROTATE
727 004626 005037 026256 CLR BUFFER+10 ;RESET Y ORGIN
728 004632 012737 000100 026254 3$: MOV #100,BUFFER+6 ;LOAD X ORGIN
729 004640 004537 007676 JSR R5,DSPLA ;DISPLAY FRAME
730 004644 000001 1
731 004646 026246 BUFFER
732
733 004650 012737 001400 026254 MOV #1400,BUFFER+6 ;REPOSITION THE X ORGIN
734 004656 004537 007676 JSR R5,DSPLA ;DISPLAY FRAME AT NEW ORGIN
735 004662 000001 1
736 004664 026246 BUFFER
737
738 004666 004537 007676 JSR R5,DSPLA ;DISPLAY TEXT
739 004672 000001 1
740 004674 015304 SUPCO
741
742 004676 005337 007672 DEC TEMPA ;FINISHED DELAY ?
743 004702 001353 BNE 3$ ;BR IF NOT
744 004704 005737 010012 TST SWITCH ;TEST IF FREEZE
745 004710 001340 BNE 2$ ;BR IF YES
746

```

748
(3)
(3)
(2) 004712 000004
749 047516
750 030064
751 030063
752
753 004714 012777 000377 174372
754 004722 012737 047516 015504
755 004730 012737 170000 015412
756 004736 004537 007676
757 004742 010000
758 004744 015412
759 004746 005737 010012
760 004752 001371
761
762 004754 012777 000377 174332
763 004762 012737 030064 015504
764 004770 012737 170010 015412
765 004776 004537 007676
766 005002 000200
767 005004 015412
768 005006 005737 010012
769 005012 001371
770
771 005014 012777 000377 174272
772 005022 012737 030063 015504
773 005030 012737 170004 015412
774 005036 004537 007676
775 005042 000200
776 005044 015412
777 005046 005737 010012
778 005052 001371
779

```
*****  
*TEST 16 N SYNC SPEED AND CHARACTER TERMINATE TEST  
*****  
TST16: SCOPE  
NO=47516 ;ASCII VALUE FOR 'NO'  
S40=30064 ; " " '40'  
S30=30063 ; " " '30'  
  
1$: MOV #377,@VSTERM ;LOAD TERMINATE REG.  
MOV #NO,SYNSPD ;LOAD SYNC ASCII VALUE  
MOV #STATSA,SYNPIC ;LOAD NO SYNC ENABLE  
JSR R5,DSPLA ;DISPLAY THAT FRAME WITH 'NO' SYNC  
10000  
SYNPIC  
TST SWITCH ;TEST IF HOLD SET  
BNE 1$ ;BR IF HOLD  
  
2$: MOV #377,@VSTERM ;LOAD TERMINATE REG.  
MOV #S40,SYNSPD ;LOAD SYNC ASCII VALUE  
MOV #STATSA!SYNC40,SYNPIC ;LOAD SYNC ENABLE TO 40  
JSR R5,DSPLA ;DISPLAY THAT FRAME WITH '40' SYNC  
200  
SYNPIC  
TST SWITCH ;TEST IF HOLD SET  
BNE 2$ ;BR IF HOLD  
  
3$: MOV #377,@VSTERM ;LOAD TERMINATE REG.  
MOV #S30,SYNSPD ;LOAD ASCII VALUE OF 30  
MOV #STATSA!SYNC30,SYNPIC ;LOAD 30 CPS ENABLE  
JSR R5,DSPLA ;DISPLAY THAT FRAME AT '30' SYNC  
200  
SYNPIC  
TST SWITCH ;TEST IF HOLD  
BNE 3$ ;BR IF HOLD
```

781
 (3)
 (3)
 (2) 005054 000004
 782 005056 004537 007676
 783 005062 020000
 784 005064 015634
 785
 786
 (3)
 (3)
 (2) 005066 000004
 787 005070 012700 026246
 788 005074 012737 041776 010070
 789 005102 012737 000001 010072
 790 005110 004737 005214
 791 005114 062737 000002 010072
 792 005122 023727 010072 001777
 793 005130 003767
 794 005132 012737 001777 010072
 795 005140 000407
 796 005142 162737 000002 010070
 797 005150 023727 010070 040000
 798 005156 002403
 799 005160 004737 005214
 800 005164 000766
 801 005166 012720 173400
 802 005172 012720 160000
 803 005176 012720 016062
 804 005202 004537 007676
 805 005206 000200
 806 005210 016062
 807 005212 000407
 808 005214 013720 010070
 809 005220 013720 010072
 810 005224 005020
 811 005226 005020
 812 005230 000207

```

:*****
:*TEST 17      0      DASH LINES AND BLINK
:*****
TST17: SCOPE
        JSR      R5,DSPLA      ;EXIT TO DISPLAY A FRAME
        20000
        FRME5      ;USING THE DASH AND BLINK FRAME

:*****
:*TEST 20      P      VECTOR SPRAY (LENGTH) TEST
:*****
TST20: SCOPE
        MOV      #BUFFER,R0      ;LOAD BUFFER POINTER
        MOV      #INTX!MAXX-1,DELTX6 ;LOAD X PRESET VALUE
        MOV      #1,DELTY6      ;LOAD Y PRESET VALUE
1$:     JSR      PC,SPRAY      ;LOAD INCREASING ANGLE VECTOR
        ADD      #2,DELTY6      ;UPDATE Y LENGTH
        CMP      DELTY6,#MAXY    ;TEST IF END
        BLE      1$            ;BR IF NOT
        MOV      #MAXY,DELTY6    ;RESET MAX Y LENGTH
        BR      4$
3$:     SUB      #2,DELTX6      ;REDUCE X LENGHT
        CMP      DELTX6,#INTX    ;TEST IF END
        BLT      2$            ;BR IF DONE
4$:     JSR      PC,SPRAY      ;LOAD DECREASING ANGLE VECTOR
        BR      3$
2$:     MOV      #DSTOP,(R0)+    ;LOAD STOP
        MOV      #DJMP,(R0)+
        MOV      #FRME6,(R0)+    ;RESTART DISPLAY FRAME
        JSR      R5,DSPLA      ;DISPALY PICTURE
        200                      ;COUNT
        FRME6
        BR      TST21          ;;BR TO NEXT TEST
SPRAY: MOV      DELTX6,(R0)+    ;LOAD X VECTOR LENGTH
        MOV      DELTY6,(R0)+    ;LOAD Y VECTOR LENGTH
        CLR      (R0)+          ;VECTOR BACK TO ORGIN
        CLR      (R0)+
        RTS      PC            ;EXIT
  
```



```
814 (3) *****  
(3) *TEST 21 Q HORIZONTAL PHOSPHOR TEST  
(2) 005232 000004 TST21: SCOPE  
815 005234 005037 016136 CLR DELTX7  
816 005240 012737 000004 007672 1$: MOV #4,TEMPA ;LOAD DELAY COUNT  
817 005246 004537 007676 2$: JSR R5,DSPLA ;EXIT TO DISPLAY A FRAME  
818 005252 000004 4  
819 005254 016134 FRME10 ;USING THE HORIZ FRAME  
820 005256 004537 007676 JSR R5,DSPLA ;EXIT TO DISPLAY A FRAME  
821 005262 000001 1  
822 005264 017150 FRM10 ;USING THE PERIMETER BOX  
823 005266 005737 010012 TST SWITCH ;TEST THE 'MOTION-SWITCH'  
824 005272 001362 BNE 1$ ;BR IF FREEZE THE MOVEMENT  
825 005274 005337 007672 DEC TEMPA ;DELAY DONE ?  
826 005300 100362 BPL 2$ ;BR IF NOT  
827 005302 005237 016136 INC DELTX7 ;UPDATE THE X ORIGIN  
828 005306 022737 001777 016136 CMP #1777,DELTX7 ;TEST IF THE END  
829 005314 001351 BNE 1$ ;BR IF NOT  
830 (3) *****  
(3) *TEST 22 R VERTICAL PHOSPHOR TEST  
(2) 005316 000004 TST22: SCOPE  
831 005320 005037 016676 CLR DELT11  
832 005324 005037 016416 CLR DELTY7  
833 005330 012737 000004 007672 1$: MOV #4,TEMPA ;LOAD DELAY COUNT  
834 005336 004537 007676 2$: JSR R5,DSPLA ;EXIT TO DISPLAY A FRAME  
835 005342 000004 4  
836 005344 016412 FRME11 ;USING THE VERT FRAME  
837 005346 004537 007676 JSR R5,DSPLA ;EXIT TO DISPLAY A FRAME  
838 005352 000001 1  
839 005354 017150 FRM10 ;USING THE PERIMETER BOX  
840 005356 004537 007676 JSR R5,DSPLA ;DISPLAY THE MENU BOX  
841 005362 000001 1  
842 005364 017210 FRM11M  
843 005366 004537 007676 JSR R5,DSPLA ;DISPLAY THE TEST IN THE MENU  
844 005372 000004 4  
845 005374 016672 FRM11S ;DISPLAY THE 'MENU' PHOSPHOR PIC.  
846 005376 005737 010012 TST SWITCH ;TEST THE 'MOTION-SWITCH'  
847 005402 001352 BNE 1$ ;BR IF FREEZE THE MOVEMENT  
848 005404 005337 007672 DEC TEMPA ;DELAY DONE ?  
849 005410 100352 BPL 2$ ;BR IF NOT  
850 005412 022737 001277 016676 CMP #1277,DELT11 ;TEST IF AT TOP OF MENU  
851 005420 001402 BEQ 3$ ;BR IF YES, DONT ADVANCE THE MENU  
852 005422 005237 016676 INC DELT11 ;UPDATE THE Y MENU ORGIN  
853 005426 005237 016416 3$: INC DELTY7 ;UPDATE THE Y ORIGIN  
854 005432 022737 001777 016416 CMP #1777,DELT7 ;TEST IF THE END  
855 005440 001333 BNE 1$ ;BR IF NOT
```

```
857 (3) (3) (2) 005442 000004  
858 005444 012700 026246  
859 005450 012720 114000  
860 005454 012720 000240  
861 005460 012720 001000  
862 005464 012720 104000  
863 005470 004737 005522  
864 005474 012720 130000  
865 005500 004737 005522  
866 005504 012720 173400  
867 005510 012720 160000  
868 005514 012720 026246  
869 005520 000413  
870  
871 005522 012737 000024 007670 LOADVT: MOV #24,CNTR ;LOAD A COUNTER  
872 005530 012720 040077 1$: MOV #INTX+77,(R0)+ ;LOAD A DELTA Y  
873 005534 012720 004177 MOV #4177,(R0)+ ;LOAD A DELTA X,Y  
874 005540 005337 007670 DEC CNTR ;FINISHED?  
875 005544 001371 BNE 1$ ;BR IF NOT  
876 005546 000207 RTS PC ;EXIT  
877  
878 ;DISPLAY FOUR SHORT VECTOR/RELATIVE POINT OCTAGONS IN DIFFERENT QUADRANTS  
879 005550 012737 006000 007672 FIL14A: MOV #6000,TEMPA ;LOAD COUNTER  
880 005556 012737 000200 017254 1$: MOV #200,FRM14A ;LOAD FIRST OCTAGON  
881 005564 012737 000200 017256 MOV #200,FRM14B  
882 005572 004537 007676 JSR R5,DSPLA ;DISPLAY OCT.  
883 005576 000001 1  
884 005600 017250 FRME14  
885 005602 012737 001400 017254 MOV #1400,FRM14A ;LOAD SECOND OCTAGON  
886 005610 012737 000200 017256 MOV #200,FRM14B  
887 005616 004537 007676 JSR R5,DSPLA ;DISPLAY 2ND OCT.  
888 005622 000001 1  
889 005624 017250 FRME14  
890 005626 012737 001400 017254 MOV #1400,FRM14A ;LOAD THIRD OCTAGON  
891 005634 012737 001400 017256 MOV #MAXY-377,FRM14B  
892 005642 004537 007676 JSR R5,DSPLA  
893 005646 000001 1  
894 005650 017250 FRME14  
895 005652 012737 000200 017254 MOV #200,FRM14A ;LOAD FOURTH OCTAGON  
896 005660 012737 001400 017256 MOV #MAXY-377,FRM14B  
897 005666 004537 007676 JSR R5,DSPLA ;DISPLAY 4TH OCT.  
898 005672 000001 1  
899 005674 017250 FRME14  
900 ;NOW DISPLAY THE SHORT VECTOR/RELATIVE POINT VERTICAL LINES  
901 005676 004537 007676 JSR R5,DSPLA ;DISPLAY BAR  
902 005702 000001 1  
903 005704 026246 BUFFER  
904 005706 005337 007672 DEC TEMPA ;FINISHED ?  
905 005712 001321 BNE 1$ ;BR IF NOT  
906
```

```

908
(3)
(3)
(2) 005714 000004
909 005716 012737 174100 017406
910 005724 004537 007676
911 005730 002000
912 005732 017362
913 005734 005237 017406
914 005740 022737 174110 017406
915 005746 001366
916
917
(3)
(3)
(2) 005750 000004
918 005752 012777 007242 173342
919 005760 113777 001243 173336
920 005766 042777 177400 173330
921 005774 012737 000010 007656
922 006002 012737 022454 022344
923 006010 012737 022534 022414
924 006016 012700 026246
925 006022 012737 000100 007654
926 006030 012720 117600
927 006034 012720 001400
928 006040 012720 000300
929 006044 004737 007204
930 006050 012720 173400
931 006054 012720 160000
932 006060 012720 026246
933 006064 005037 007652
934 006070 012737 030060 022446
935 006076 012737 030060 022444
936
937 006104 005037 007650
938 006110 004537 007676
939 006114 000004
940 006116 022224
941
942 006120 005237 007650
943 006124 004537 007676
944 006130 000001
945 006132 026246
946
947 006134 005337 007654
948 006140 001361
949 006142 005337 007656
950 006146 001323
951 006150 013777 001324 173144
952 006156 005077 173142
953

```

```

*****
*TEST 24 T GRAPHPLOT INCREMENT REGISTER TEST USING GRAPHPLOT X AND Y
*****
TST24: SCOPE
MOV #STATSB!INCR,GRPINC ;LOAD BASIC INCREMENT VALUE
1$: JSR R5,DSPLA ;DISPLAY FRAME
2000
GRAPH
INC GRPINC ;UPDATE INCR. VALUE
CMP #STATSB!INCR+10,GRPINC ;TEST IF #10
BNE 1$ ;BR IF NOT

*****
*TEST 25 U INTENSITY LEVEL AND LIGHT PEN TEST
*****
TST25: SCOPE
MOV #RET14,@LPVCT ;LOAD LIGHT PEN VECTOR
MOVB $VECT1+1,@LPVCT1
BIC #177400,@LPVCT1 ;MASK
MOV #10,DSAVE1 ;SET UP COUNT
MOV #PENOF0,MSOPEN ;RESET PEN MESSAGE #0
MOV #PENOF1,MS1PEN ;RESET PEN MESSAGE #1
1$: MOV #BUFFER,R0 ;LOAD START ADDR.
MOV #100,DSAVE
MOV #POINT!INT7,(R0)+ ;LOAD POINT
MOV #1400,(R0)+ ;LOAD X POINT
MOV #300,(R0)+ ;LOAD Y POINT
JSR PC,LOADUP ;LOAD UP THE BUFFER
MOV #DSTOP,(R0)+ ;LOAD DSTOP
MOV #DJMP,(R0)+ ;LOAD DJUMP
MOV #BUFFER,(R0)+ ;LOAD RETURN ADDRESS
CLR HITCNT ;CLEAR HIT COUNT
MOV #30060,FRM16B-2 ;PRESET THE HIT COUNT VALUE
MOV #30060,FRM16B-4

2$: CLR VIEW
JSR R5,DSPLA ;EXIT TO DISPLAY FRAME
4
FRME16 ;SUB-PICTURE

INC VIEW
JSR R5,DSPLA ;EXIT TO DISPLAY FRAME
1
BUFFER

DEC DSAVE ;FINISHED ?
BNE 2$ ;BR IF NOT MINI-LOOP
DEC DSAVE1 ;FINISHED ?
BNE 1$ ;BR IF NOT
MOV LPVCT1,@LPVCT ;RESET VECTOR
CLR @LPVCT1

```

```

955
(3)
(3)
(2) 006162 000004
956 006164 013700 001242
957 006170 042700 160000
958 006174 012720 006372
959 006200 012720 000200
960 006204 012720 006456
961 006210 012720 000340
962 006214 012720 006470
963 006220 012720 000340
964 006224 012720 006502
965 006230 012720 000340
966 006234 005037 177776
967
968 006240 012777 024754 173016
969 006246 012702 000400
970 006252 012700 001336
971 006256 112001
972 006260 042701 177700
973 006264 022700 026246
974 006270 001770
975 006272 005037 007650
976 006276 005301
977 006300 001376
978
979 006302 052777 000200 172766
980 006310 012703 010000
981 006314 017737 172744 001126
982 006322 012737 024754 001124
983 006330 023737 001124 001126
984 006336 101066
985 006340 012737 025430 001124
986 006346 023737 001124 001126
987 006354 103463
988 006356 005737 007650
989 006362 001335
990 006364 005303
991 006366 001352
992 006370 000426
993 006372 105777 172700
994 006376 100407
995 006400 005777 172662
996 006404 100053
997 006406 012777 024754 172650
998 006414 000002
999 006416 005302
1000 006420 001407
1001 006422 012777 000001 172634
1002 006430 052737 000001 007650
1003 006436 000002
1004 006440 022626
1005 006442 000137 006620

*****
*TEST 26 W DYNAMIC EXT. DISPLAY STOP
*****
TST26: SCOPE
MOV $VECT1,R0 ;LOAD VECTOR POINTER
BIC #160000,R0 ;MASK
MOV #4,(R0)+ ;LOAD STOP VECTOR
MOV #200,(R0)+
MOV #BAD1,(R0)+ ;LOAD UNEXPT. INTR
MOV #340,(R0)+
MOV #BAD2,(R0)+ ;LOAD UNEXPT. INTR
MOV #340,(R0)+
MOV #BAD3,(R0)+ ;LOAD UNEXPT. INTR
MOV #340,(R0)+
CLR PSW
;START DISPLAY AND DELAY
MOV #FRME17,@DPC ;START DISPLAY
MOV #400,R2 ;LOAD TIMER COUNTER
1$: MOV #BEGIN,R0 ;LOAD RANDOM NUMBER POINTER
2$: MOVB (R0)+,R1 ;GET A RANDOM NUMBER
BIC #177700,R1 ;MASK OFF OTHER BITS
CMP #BUFFER,R0 ;TEST IF DONE
BEQ 1$ ;BR BACK
CLR DIDINT ;CLEAR 'DID INTERRUPT '' FLAG
3$: DEC R1 ;DELAY
BNE 3$
;NOW SET EXT. STOP FLAG
BIS #BIT7,@DSR1 ;SET EXT. STOP FLAG
MOV #BIT12,R3 ;LOAD DELAY COUNTER
7$: MOV @DPC,$BDDAT ;TEST IF DPC IS OUT OF RANGE
MOV #FRME17,$GDDAT ;LOAD LOW LIMIT
CMP $GDDAT,$BDDAT ;COMAPRE
BHI BAD4 ;BR IF TOO LOW
MOV #FRM17F+4,$GDDAT ;LOAD HIGH LIMIT
CMP $GDDAT,$BDDAT ;COMAPRE
BLO BAD5 ;BR IF TOO HIGH
TST DIDINT ;TEST IF EXT. STOP INTR. OCCURRED
BNE 2$ ;BR IF YES
DEC R3 ;DELAY
BNE 7$ ;BR AND TEST DPC VALUE
BR BAD0 ;NO EXT. STOP INTR. REPORT ERROR
4$: TSTB @DSR1 ;TEST IF EXT. STOP FLAG
BMI 5$ ;BR IF EXT. STOP
TST @DSR ;TEST FOR DISPLAY STOP
BPL BAD6 ;BR IF NOT
MOV #FRME17,@DPC ;START DPU IF NOT EXT. STOP
RTI ;RETURN
5$: DEC R2 ;FINISHED ?
BEQ 6$ ;BR IF DONE
MOV #BIT0,@DPC ;RESUME THE DPU IF EXT. STOP AND NOT FILISHED LO
BIS #1,DIDINT ;SET EXT. STOP FLAG DID INTR.
RTI ;RETURN
6$: CMP (SP)+,(SP)+ ;CLEAN THE STACK
JMP $EOP

```

```
1007 ;BR HERE IF AN ERROR OCCURRED
1008
1009 006446 012737 025462 025450 BAD0: MOV #WHY0,WHY ;INDICATE NO EXT. STOP INTERRUPT
1010 006454 000432 BR BADDON
1011 006456 012737 025516 025450 BAD1: MOV #WHY1,WHY ;INDICATE UNEXPECTED INTR.
1012 006464 022626 CMP (SP)+,(SP)+
1013 006466 000425 BR BADDON
1014 006470 012737 025562 025450 BAD2: MOV #WHY2,WHY ;INDICATE UNEXPECTED INTR.
1015 006476 022626 CMP (SP)+,(SP)+
1016 006500 000420 BR BADDON
1017 006502 012737 025626 025450 BAD3: MOV #WHY3,WHY ;INDICATE UNEXPECTED INTR.
1018 006510 022626 CMP (SP)+,(SP)+
1019 006512 000413 BR BADDON
1020 006514 012737 025672 025450 BAD4: MOV #WHY4,WHY ;INDICATE DPC WAS TOO LOW
1021 006522 000407 BR BADDON
1022 006524 012737 025712 025450 BAD5: MOV #WHY5,WHY ;INDICATE DPC WAS TOO HIGH
1023 006532 000403 BR BADDON
1024 006534 012737 025734 025450 BAD6: MOV #WHY6,WHY ;INDICATE DONE INTR. BUT NO FLAG
1025
1026
1027 006542 017737 172516 006612 BADDON: MOV @DPC,PCERR ;SAVE DPC
1028 006550 017737 172512 006614 MOV @DSR,SRERR ;SAVE SR
1029 006556 017737 172514 006616 MOV @DSR1,SR1ERR ;SAVE SR1
1030 006564 000240 NOP
1031 006566 000240 NOP
1032 006570 012777 007734 172520 MOV #STOPI,@DDONE ;LOAD DISPLAY STOP VECTOR
1033 006576 004537 007676 JSR R5,DSPLA ;DISPLAY ERROR MESSAGE
1034 006602 040000 BIT14
1035 006604 025426 FRM17E
1036 006606 000005 RESET
1037 006610 000403 BR $EOP ;END OF PASS
1038
1039 006612 000000 PCERR: 0 ;D.P.C. UPON ERROR
1040 006614 000000 SRERR: 0 ;SR UPON ERROR
1041 006616 000000 SR1ERR: 0 ;SR1 UPON ERROR
```



```

1054 (3)
1055 (3)
1056 (2) 006700 000004
1057 006702 012737 030060 024716
1058 006710 012737 030060 024720
1059 006716 012700 026246
1060 006722 012701 001000
1061 006726 005020
1062 006730 005301
1063 006732 001375
1064 006734 012720 160000
1065 006740 012720 024564
1066 006744 012737 161010 024724
1067 006752 005037 007650
1068 006756 012700 026246
1069 006762 012701 002000
1070 006766 012777 007022 172264
1071 006774 012777 000200 172260
1072 007002 052777 000100 172134
1073 007010 004537 007676
1074 007014 000001
1075 007016 024564
1076 007020 000736
1077 007022 017703 172120
1078 007026 042703 177600
1079 007032 005301
1080 007034 001443
1081 007036 022703 000003
1082 007042 001002
1083 007044 000137 006620
1084 007050 022703 000016
1085 007054 001005
1086 007056 005237 007650
1087 007062 012737 164000 024724
1088 007070 005737 007650
1089 007074 001415
1090 007076 022703 000017
1091 007102 001005
1092 007104 005037 007650
1093 007110 012737 161010 024724
1094 007116 122703 000037
1095 007122 100002
1096 007124 042703 177740
1097 007130 110320
1098 007132 012702 024722
1099 007136 004737 010014
1100 007142 000002
1101 007144 022626
1102 007146 013702 001264
1103 007152 052762 000200 000012
1104 007160 000656

*****
*TEST 27 V KEYBOARD CHARACTER ECHO LOOP
*****
TST27: SCOPE
MOV #30060,ECODEV-4 ;PRESET READOUT TO 00
MOV #30060,ECODEV-2 ;PRESET READOUT TO 00
20$: MOV #BUFFER,R0 ;LOAD BUFFER POINTER
MOV #512.,R1 ;LOAD CHARACTER COUNT
1$: CLR (R0)+ ;CLEAR THE BUFFER
DEC R1 ;FINISHED ?
BNE 1$ ;BR IF NOT
MOV #DJMP,(R0)+ ;LOAD JUMP RETURN TO START OF BUFFER
MOV #ECHOFR,(R0)+ ; THE ECHO FRAME
MOV #161010,ECHJMP ;PRESET JUMP
CLR SHIFTO ;CLEAR SHIFT IND.
MOV #BUFFER,R0 ;LOAD BUFFER POINTER
MOV #1024.,R1 ;LOAD CHARACTER COUNT
MOV #10$,@TKBVCT ;LOAD INTR. RETURN
MOV #200,@TKBVT1 ;LOAD RETURN INTR. LEVEL
3$: BIS #BIT6,@$TKS ;ENABLE KEYBOARD INTR.
JSR R5,DSPLA ;DISPLAY FRAME AND BUFFER
1
ECHOFR ;ADDRESS OF SUB-PICTURE
BR 20$ ;BR UPON EXT. STOP INTERRUPT
;RETURN HERE UPON KEYBOARD INTR. <D.P.U. SHOULD STILL BE RUNNING>
10$: MOV @$TKB,R3 ;READ KEYBOARD DATA
BIC #177600,R3 ;MASK TO LOWER 7 BITS
DEC R1 ;FINISHED INPUTING MAX. CHARS ?
BEQ 12$ ;BR IF DONE MAX CHARACTERS INPUT
CMP #3,R3 ;TEST IF CHAR WAS A CTRL C ?
BNE 11$ ;BR IF NOT
JMP $EOP ;REPORT END OF PASS AND START OVER
11$: CMP #16,R3 ;TEST FOR SHIFT OUT CODE
BNE 4$ ;BR IF NOT
INC SHIFTO ;SET SHIFT OUT FLAG
MOV #DNOP,ECHJMP ;NOP THE BYPASS DISP. JMP
4$: TST SHIFTO ;TEST IF SHIFT OUT
BEQ 2$ ;BR IF NOT
CMP #17,R3 ;TEST FOR SHIFT IN CODE
BNE 5$ ;BR IF NOT
CLR SHIFTO ;CLEAR SHIFT OUT FLAG
MOV #161010,ECHJMP ;LOAD BYPASS DISP. JMP
5$: CMPB #37,R3 ;TEST IF TOO BIG
BPL 2$ ;BR IF NOT
BIC #177740,R3 ;MASK OFF BITS
2$: MOVB R3,(R0)+ ;LOAD CHARACTER INTO NEXT BUFFER LOC.
MOV #ECODEV,R2 ;LOAD POINTER TO ASCII CHARACTER VALUE WAS = ''
JSR PC,KBCHR ;CONVERT CHARACTER VALUE TO OCTAL
RTI ;RETURN TO WAIT
12$: CMP (SP)+,(SP)+ ;ADJUST STACK
MOV DPC,R2 ;GET DPC ADDRESS
BIS #BIT7,12(R2) ;EXTERNAL STOP TO DISPLAY
BR 20$ ;CLEAR THE BUFFER AND START AGAIN

```

| | | | | | | | | |
|------|--------|--------|--------|--------|-------|---------|---|---------------------------------|
| 1108 | | | | | | .SBTTL | SUBROUTINE FOR VERT. LIGHT PEN FIELD OF VIEW | |
| 1109 | 007162 | 012701 | 000030 | | | LOADAC: | MOV #24.,R1 | :LOAD COUNT |
| 1110 | 007166 | 012720 | 130000 | | | | MOV #RELATP,(R0)+ | :LOAD RELATIVE POINT |
| 1111 | 007172 | 012720 | 040004 | | | 1\$: | MOV #INTX+4,(R0)+ | :LOAD INTEN BIT |
| 1112 | 007176 | 005301 | | | | | DEC R1 | :FINISHED ? |
| 1113 | 007200 | 001374 | | | | | BNE 1\$ | :BR IF NOT |
| 1114 | 007202 | 000207 | | | | | RTS PC | :EXIT |
| 1115 | | | | | | | | |
| 1116 | | | | | | .SBTTL | SUBROUTINE FOR HORIZ. LIGHT PEN FIELD OF VIEW | |
| 1117 | 007204 | 012737 | 000030 | 007670 | | LOADUP: | MOV #24.,CNTR | :LOAD COUNT |
| 1118 | 007212 | 004737 | 007162 | | | 1\$: | JSR PC,LOADAC | :LOAD ACCROSS |
| 1119 | 007216 | 012720 | 110000 | | | | MOV #LONGV,(R0)+ | :LOAD LONG VECTOR |
| 1120 | 007222 | 012720 | 000004 | | | | MOV #4,(R0)+ | :LOAD VECTOR OVER |
| 1121 | 007226 | 012720 | 020140 | | | | MOV #MINUSX+140,(R0)+ | :AND UP |
| 1122 | 007232 | 005337 | 007670 | | | | DEC CNTR | |
| 1123 | 007236 | 001365 | | | | | BNE 1\$ | :BR IF NOT DONE |
| 1124 | 007240 | 000207 | | | | | RTS PC | :EXIT |
| 1125 | | | | | | | | |
| 1126 | | | | | | .SBTTL | LIGHT-PEN INTERRUPT SERVICE | |
| 1127 | 007242 | 017737 | 172040 | 001126 | | RET14: | MOV @VSCONS,\$BDDAT | :READ CONSOLE STATUS REG |
| 1128 | 007250 | 017737 | 172014 | 007660 | | | MOV @XPOS,DSAVE2 | :READ X POSITION |
| 1129 | 007256 | 017737 | 172010 | 007662 | | | MOV @YPOS,DSAVE3 | :READ Y POSITION |
| 1130 | 007264 | 042737 | 176000 | 007660 | | | BIC #176000,DSAVE2 | :MASK HIGH SIX BITS |
| 1131 | 007272 | 042737 | 176000 | 007662 | | | BIC #176000,DSAVE3 | |
| 1132 | 007300 | 005037 | 007646 | | | | CLR 40\$ | :CLEAR SWITCH FLAG HAPPEN LOC. |
| 1133 | 007304 | 032737 | 000100 | 001126 | | | BIT #BIT6,\$BDDAT | :TEST IF CONSOLE #1 SWITCH FLAG |
| 1134 | 007312 | 001405 | | | | | BEQ 3\$ | :BR IF NOT |
| 1135 | 007314 | 012737 | 022534 | 022414 | | | MOV #PENOF1,MS1PEN | :INFORM PEN OF ON #1 SET |
| 1136 | 007322 | 005237 | 007646 | | | | INC 40\$ | :SET SW HAPPENED FLAG |
| 1137 | 007326 | 032737 | 000200 | 001126 | 3\$: | | BIT #BIT7,\$BDDAT | :TEST IF CONSOLE #1 SWITCH FLAG |
| 1138 | 007334 | 001405 | | | | | BEQ 4\$ | :BR IF NOT |
| 1139 | 007336 | 012737 | 022564 | 022414 | | | MOV #PENON1,MS1PEN | :INFORM PEN ON #1 SET |
| 1140 | 007344 | 005237 | 007646 | | | | INC 40\$ | :SET SW HAPPENED FLAG |
| 1141 | 007350 | 032737 | 010000 | 001126 | 4\$: | | BIT #BIT12,\$BDDAT | :TEST IF CONSOLE #0 SWITCH FLAG |
| 1142 | 007356 | 001405 | | | | | BEQ 5\$ | :BR IF NOT |
| 1143 | 007360 | 012737 | 022454 | 022344 | | | MOV #PENOF0,MSOPEN | :INFORM PEN OFF #0 SET |
| 1144 | 007366 | 005237 | 007646 | | | | INC 40\$ | :SET SW HAPPENED FLAG |
| 1145 | 007372 | 032737 | 020000 | 001126 | 5\$: | | BIT #BIT13,\$BDDAT | :TEST IF CONSOLE #0 SET |
| 1146 | 007400 | 001405 | | | | | BEQ 6\$ | :BR IF NOT |
| 1147 | 007402 | 012737 | 022504 | 022344 | | | MOV #PENON0,MSOPEN | :INFORM PEN ON #0 SET |
| 1148 | 007410 | 005237 | 007646 | | | | INC 40\$ | :SET SW HAPPENED FLAG |
| 1149 | 007414 | 005737 | 007646 | | 6\$: | | TST 40\$ | :TEST IF SWITCH FUNCTION |
| 1150 | 007420 | 001003 | | | | | BNE 12\$ | :BR IF YES |
| 1151 | 007422 | 005737 | 007650 | | | | TST VIEW | :TEST IF FIELD OF VIEW |
| 1152 | 007426 | 001043 | | | | | BNE 20\$ | :BR IF YES |
| 1153 | 007430 | 032737 | 040000 | 001126 | 12\$: | | BIT #BIT14,\$BDDAT | :TEST IF PEN FLAG #0 SET |
| 1154 | 007436 | 001414 | | | | | BEQ 7\$ | :BR IF NOT |
| 1155 | 007440 | 013703 | 007660 | | | | MOV DSAVE2,R3 | :LOAD R3 |
| 1156 | 007444 | 012702 | 022320 | | | | MOV #DLT14A+4,R2 | :LOAD ADDRESS |
| 1157 | 007450 | 004737 | 010014 | | | | JSR PC,KBCHR | :LOAD X READOUT |
| 1158 | 007454 | 013703 | 007662 | | | | MOV DSAVE3,R3 | :LOAD R3 |
| 1159 | 007460 | 012702 | 022332 | | | | MOV #DLT14B+4,R2 | :LOAD ADDRESS |
| 1160 | 007464 | 004737 | 010014 | | | | JSR PC,KBCHR | :LOAD Y READOUT |
| 1161 | 007470 | 032737 | 000400 | 001126 | 7\$: | | BIT #BIT8,\$BDDAT | :TEST IF PEN #1 FLAG |
| 1162 | 007476 | 001414 | | | | | BEQ 10\$ | :BR IF NOT |
| 1163 | 007500 | 013703 | 007660 | | | | MOV DSAVE2,R3 | :GET X VALUE |

| | | | | | | | | |
|------|--------|--------|---------------|-------|---------|---------------------|--|--|
| 1164 | 007504 | 012702 | 022370 | | MOV | #DLT14C+4,R2 | | :LOAD POINTER |
| 1165 | 007510 | 004737 | 010014 | | JSR | PC,KBCHR | | :CONVERT TO ASCII |
| 1166 | 007514 | 013703 | 007662 | | MOV | DSAVE3,R3 | | :GET Y VALUE |
| 1167 | 007520 | 012702 | 022402 | | MOV | #DLT14D+4,R2 | | :LOAD POINTER |
| 1168 | 007524 | 004737 | 010014 | | JSR | PC,KBCHR | | :CONVERT TO ASCII |
| 1169 | 007530 | 012716 | 007716 | 10\$: | MOV | #RESUME,(SP) | | :FIX RETURN PC... |
| 1170 | 007534 | 000002 | | | RTI | | | :...AND RESUME. |
| 1171 | | | | | | | | |
| 1172 | | | | | | | | :COME HERE IF LIGHT-PEN HIT DURING THE FIELD OF VIEW SUB-PICTURE |
| 1173 | | | | | | | | |
| 1174 | 007536 | 005237 | 007652 | 20\$: | INC | HITCNT | | :UPDATE COUNT |
| 1175 | 007542 | 013703 | 007652 | | MOV | HITCNT,R3 | | :LOAD COUNT # |
| 1176 | 007546 | 012702 | 022450 | | MOV | #FRM16B,R2 | | :LOAD MESSAGE POINTER |
| 1177 | 007552 | 004737 | 010014 | | JSR | PC,KBCHR | | :CONVERT TO ASCII |
| 1178 | 007556 | 005001 | | | CLR | R1 | | |
| 1179 | 007560 | 005002 | | | CLR | R2 | | |
| 1180 | 007562 | 013700 | 007660 | | MOV | DSAVE2,R0 | | :GET X AXIS |
| 1181 | 007566 | 162700 | 001400 | | SUB | #1400,R0 | | :GET A BASE ADDRESS |
| 1182 | 007572 | 006200 | | | ASR | R0 | | |
| 1183 | 007574 | 006200 | | | ASR | R0 | | |
| 1184 | 007576 | 001404 | | | BEQ | 30\$ | | |
| 1185 | 007600 | 062701 | 000070 | 21\$: | ADD | #70,R1 | | :UPDATE OFFSET |
| 1186 | 007604 | 005300 | | | DEC | R0 | | |
| 1187 | 007606 | 001374 | | | BNE | 21\$ | | :BR UNTIL DONE |
| 1188 | 007610 | 013700 | 007662 | 30\$: | MOV | DSAVE3,R0 | | :GET Y AXIS |
| 1189 | 007614 | 162700 | 000304 | | SUB | #304,R0 | | :MAKE BASE ADDRESS |
| 1190 | 007620 | 006200 | | | ASR | R0 | | |
| 1191 | 007622 | 006200 | | | ASR | R0 | | :SHIFT RIGHT |
| 1192 | 007624 | 001404 | | | BEQ | 32\$ | | |
| 1193 | 007626 | 062701 | 000002 | 31\$: | ADD | #2,R1 | | |
| 1194 | 007632 | 005300 | | | DEC | R0 | | |
| 1195 | 007634 | 001374 | | | BNE | 31\$ | | |
| 1196 | 007636 | 042761 | 040000 026256 | 32\$: | BIC | #INTX,BUFFER+10(R1) | | :CLEAR THE BIT |
| 1197 | 007644 | 000731 | | | BR | 10\$ | | |
| 1198 | | | | | | | | |
| 1199 | 007646 | 000000 | | 40\$: | 0 | | | |
| 1200 | | | | | | | | |
| 1201 | 007650 | | | | SHIFTO: | | | |
| 1202 | 007650 | | | | DIDINT: | | | |
| 1203 | 007650 | 000000 | | | VIEW: | 0 | | |
| 1204 | 007652 | 000000 | | | HITCNT: | 0 | | |
| 1205 | 007654 | 000000 | | | DSAVE: | 0 | | |
| 1206 | 007656 | 000000 | | | DSAVE1: | 0 | | |
| 1207 | 007660 | 000000 | | | DSAVE2: | 0 | | |
| 1208 | 007662 | 000000 | | | DSAVE3: | 0 | | |
| 1209 | 007664 | 000000 | | | HOLD: | 0 | | |
| 1210 | 007666 | 000000 | | | TSAVE: | 0 | | |
| 1211 | 007670 | 000000 | | | CNTR: | 0 | | |
| 1212 | 007672 | 000000 | | | TEMPA: | 0 | | |
| 1213 | 007674 | 000000 | | | TEMPB: | 0 | | |

```
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222 007676 012537 010006  
1223 007702 012537 010010  
1224 007706 013777 010010 171350  
1225 007714 000403  
1226 007716 012777 000001 171340  
1227 007724 005037 177776  
1228 007730 000001  
1229 007732 000774  
1230  
1231  
1232  
1233 007734 005337 010006  
1234 007740 001403  
1235 007742 012716 007716  
1236 007746 000002  
1237  
1238 007750 105777 171164  
1239 007754 100410  
1240 007756 005037 010012  
1241 007762 032777 001000 171150  
1242 007770 001402  
1243 007772 005137 010012  
1244 007776 012716 010004  
1245 010002 000002  
1246 010004 000205  
1247  
1248 010006 000000  
1249 010010 000000  
1250 010012 000000
```

```
          .SBTTL DISPLAY SUB-ROUTINE  
          :  
          : ARGUMENTS ARE LOOP COUNT AND DISPLAY BUFFER ADDRESS  
          : UPON INTERRUPT , DEC LOOP COUNT  
          : RESUME DISPLAY IF NOT 0  
          : RTS R5 IF COMPLETED  
          :  
          : DSPLA: MOV      (R5)+,COUNT      ; ITERATION COUNT.  
          :          MOV      (R5)+,FILE      ; DISPLAY FILE ADDRESS.  
          :          MOV      FILE,@DPC      ; START DISPLAY  
          :          BR        .+10  
          : RESUME: MOV      #1,@DPC      ; RESUME DISPLAY.  
          : 1$:   CLR      PSW  
          :          WAIT  
          :          BR        1$          ; LOOP BACK  
          :  
          : RETURN HERE UPON STOP INTERRUPT  
          :  
          : STOPI: DEC      COUNT          ; FINISHED LOOPING ?  
          :          BEQ      1$          ; BR IF SO.  
          :          MOV      #RESUME,(SP) ; RESUME...  
          :          RTI  
          :          ;...OTHERWISE.  
          :  
          : 1$:   TSTB     @SWR          ; KEYBOARD CONTROL ??  
          :          BMI     2$          ; EXIT IF SO.  
          :          CLR     SWITCH  
          :          BIT     #BIT9,@SWR ; TEST SWITCH BIT 9  
          :          BEQ     2$  
          :          COM     SWITCH      ; SET FLAG IF SWR 9 = 1  
          :          MOV     #3$,(SP)    ; FIX RETURN PC...  
          :          RTI  
          :          ;...AND...  
          :          ;...RETURN TO CALLER.  
          :  
          : COUNT: 0  
          : FILE: 0  
          : SWITCH: 0
```

1252
1253
1254
1255
1256
1257
1258 010014 004737 010054
1259 010020 110442
1260 010022 004737 010046
1261 010026 110442
1262 010030 004737 010046
1263 010034 110442
1264 010036 004737 010046
1265 010042 110442
1266 010044 000207
1267 010046 006003
1268 010050 006003
1269 010052 006003
1270 010054 010304
1271 010056 042704 177770
1272 010062 062704 000060
1273 010066 000207
1274
1275 010070 000000
1276 010072 000000
1277

.SBTTL UPDATE OCTAL READOUT OF THE X-Y FOR LIGHT PEN HIT
:
: CALL: MOV VAL,R3 : VALUE TO ENCODE.
: MOV ADDR,R2 : ADDRESS OF LO ORDER CHAR +1.
: JSR PC,KBCHR : ENCODE 4 DIGIT OCTAL ASCII.
:
KBCHR: JSR PC,10\$:LOAD BITS
:MOVB R4,-(R2) :SAVE BITS
:JSR PC,11\$:MOVE BITS
:MOVB R4,-(R2) :SAVE BITS
:JSR PC,11\$:MOVE BITS
:MOVB R4,-(R2) :SAVE BITS
:JSR PC,11\$:MOVE BITS
:MOVB R4,-(R2) :SAVE BITS
:RTS PC
11\$: ROR R3
:ROR R3
:ROR R3
10\$: MOV R3,R4 :LOAD R4
:BIC #177770,R4 :MASK BITS
:ADD #60,R4 :MAKE A NUMBER
:RTS PC
DELTX6: 0
DELTY6: 0

```
1279 .SBTTL X - Y POSITIONS FOR THE SHORT TERM DRIFT TEST
1280
1281 010074 000000 000000 000000 TABA: .WORD 0,0,0,0,0,0
1282 010110 001000 001000 .WORD 1000,1000
1283 010114 000000 001777 000000 .WORD 0,MAXY,0,MAXY,0,MAXY
1284 010130 001000 001000 .WORD 1000,1000
1285 010134 001777 001777 001777 .WORD MAXX,MAXY,MAXX,MAXY,MAXX,MAXY
1286 010150 001000 001000 .WORD 1000,1000
1287 010154 001777 000000 001777 .WORD MAXX,0,MAXX,0,MAXX,0
1288 010170 001000 001000 .WORD 1000,1000
1289 010174 100000 BIT15
1308
1309
```

| | | | | |
|------|--------|--------|--------|--------|
| 1321 | 010176 | 117000 | | |
| 1322 | 010200 | 000000 | | |
| 1323 | 010202 | 001600 | | |
| 1324 | 010204 | 164300 | | |
| 1325 | 010206 | 164700 | | |
| 1326 | 010210 | 100000 | | |
| 1327 | 010212 | 051526 | 030066 | 053040 |
| 1328 | 010256 | 015 | 012 | 012 |
| 1329 | 010261 | 040 | 020040 | 044504 |
| 1330 | 010312 | 015 | 012 | 012 |
| 1331 | 010315 | 101 | 036440 | 030040 |
| 1332 | 010345 | 015 | 012 | |
| 1333 | 010347 | 102 | 036440 | 030040 |
| 1334 | 010410 | 015 | 012 | |
| 1335 | 010412 | 020103 | 020075 | 031460 |
| 1336 | 010443 | 015 | 012 | |
| 1337 | 010445 | 104 | 036440 | 030040 |
| 1338 | 010475 | 015 | 012 | |
| 1339 | 010477 | 105 | 036440 | 030040 |
| 1340 | 010531 | 015 | 012 | |
| 1341 | 010533 | 106 | 036440 | 030040 |
| 1342 | 010566 | 015 | 012 | |
| 1343 | 010570 | 020107 | 020075 | 033460 |
| 1344 | 010613 | 015 | 012 | |
| 1345 | 010615 | 110 | 036440 | 030440 |
| 1346 | 010652 | 015 | 012 | |
| 1347 | 010654 | 020111 | 020075 | 030461 |
| 1348 | 010721 | 015 | 012 | |
| 1349 | 010723 | 112 | 036440 | 030440 |
| 1350 | 010767 | 015 | 012 | |
| 1351 | 010771 | 113 | 036440 | 030440 |
| 1352 | 011021 | 015 | 012 | |
| 1353 | 011023 | 114 | 036440 | 030440 |
| 1354 | 011101 | 015 | 012 | |
| 1355 | 011103 | 115 | 036440 | 030440 |
| 1356 | 011205 | 015 | 012 | |
| 1357 | 011207 | 116 | 036440 | 030440 |
| 1358 | 011262 | 015 | 012 | |
| 1359 | 011264 | 020117 | 020075 | 033461 |
| 1360 | 011321 | 015 | 012 | |
| 1361 | 011323 | 120 | 036440 | 031040 |
| 1362 | 011351 | 015 | 012 | |
| 1363 | 011353 | 121 | 036440 | 031040 |
| 1364 | 011414 | 015 | 012 | |
| 1365 | 011416 | 020122 | 020075 | 031062 |
| 1366 | 011455 | 015 | 012 | |
| 1367 | 011457 | 123 | 036440 | 031040 |
| 1368 | 011530 | 015 | 012 | |
| 1369 | 011532 | 020124 | 020075 | 032062 |
| 1370 | 011603 | 015 | 012 | |
| 1371 | 011605 | 125 | 036440 | 031040 |
| 1372 | 011661 | 015 | 012 | |
| 1373 | 011663 | 126 | 036440 | 031040 |
| 1374 | 011730 | 015 | 012 | |
| 1375 | 011732 | 020127 | 020075 | 033462 |
| 1376 | 011775 | 015 | 012 | 012 |

```
FRME0: POINT!INT4
0
MAXY-177
CONSL0!BIT7!BIT6           :ENABLE CONSOLE #0
CONSL1!BIT7!BIT6           :ENABLE CONSOLE #1
CHAR
.ASCII /VS60 VISUAL DISPLAY TEST < CZVSDC >/
.BYTE 15,12,12
.ASCII / DIRECTORY OF THE TESTS/
.BYTE 15,12,12
.ASCII /A = 01 = DIRECTORY FRAME/
.BYTE 15,12
.ASCII /B = 02 = ASTIGMATISM AND SETTling/
.BYTE 15,12
.ASCII /C = 03 = SHORT TERM DRIFT/
.BYTE 15,12
.ASCII /D = 04 = MINOR AXIS GAIN/
.BYTE 15,12
.ASCII /E = 05 = MAJOR AXIS OFFSET/
.BYTE 15,12
.ASCII /F = 06 = VECTOR LENGTH GAIN/
.BYTE 15,12
.ASCII /G = 07 = PINCUSHION/
.BYTE 15,12
.ASCII /H = 10 = OCTAGONS AND CIRCLES/
.BYTE 15,12
.ASCII /I = 11 = SCISSORING AND VECTOR SCALES/
.BYTE 15,12
.ASCII /J = 12 = X AND Y DYNAMIC OFFSET TEST/
.BYTE 15,12
.ASCII /K = 13 = CHARACTER SCALE/
.BYTE 15,12
.ASCII /L = 14 = CHARACER QUALITY AND CHARACTER ROTATE/
.BYTE 15,12
.ASCII /M = 15 = CHARACTER SET, SUPER AND SUBSCRIPT, AND ITALIC CHARACTERS/
.BYTE 15,12
.ASCII /N = 16 = SYNC SPEED AND CHARACTER TERMINATE/
.BYTE 15,12
.ASCII /O = 17 = DASH LINES AND BLINK/
.BYTE 15,12
.ASCII /P = 20 = VECTOR LENGTH/
.BYTE 15,12
.ASCII /Q = 21 = HORIZONTAL PHOSPHOR TEST/
.BYTE 15,12
.ASCII /R = 22 = VERTICAL PHOSPHOR TEST/
.BYTE 15,12
.ASCII /S = 23 = SHORT VECTORS AND RELATIVE POINT/
.BYTE 15,12
.ASCII /T = 24 = GRAPHPLOT X AND GRAPHPLOT Y TEST/
.BYTE 15,12
.ASCII /U = 25 = INTENSITY LEVEL AND LIGHT PEN TESTS/
.BYTE 15,12
.ASCII /V = 26 = KEYBOARD CHARACTER ECHO LOOP/
.BYTE 15,12
.ASCII /W = 27 = DYNAMIC EXTERNAL STOP TEST/
.BYTE 15,12,12
```

| | | | | | |
|------|--------|--------|--------|--------|--|
| 1377 | 012000 | 020040 | 052522 | 047502 | .ASCII / RUBOUT (DELETE) TO LOOP ON CURRENT PATTERN/ .BYTE 15,12 |
| 1378 | 012054 | 015 | 012 | | .ASCII / CR TO STOP FRAME MOTION, SPACE TO RESUME/ .ASCIIZ <15><12> |
| 1379 | 012056 | 020040 | 051103 | 052040 | .=-1 .EVEN |
| 1380 | 012130 | 005015 | 000 | | CONSL1!BIT7 ;DISABLE CONSOLE #1 |
| 1381 | | 012132 | | | CHAR |
| 1382 | | | | | .ASCIIZ / THIS IS CONSOLE 0/<15><12> |
| 1383 | 012132 | 164600 | | | .=-1 .EVEN |
| 1384 | 012134 | 100000 | | | CONSL0!BIT7 ;DISABLE CONSOLE #0 |
| 1385 | 012136 | 020040 | 044124 | 051511 | CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1 |
| 1386 | | 012163 | | | CHAR |
| 1387 | | 012164 | | | .ASCIIZ / THIS IS CONSOLE 1/<15><12> |
| 1388 | 012164 | 164200 | | | .=-1 .EVEN |
| 1389 | 012166 | 164700 | | | CONSLC!BIT7!BIT6 ; ENABLE CONSOLE #0 |
| 1390 | 012170 | 100000 | | | DSTOP |
| 1391 | 012172 | 020040 | 044124 | 051511 | DJMP |
| 1392 | | 012217 | | | FRME0 |
| 1393 | | 012220 | | | .SBTTL X AND Y POSITIONS FOR THE SETTLING TEST |
| 1394 | 012220 | 164300 | | | TABB: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1 |
| 1395 | 012222 | 173400 | | | POINT |
| 1396 | 012224 | 160000 | | | INTX!BIT9 |
| 1397 | 012226 | 010176 | | | BIT9 |
| 1398 | | | | | INTX!BIT8 |
| 1399 | | | | | BIT8 |
| 1400 | | | | | INTX!BIT7 |
| 1401 | 012230 | 164700 | | | BIT7 |
| 1402 | 012232 | 114000 | | | INTX!BIT6 |
| 1403 | 012234 | 041000 | | | BIT6 |
| 1404 | 012236 | 001000 | | | INTX!BIT5 |
| 1405 | 012240 | 040400 | | | BIT5 |
| 1406 | 012242 | 000400 | | | INTX!BIT4 |
| 1407 | 012244 | 040200 | | | BIT4 |
| 1408 | 012246 | 000200 | | | INTX!BIT3 |
| 1409 | 012250 | 040100 | | | BIT3 |
| 1410 | 012252 | 000100 | | | INTX!BIT2 |
| 1411 | 012254 | 040040 | | | BIT2 |
| 1412 | 012256 | 000040 | | | INTX!BIT1 |
| 1413 | 012260 | 040020 | | | BIT1 |
| 1414 | 012262 | 000020 | | | INTX!BIT0 |
| 1415 | 012264 | 040010 | | | BIT0 |
| 1416 | 012266 | 000010 | | | INTX |
| 1417 | 012270 | 040004 | | | 0 |
| 1418 | 012272 | 000004 | | | |
| 1419 | 012274 | 040002 | | | INTX!1400 |
| 1420 | 012276 | 000002 | | | BIT8 |
| 1421 | 012300 | 040001 | | | INTX!1600 |
| 1422 | 012302 | 000001 | | | BIT7 |
| 1423 | 012304 | 040000 | | | INTX!1700 |
| 1424 | 012306 | 000000 | | | BIT6 |
| 1425 | | | | | INTX!1740 |
| 1426 | 012310 | 041400 | | | |
| 1427 | 012312 | 000400 | | | |
| 1428 | 012314 | 041600 | | | |
| 1429 | 012316 | 000200 | | | |
| 1430 | 012320 | 041700 | | | |
| 1431 | 012322 | 000100 | | | |
| 1432 | 012324 | 041740 | | | |

| | | | |
|------|--------|--------|-----------|
| 1433 | 012326 | 000040 | BIT5 |
| 1434 | 012330 | 041760 | INTX!1760 |
| 1435 | 012332 | 000020 | BIT4 |
| 1436 | 012334 | 041770 | INTX!1770 |
| 1437 | 012336 | 000010 | BIT3 |
| 1438 | 012340 | 041774 | INTX!1774 |
| 1439 | 012342 | 000004 | BIT2 |
| 1440 | 012344 | 041776 | INTX!1776 |
| 1441 | 012346 | 000002 | BIT1 |
| 1442 | 012350 | 041777 | INTX!1777 |
| 1443 | 012352 | 000001 | BIT0 |
| 1444 | | | |
| 1445 | 012354 | 041400 | INTX!1400 |
| 1446 | 012356 | 001400 | 1400 |
| 1447 | 012360 | 041600 | INTX!1600 |
| 1448 | 012362 | 001600 | 1600 |
| 1449 | 012364 | 041700 | INTX!1700 |
| 1450 | 012366 | 001700 | 1700 |
| 1451 | 012370 | 041740 | INTX!1740 |
| 1452 | 012372 | 001740 | 1740 |
| 1453 | 012374 | 041760 | INTX!1760 |
| 1454 | 012376 | 001760 | 1760 |
| 1455 | 012400 | 041770 | INTX!1770 |
| 1456 | 012402 | 001770 | 1770 |
| 1457 | 012404 | 041774 | INTX!1774 |
| 1458 | 012406 | 001774 | 1774 |
| 1459 | 012410 | 041776 | INTX!1776 |
| 1460 | 012412 | 001776 | 1776 |
| 1461 | 012414 | 041777 | INTX!1777 |
| 1462 | 012416 | 001777 | 1777 |
| 1463 | | | |
| 1464 | 012420 | 040400 | INTX!BIT8 |
| 1465 | 012422 | 001400 | 1400 |
| 1466 | 012424 | 040200 | INTX!BIT7 |
| 1467 | 012426 | 001600 | 1600 |
| 1468 | 012430 | 040100 | INTX!BIT6 |
| 1469 | 012432 | 001700 | 1700 |
| 1470 | 012434 | 040040 | INTX!BIT5 |
| 1471 | 012436 | 001740 | 1740 |
| 1472 | 012440 | 040020 | INTX!BIT4 |
| 1473 | 012442 | 001760 | 1760 |
| 1474 | 012444 | 040010 | INTX!BIT3 |
| 1475 | 012446 | 001770 | 1770 |
| 1476 | 012450 | 040004 | INTX!BIT2 |
| 1477 | 012452 | 001774 | 1774 |
| 1478 | 012454 | 040002 | INTX!BIT1 |
| 1479 | 012456 | 001776 | 1776 |
| 1480 | 012460 | 040001 | INTX!BIT0 |
| 1481 | 012462 | 001777 | 1777 |
| 1482 | 012464 | 173400 | DSTOP |
| 1483 | 012466 | 160000 | DJMP |
| 1484 | 012470 | 012230 | TABB |

;RETURN ADDRESS

1486
1487
1488
1489 012472 164700
1490
1491 012474 114000
1492 012476 040000
1493 012500 000000
1494 012502 040000
1495 012504 000000
1496 012506 040000
1497 012510 000000
1498 012512 041777
1499 012514 000000
1500 012516 041777
1501 012520 000000
1502 012522 041777
1503 012524 000000
1504 012526 041777
1505 012530 001777
1506 012532 041777
1507 012534 001777
1508 012536 041777
1509 012540 001777
1510 012542 040000
1511 012544 001777
1512 012546 040000
1513 012550 001777
1514 012552 040000
1515 012554 001777
1516
1517 012556 114000
1518 012560 000000
1519 012562 000000
1520 012564 110000
1521 012566 041777
1522 012570 000000
1523 012572 040000
1524 012574 001777
1525 012576 061777
1526 012600 000000
1527 012602 040000
1528 012604 021777
1529
1530 012606 060000
1531 012610 001777
1532 012612 041777
1533 012614 020000
1534 012616 060000
1535 012620 021777
1536 012622 061777
1537 012624 020000
1538
1539 012626 041777
1540 012630 001777
1541 012632 061777

```
;FILE 2 <ANALOG TUNE-UP TEST >
FRME2:  CONSL1!BIT7!BIT6           ;ENABLE CONSOLE #1
;INTENSIFY A POINT 3 TIMES IN EACH CORNER
POINT                                     ;LOWER LEFT
INTX
0
INTX
0
INTX
0
INTX!MAXX                               ;LOWER RIGHT
0
INTX!MAXX
0
INTX!MAXX                               ;UPPER RIGHT
0
MAXY
INTX!MAXX
MAXY
INTX!MAXX
MAXY
INTX                                     ;UPPER LEFT
MAXY
INTX
MAXY
INTX
MAXY
;NOW DRAW THE OUTER REF. BOX
PICST0: POINT
0
0
LONGV
INTX!MAXX                               ; +X, +Y
0
INTX                                     ; +X, +Y
MAXY
INTX!MINUSX!MAXX                       ; -X, +Y
0
INTX                                     ; +X, -Y
MINUSY!MAXY
;NOW RE-DO THE BOX WITH NEGATIVE SIGN BITS
PICST1: INTX!MINUSX
MAXY
INTX!MAXX
MINUSY
INTX!MINUSX
MINUSX!MAXY
INTX!MINUSX!MAXX
MINUSY
;NOW DRAW LOWER LEFT TO UPPER RIGHT DIAG.
INTX!MAXX
MAXY
INTX!MINUSX!MAXX
```


| | | | | |
|------|--------|--------|---|--------------------------------------|
| 1542 | 012634 | 021777 | MINUSX!MAXY | |
| 1543 | | | ;REPOSITION TO LOWER RIGHT AND DRAW LOWER RIGHT | |
| 1544 | | | ; TO UPPER LEFT DIAG. | |
| 1545 | 012636 | 001777 | MAXX | |
| 1546 | 012640 | 000000 | 0 | |
| 1547 | 012642 | 061777 | INTX!MINUSX!MAXX | |
| 1548 | 012644 | 001777 | MAXY | |
| 1549 | 012646 | 041777 | INTX!MAXX | |
| 1550 | 012650 | 021777 | MINUSX!MAXY | |
| 1551 | | | .SBTTL MENU 1 SUB-PICTURE | |
| 1552 | | | ;DRAW REF. BOX IN THE MENU | |
| 1553 | 012652 | 170003 | DMENU1 | ;ENABLE MENU |
| 1554 | 012654 | 114000 | POINT | |
| 1555 | 012656 | 000000 | 0 | |
| 1556 | 012660 | 000000 | 0 | |
| 1557 | 012662 | 110000 | LONGV | ;DRAW REF. BOX |
| 1558 | 012664 | 040177 | INTX!177 | |
| 1559 | 012666 | 000000 | 0 | |
| 1560 | 012670 | 040000 | INTX | |
| 1561 | 012672 | 001777 | MAXY | |
| 1562 | 012674 | 060177 | INTX!MINUSX!177 | |
| 1563 | 012676 | 000000 | 0 | |
| 1564 | 012700 | 040000 | INTX | |
| 1565 | 012702 | 021777 | MINUSX!MAXY | |
| 1566 | | | ;NOW REVERSE THE DRAWING PROCEDURE | |
| 1567 | 012704 | 060000 | INTX!MINUSX | |
| 1568 | 012706 | 001777 | MAXY | |
| 1569 | 012710 | 040177 | INTX!177 | |
| 1570 | 012712 | 020000 | MINUSX | |
| 1571 | 012714 | 060000 | INTX!MINUSX | |
| 1572 | 012716 | 021777 | MINUSX!MAXY | |
| 1573 | 012720 | 060177 | INTX!MINUSX!177 | |
| 1574 | 012722 | 020000 | MINUSX | |
| 1575 | | | ;NOW DRAW THE DIAG. X IN THE MENU | |
| 1576 | 012724 | 040177 | INTX!177 | ;LOWER LEFT, IN MENU, TO UPPER RIGHT |
| 1577 | 012726 | 001777 | MAXY | |
| 1578 | 012730 | 060177 | INTX!MINUSX!177 | |
| 1579 | 012732 | 021777 | MINUSX!MAXY | |
| 1580 | 012734 | 000177 | 177 | ;REPOSITION TO LOWER LEFT OF MENU |
| 1581 | 012736 | 000000 | 0 | |
| 1582 | 012740 | 060177 | INTX!MINUSX!177 | ;LOWER RIGHT TO UPPER LEFT |
| 1583 | 012742 | 001777 | MAXY | |
| 1584 | 012744 | 040177 | INTX!177 | |
| 1585 | 012746 | 021777 | MINUSX!MAXY | |
| 1586 | 012750 | 170002 | DMENU0 | ;RETURN TO MAIN SCREEN |
| 1587 | | | ;CONTINUE MAIN SCREEN PICTURE | |
| 1588 | 012752 | 114000 | POINT | |
| 1589 | 012754 | 001400 | 1400 | |
| 1590 | 012756 | 001000 | 1000 | |
| 1591 | | | ;DRAW A 100 UNIT BOX, SAME METHOD AS OUTER REF. BOX | |
| 1592 | 012760 | 110000 | LONGV | |
| 1593 | 012762 | 040144 | INTX!100. | ; +X, +Y |
| 1594 | 012764 | 000000 | 0 | |
| 1595 | 012766 | 040000 | INTX | ; +X, +Y |
| 1596 | 012770 | 000144 | 100. | |
| 1597 | 012772 | 060144 | INTX!MINUSX!100. | ; -X, +Y |

```
1598 012774 000000 0
1599 012776 040000 INTX ; +X, -Y
1600 013000 020144 MINUSY!100.
1601 013002 040144 INTX!100. ; +X, -Y
1602 013004 020000 MINUSY
1603 013006 060000 INTX!MINUSX ; -X, +Y
1604 013010 000144 100.
1605 013012 060144 INTX!MINUSX!100. ; -X, -Y
1606 013014 020000 MINUSY
1607 013016 060000 INTX!MINUSX ; -X, -Y
1608 013020 020144 MINUSY!100.
1609 013022 040144 INTX!100.
1610 013024 000144 100.
1611 013026 060144 INTX!MINUSX!100.
1612 013030 020144 MINUSX!100.
1613 013032 000144 100.
1614 013034 000000 0
1615 013036 060144 INTX!MINUSX!100.
1616 013040 000144 100.
1617 013042 040144 INTX!100.
1618 013044 020144 MINUSX!100.
1619 ;DRAW A 10 UNIT BOX, SAME METHOD AS OUTER BOX
1620 013046 114000 POINT
1621 013050 001400 1400
1622 013052 000700 700
1623 013054 110000 LONGV
1624 013056 040012 INTX!10. ; +X, +Y
1625 013060 000000 0
1626 013062 040000 INTX ; +X, +Y
1627 013064 000012 10.
1628 013066 060012 INTX!MINUSX!10. ; -X, +Y
1629 013070 000000 0
1630 013072 040000 INTX ; +X, -Y
1631 013074 020012 MINUSY!10.
1632 013076 040012 INTX!10. ; +X, -Y
1633 013100 020000 MINUSY
1634 013102 060000 INTX!MINUSX ; -X, +Y
1635 013104 000012 10.
1636 013106 060012 INTX!MINUSX!10. ; -X, -Y
1637 013110 020000 MINUSY
1638 013112 060000 INTX!MINUSX ; -X, -Y
1639 013114 020012 MINUSY!10.
1640 013116 040012 INTX!10.
1641 013120 000012 10.
1642 013122 060012 INTX!MINUSX!10.
1643 013124 020012 MINUSX!10.
1644 013126 000012 10.
1645 013130 000000 0
1646 013132 060012 INTX!MINUSX!10.
1647 013134 000012 10.
1648 013136 040012 INTX!10.
1649 013140 020012 MINUSX!10.
1650 ;DRAW FOUR VECTORS FROM A 'COMMON' POINT WHICH WILL BE THE
1651 ;SUPERIMPOSED UPON BY THE NEXT SUB-PICTURE
1652 013142 114000 POINT
1653 013144 001000 1000
```

| | | | | |
|------|--------|--------|-----------------|--|
| 1654 | 013146 | 000400 | 400 | |
| 1655 | 013150 | 110000 | LONGV | |
| 1656 | 013152 | 040000 | INTX | |
| 1657 | 013154 | 000200 | 200 | |
| 1658 | 013156 | 114000 | POINT | |
| 1659 | 013160 | 001000 | 1000 | |
| 1660 | 013162 | 000400 | 400 | |
| 1661 | 013164 | 110000 | LONGV | |
| 1662 | 013166 | 040200 | INTX!200 | |
| 1663 | 013170 | 000000 | 0 | |
| 1664 | 013172 | 114000 | POINT | |
| 1665 | 013174 | 001000 | 1000 | |
| 1666 | 013176 | 000400 | 400 | |
| 1667 | 013200 | 110000 | LONGV | |
| 1668 | 013202 | 040000 | INTX | |
| 1669 | 013204 | 020200 | MINUSY!200 | |
| 1670 | 013206 | 114000 | POINT | |
| 1671 | 013210 | 001000 | 1000 | |
| 1672 | 013212 | 000400 | 400 | |
| 1673 | 013214 | 110000 | LONGV | |
| 1674 | 013216 | 060200 | INTX!MINUSX!200 | |
| 1675 | 013220 | 000000 | 0 | |
| 1676 | | | .SBTTL | DRAW 10 VERTICAL VECTORS IN THE LEFT CENTER AERA |
| 1677 | 013222 | 114000 | POINT | |
| 1678 | 013224 | 000200 | 200 | |
| 1679 | 013226 | 000740 | 740 | |
| 1691 | 013230 | 104000 | SHORTV | |
| (1) | 013232 | 040010 | INTX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013234 | 130000 | RELATP | |
| (1) | 013236 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013240 | 000002 | 2 | ;MOVE THE Y AXIS |
| (1) | 013242 | 104000 | SHORTV | |
| (1) | 013244 | 060010 | INTX!MINUSX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013246 | 130000 | RELATP | |
| (1) | 013250 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013252 | 000002 | 2 | ;MOVE THE Y AXIS |
| (1) | 013254 | 104000 | SHORTV | |
| (1) | 013256 | 040010 | INTX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013260 | 130000 | RELATP | |
| (1) | 013262 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013264 | 000002 | 2 | ;MOVE THE Y AXIS |
| (1) | 013266 | 104000 | SHORTV | |
| (1) | 013270 | 060010 | INTX!MINUSX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013272 | 130000 | RELATP | |
| (1) | 013274 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013276 | 000002 | 2 | ;MOVE THE Y AXIS |
| (1) | 013300 | 104000 | SHORTV | |
| (1) | 013302 | 040010 | INTX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013304 | 130000 | RELATP | |
| (1) | 013306 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013310 | 000002 | 2 | ;MOVE THE Y AXIS |
| (1) | 013312 | 104000 | SHORTV | |
| (1) | 013314 | 060010 | INTX!MINUSX!10 | ;DRAW A 8. UNIT VERTICAL VECTOR |
| (1) | 013316 | 130000 | RELATP | |
| (1) | 013320 | 040002 | INTX!2 | ;INTENSIFY A POINT 2 UNITS AWAY |
| (1) | 013322 | 000002 | 2 | ;MOVE THE Y AXIS |

```
(1) 013324 104000 SHORTV
(1) 013326 040010 INTX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 013330 130000 RELATP
(1) 013332 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 013334 000002 2 ;MOVE THE Y AXIS
(1) 013336 104000 SHORTV
(1) 013340 060010 INTX!MINUSX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 013342 130000 RELATP
(1) 013344 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 013346 000002 2 ;MOVE THE Y AXIS
(1) 013350 104000 SHORTV
(1) 013352 040010 INTX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 013354 130000 RELATP
(1) 013356 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 013360 000002 2 ;MOVE THE Y AXIS
(1) 013362 104000 SHORTV
(1) 013364 060010 INTX!MINUSX!10 ;DRAW A 8. UNIT VERTICAL VECTOR
(1) 013366 130000 RELATP
(1) 013370 040002 INTX!2 ;INTENSIFY A POINT 2 UNITS AWAY
(1) 013372 000002 2 ;MOVE THE Y AXIS
1692 .SBTTL DRAW THE DELAY INTENSITY SUB-PICTURE IN THE LEFT CENTER AERA
1693 : DRAW 8 VECTORS USING BASIC VECTOR INSTRUCTION AWAY FROM A COMMON POINT
1694 : BUT OFFSET BY ONE UNIT.
1695 : THE COMMON POINT X=353 AND Y =1003
1696 :
1697 000353 XQ6=353
1698 001003 YQ6=1003
1699 000040 LQ6=40
1700 013374 114000 POINT
1701 013376 000354 XQ6+1
1702 013400 001003 YQ6
1703 013402 120000 BASICV
1704 013404 042040 INTX!PATH0!LQ6
1705 013406 114000 POINT
1706 013410 000354 XQ6+1
1707 013412 001004 YQ6+1 ;VECTOR #1
1708 013414 120000 BASICV
1709 013416 046040 INTX!PATH1!LQ6
1710 013420 114000 POINT
1711 013422 000353 XQ6
1712 013424 001004 YQ6+1 ;VECTOR #2
1713 013426 120000 BASICV
1714 013430 052040 INTX!PATH2!LQ6
1715 013432 114000 POINT
1716 013434 000352 XQ6-1
1717 013436 001004 YQ6+1 ;VECTOR #3
1718 013440 120000 BASICV
1719 013442 056040 INTX!PATH3!LQ6
1720 013444 114000 POINT
1721 013446 000352 XQ6-1
1722 013450 001003 YQ6 ;VECTOR #4
1723 013452 120000 BASICV
1724 013454 062040 INTX!PATH4!LQ6
1725 013456 114000 POINT
1726 013460 000352 XQ6-1
1727 013462 001002 YQ6-1 ;VECTOR #5
```

| | | | | |
|------|--------|--------|----------------|------------|
| 1728 | 013464 | 120000 | BASICV | |
| 1729 | 013466 | 066040 | INTX!PATH5!LQ6 | |
| 1730 | 013470 | 114000 | POINT | |
| 1731 | 013472 | 000353 | XQ6 | |
| 1732 | 013474 | 001002 | YQ6-1 | :VECTOR #6 |
| 1733 | 013476 | 120000 | BASICV | |
| 1734 | 013500 | 072040 | INTX!PATH6!LQ6 | |
| 1735 | 013502 | 114000 | POINT | |
| 1736 | 013504 | 000354 | XQ6+1 | |
| 1737 | 013506 | 001002 | YQ6-1 | :VECTOR #7 |
| 1738 | 013510 | 120000 | BASICV | |
| 1739 | 013512 | 076040 | INTX!PATH7!LQ6 | |
| 1740 | | | | |
| 1741 | 013514 | 173400 | DSTOP | |
| 1742 | 013516 | 160000 | DJMP | |
| 1743 | 013520 | 012472 | FRME2 | |

1745
1746
1747
1748 013522 114000
1749 013524 000774
1750 013526 000764
1751 013530 164700
1771 013532 110000
(1) 013534 040007
(1) 013536 000000
(1) 013540 040007
(1) 013542 000007
(1) 013544 040000
(1) 013546 000007
(1) 013550 060007
(1) 013552 000007
(1) 013554 060007
(1) 013556 000000
(1) 013560 060007
(1) 013562 020007
(1) 013564 040000
(1) 013566 020007
(1) 013570 040007
(1) 013572 020007

.SBTTL OCTAGONS USING LONG AND ABSOLUTE VECTORS (WIDTHS OF 7,77,177,377 AND 52
.SBTTL CIRCLES USING ABSOLUTE VECTORS (WIDTHS OF 64., 128., AND 256.)
FRME3: POINT
774
764
CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
LONGV ;OCTOGON BY LENGTH OF 7
INTX+7
0
INTX+7
7
INTX
7
INTX!MINUSX+7
7
INTX!MINUSX+7
0
INTX!MINUSX+7
MINUSX+7
INTX
MINUSX+7
INTX+7
MINUSX+7

K 5

:CIRCLE 8 DEG. RADIUS OF 64

1773
1774
1775 013574 114000
1776 013576 001077
1777 013600 000777
1778 013602 144000
1779 013604 041076 001010
1780 013610 041075 001021
1781 013614 041071 001031
1782 013620 041065 001041
1783 013624 041060 001050
1784 013630 041052 001057
1785 013634 041043 001064
1786 013640 041033 001071
1787 013644 041023 001074
1788 013650 041012 001076
1789 013654 041001 001077
1790 013660 040771 001077
1791 013664 040761 001075
1792 013670 040750 001072
1793 013674 040740 001066
1794 013700 040731 001061
1795 013704 040722 001053
1796 013710 040714 001045
1797 013714 040707 001035
1798 013720 040704 001025
1799 013724 040701 001014
1800 013730 040700 001003
1801 013734 040700 000774
1802 013740 040701 000763
1803 013744 040704 000752
1804 013750 040707 000742
1805 013754 040714 000732
1806 013760 040722 000724
1807 013764 040731 000716
1808 013770 040740 000711
1809 013774 040750 000705
1810 014000 040760 000702
1811 014004 040771 000700
1812 014010 041001 000700
1813 014014 041012 000701
1814 014020 041023 000703
1815 014024 041033 000706
1816 014030 041043 000713
1817 014034 041052 000720
1818 014040 041060 000727
1819 014044 041065 000736
1820 014050 041071 000746
1821 014054 041075 000756
1822 014060 041076 000767
1823 014064 041077 000777
1824 014070 164000
1825 014072 164000
1826 014074 164000
1827 014076 164000
1828 014100 164000

POINT
1077
777
ABSVCT
.WORD INTX!1076,1010
.WORD INTX!1075,1021
.WORD INTX!1071,1031
.WORD INTX!1065,1041
.WORD INTX!1060,1050
.WORD INTX!1052,1057
.WORD INTX!1043,1064
.WORD INTX!1033,1071
.WORD INTX!1023,1074
.WORD INTX!1012,1076
.WORD INTX!1001,1077
.WORD INTX!771,1077
.WORD INTX!761,1075
.WORD INTX!750,1072
.WORD INTX!740,1066
.WORD INTX!731,1061
.WORD INTX!722,1053
.WORD INTX!714,1045
.WORD INTX!707,1035
.WORD INTX!704,1025
.WORD INTX!701,1014
.WORD INTX!700,1003
.WORD INTX!700,774
.WORD INTX!701,763
.WORD INTX!704,752
.WORD INTX!707,742
.WORD INTX!714,732
.WORD INTX!722,724
.WORD INTX!731,716
.WORD INTX!740,711
.WORD INTX!750,705
.WORD INTX!760,702
.WORD INTX!771,700
.WORD INTX!1001,700
.WORD INTX!1012,701
.WORD INTX!1023,703
.WORD INTX!1033,706
.WORD INTX!1043,713
.WORD INTX!1052,720
.WORD INTX!1060,727
.WORD INTX!1065,736
.WORD INTX!1071,746
.WORD INTX!1075,756
.WORD INTX!1076,767
.WORD INTX!1077,777

:ENABLE ABSOLUTE VECTOR MODE

DNOP
DNOP
DNOP
DNOP
DNOP

1829
1830
1831 014102 114000
1832 014104 001177
1833 014106 000777
1834 014110 144000
1835 014112 041176 001021
1836 014116 041172 001042
1837 014122 041164 001063
1838 014126 041154 001103
1839 014132 041141 001121
1840 014136 041125 001136
1841 014142 041107 001151
1842 014146 041067 001162
1843 014152 041047 001171
1844 014156 041025 001175
1845 014162 041003 001177
1846 014166 040763 001176
1847 014172 040741 001173
1848 014176 040720 001166
1849 014202 040700 001156
1850 014206 040661 001144
1851 014212 040644 001130
1852 014216 040630 001112
1853 014222 040617 001073
1854 014226 040610 001053
1855 014232 040603 001032
1856 014236 040600 001010
1857 014242 040600 000767
1858 014246 040603 000745
1859 014252 040610 000724
1860 014256 040617 000704
1861 014262 040630 000665
1862 014266 040644 000647
1863 014272 040661 000633
1864 014276 040700 000621
1865 014302 040720 000611
1866 014306 040741 000604
1867 014312 040763 000601
1868 014316 041003 000600
1869 014322 041025 000602
1870 014326 041047 000606
1871 014332 041067 000615
1872 014336 041107 000626
1873 014342 041125 000641
1874 014346 041141 000656
1875 014352 041154 000674
1876 014356 041164 000714
1877 014362 041172 000735
1878 014366 041176 000756
1879 014372 041177 000777
1880 014376 164000
1881 014400 164000
1882 014402 164000
1883 014404 164000
1884

:CIRCLE 8 DEG. RADIUS OF 128

POINT
1177
777
ABSVCT
.WORD INTX:1176,1021
.WORD INTX:1172,1042
.WORD INTX:1164,1063
.WORD INTX:1154,1103
.WORD INTX:1141,1121
.WORD INTX:1125,1136
.WORD INTX:1107,1151
.WORD INTX:1067,1162
.WORD INTX:1047,1171
.WORD INTX:1025,1175
.WORD INTX:1003,1177
.WORD INTX:763,1176
.WORD INTX:741,1173
.WORD INTX:720,1166
.WORD INTX:700,1156
.WORD INTX:661,1144
.WORD INTX:644,1130
.WORD INTX:630,1112
.WORD INTX:617,1073
.WORD INTX:610,1053
.WORD INTX:603,1032
.WORD INTX:600,1010
.WORD INTX:600,767
.WORD INTX:603,745
.WORD INTX:610,724
.WORD INTX:617,704
.WORD INTX:630,665
.WORD INTX:644,647
.WORD INTX:661,633
.WORD INTX:700,621
.WORD INTX:720,611
.WORD INTX:741,604
.WORD INTX:763,601
.WORD INTX:1003,600
.WORD INTX:1025,602
.WORD INTX:1047,606
.WORD INTX:1067,615
.WORD INTX:1107,626
.WORD INTX:1125,641
.WORD INTX:1141,656
.WORD INTX:1154,674
.WORD INTX:1164,714
.WORD INTX:1172,735
.WORD INTX:1176,756
.WORD INTX:1177,777

:DISPLAY IN ABSOLUTE VECTOR MODE

:CIRCLE 8 DEG. RADIUS OF 256

1885
1886 014406 114000
1887 014410 001377
1888 014412 000777
1889 014414 144000
1890 014416 041375 001043
1891 014422 041365 001106
1892 014426 041351 001147
1893 014432 041330 001207
1894 014436 041303 001244
1895 014442 041252 001275
1896 014446 041216 001323
1897 014452 041157 001345
1898 014456 041116 001362
1899 014462 041053 001373
1900 014466 041010 001377
1901 014472 040745 001376
1902 014476 040702 001377
1903 014502 040640 001354
1904 014506 040600 001335
1905 014512 040542 001311
1906 014516 040510 001261
1907 014522 040461 001226
1908 014526 040436 001167
1909 014532 040417 001127
1910 014536 040406 001064
1911 014542 040401 001021
1912 014546 040401 000756
1913 014552 040406 000713
1914 014556 040417 000651
1915 014562 040436 000610
1916 014566 040461 000552
1917 014572 040510 000516
1918 014576 040542 000466
1919 014602 040600 000442
1920 014606 040640 000423
1921 014612 040702 000410
1922 014616 040745 000401
1923 014622 041010 000400
1924 014626 041053 000404
1925 014632 041116 000415
1926 014636 041157 000432
1927 014642 041216 000454
1928 014646 041252 000502
1929 014652 041303 000533
1930 014656 041330 000570
1931 014662 041351 000630
1932 014666 041365 000671
1933 014672 041374 000734
1934 014676 041377 000777
1935 014702 164000
1936 014704 164000
1937 014706 164000
1938 014710 164000
1939 014712 114000
1940 014714 000740

POINT
1377
777
ABSVCT
.WORD INTX:1375,1043
.WORD INTX:1365,1106
.WORD INTX:1351,1147
.WORD INTX:1330,1207
.WORD INTX:1303,1244
.WORD INTX:1252,1275
.WORD INTX:1216,1323
.WORD INTX:1157,1345
.WORD INTX:1116,1362
.WORD INTX:1053,1373
.WORD INTX:1010,1377
.WORD INTX:745,1376
.WORD INTX:702,1367
.WORD INTX:640,1354
.WORD INTX:600,1335
.WORD INTX:542,1311
.WORD INTX:510,1261
.WORD INTX:461,1226
.WORD INTX:436,1167
.WORD INTX:417,1127
.WORD INTX:406,1064
.WORD INTX:401,1021
.WORD INTX:401,756
.WORD INTX:406,713
.WORD INTX:417,651
.WORD INTX:436,610
.WORD INTX:461,552
.WORD INTX:510,516
.WORD INTX:542,466
.WORD INTX:600,442
.WORD INTX:640,423
.WORD INTX:702,410
.WORD INTX:745,401
.WORD INTX:1010,400
.WORD INTX:1053,404
.WORD INTX:1116,415
.WORD INTX:1157,432
.WORD INTX:1216,454
.WORD INTX:1252,502
.WORD INTX:1303,533
.WORD INTX:1330,570
.WORD INTX:1351,630
.WORD INTX:1365,671
.WORD INTX:1374,734
.WORD INTX:1377,777
DNOP
DNOP
DNOP
DNOP
POINT
740

;ENABLE ABSOLUTE VECTOR MODE

| | | | | |
|------|--------|--------|-----------------|---------------------------|
| 1941 | 014716 | 000640 | 640 | |
| 1942 | 014720 | 110000 | LONGV | :OCTOGON BY LENGTH OF 77 |
| (1) | 014722 | 040077 | INTX+77 | |
| (1) | 014724 | 000000 | 0 | |
| (1) | 014726 | 040077 | INTX+77 | |
| (1) | 014730 | 000077 | 77 | |
| (1) | 014732 | 040000 | INTX | |
| (1) | 014734 | 000077 | 77 | |
| (1) | 014736 | 060077 | INTX!MINUSX+77 | |
| (1) | 014740 | 000077 | 77 | |
| (1) | 014742 | 060077 | INTX!MINUSX+77 | |
| (1) | 014744 | 000000 | 0 | |
| (1) | 014746 | 060077 | INTX!MINUSX+77 | |
| (1) | 014750 | 020077 | MINUSX+77 | |
| (1) | 014752 | 040000 | INTX | |
| (1) | 014754 | 020077 | MINUSX+77 | |
| (1) | 014756 | 040077 | INTX+77 | |
| (1) | 014760 | 020077 | MINUSX+77 | |
| 1943 | 014762 | 114000 | POINT | |
| 1944 | 014764 | 000700 | 700 | |
| 1945 | 014766 | 000500 | 500 | |
| 1946 | 014770 | 110000 | LONGV | :OCTOGON BY LENGTH OF 177 |
| (1) | 014772 | 040177 | INTX+177 | |
| (1) | 014774 | 000000 | 0 | |
| (1) | 014776 | 040177 | INTX+177 | |
| (1) | 015000 | 000177 | 177 | |
| (1) | 015002 | 040000 | INTX | |
| (1) | 015004 | 000177 | 177 | |
| (1) | 015006 | 060177 | INTX!MINUSX+177 | |
| (1) | 015010 | 000177 | 177 | |
| (1) | 015012 | 060177 | INTX!MINUSX+177 | |
| (1) | 015014 | 000000 | 0 | |
| (1) | 015016 | 060177 | INTX!MINUSX+177 | |
| (1) | 015020 | 020177 | MINUSX+177 | |
| (1) | 015022 | 040000 | INTX | |
| (1) | 015024 | 020177 | MINUSX+177 | |
| (1) | 015026 | 040177 | INTX+177 | |
| (1) | 015030 | 020177 | MINUSX+177 | |
| 1947 | 015032 | 114000 | POINT | |
| 1948 | 015034 | 000600 | 600 | |
| 1949 | 015036 | 000200 | 200 | |
| 1950 | 015040 | 110000 | LONGV | :OCTOGON BY LENGTH OF 377 |
| (1) | 015042 | 040377 | INTX+377 | |
| (1) | 015044 | 000000 | 0 | |
| (1) | 015046 | 040377 | INTX+377 | |
| (1) | 015050 | 000377 | 377 | |
| (1) | 015052 | 040000 | INTX | |
| (1) | 015054 | 000377 | 377 | |
| (1) | 015056 | 060377 | INTX!MINUSX+377 | |
| (1) | 015060 | 000377 | 377 | |
| (1) | 015062 | 060377 | INTX!MINUSX+377 | |
| (1) | 015064 | 000000 | 0 | |
| (1) | 015066 | 060377 | INTX!MINUSX+377 | |
| (1) | 015070 | 020377 | MINUSX+377 | |
| (1) | 015072 | 040000 | INTX | |
| (1) | 015074 | 020377 | MINUSX+377 | |


```
1995 015220 114000 OFFTST: POINT
1996 015222 010000 OFFT1: BIT12
1997 015224 010000 OFFT2: BIT12
1998 015226 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
1999 015230 117000 POINT!INT4
2000 015232 000400 400
2001 015234 000400 400
2002 015236 120000 BASICV
2003 015240 043000 INTX!PATH0!1000 ;DRAW A SQUARE
2004 015242 053000 INTX!PATH2!1000
2005 015244 063000 INTX!PATH4!1000
2006 015246 073000 INTX!PATH6!1000
2007 015250 173400 DSTOP
2008 015252 160000 DJMP
2009 015254 015220 OFFTST
2010
2011 .SBTTL SUPER AND SUBSCRIPT SUB-PICTURE
2012
2013 015256 114000 SUPPIC: POINT
2014 015260 000400 400
2015 015262 001000 1000
2016 015264 110000 LONGV
2017 015266 041000 INTX!1000 ;DRAW REF. LINE
2018 015270 000000 0
2019 015272 114000 POINT
2020 015274 000400 400
2021 015276 001000 1000
2022 015300 160000 DJMP ;BYPASS ROTATED REF. LINE
2023 015302 015326 SUPC1
2024 015304 114000 SUPCO: POINT
2025 015306 001000 1000
2026 015310 000400 400
2027 015312 110000 LONGV
2028 015314 040000 INTX
2029 015316 001000 1000
2030 015320 114000 PCINT
2031 015322 001000 1000
2032 015324 000400 400
2033 015326 154340 SUPC1: CHARS3 ;ENSURE MAX CHAR SIZE
2034 015330 170040 STATSA!ITAL0
2035 015332 100000 CHAR
2036 015334 162000 DJSR
2037 015336 015356 SUPSUB
2038 015340 170060 STATSA!ITAL1 ;SET ITALIC
2039 015342 162000 DJSR
2040 015344 015356 SUPSUB
2041 015346 154240 CHARS1
2042 015350 173400 DSTOP
2043 015352 160000 DJMP
2044 015354 015256 SUPPIC
2045
2046 .SBTTL SUPER AND SUBSCRIPT ASCII STRING
2047
2048 015356 105 021 105 SUPSUB: .BYTE 105,SUPON,105,SUPON,105,SUPON,105
2049 .NOW REVERSE AND INCREASE SIZE
2050 015365 023 105 023 .BYTE SUPOFF,105,SUPOFF,105,SUPOFF,105
```

```
2051 ;NOW IT SHOULD BE AT THE BIGGEST SIZE
2052 015373 022 105 022 .BYTE SUBON,105,SUBON,105,SUBON,105
2053 ;REVERSE AND INCREASE SIZE
2054 015401 024 105 024 .BYTE SUBOFF,105,SUBOFF,105,SUBOFF,105
2055 015407 040 .BYTE 40
2056
2057 015410 166000 DPOP
2058
2059
2060 .SBTTL SYNC SPEED SUBPICTURE
2061
2062 015412 170000 SYNPIC: STATSA ;VARIABLE WORD TO HANDLE SYNC SPEED
2063 015414 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2064 015416 114000 POINT
2065 015420 001000 1000
2066 015422 000000 0
2067 015424 120000 BASICV
2068 015426 047000 INTX!PATH1!1000 ;DRAW A DIAMOND
2069 015430 057000 INTX!PATH3!1000
2070 015432 067000 INTX!PATH5!1000
2071 015434 077000 INTX!PATH7!1000
2072 015436 114000 POINT
2073 015440 000600 600
2074 015442 001000 1000
2075 015444 176003 STRNGi ;ENABLE CHARACTER TERMINATE
2076 015446 100000 CHAR
2077 015450 162000 DJSR
2078 015452 015464 SYNTXT ;DISPLAY SYNC SPEED MESSAGE
2079 015454 176002 STRNG0 ;DISABLE CHARACTER STRING ESCAPE
2080 015456 173400 DSTOP
2081 015460 160000 DJMP
2082 015462 015412 SYNPIC ;CONTINUE
2083
2084 015464 044124 051511 043040 SYNTXT: .ASCII /THIS FRAME USES /
2085 015504 047516 SYNSPD: .ASCII /NO/
2086 015506 051440 047131 020103 .ASCII / SYNC /
2087 015531 177 .BYTE 177
2088
2089 ;SHOULD NEVER GET HERE UNLESS CHAR TERM. FAILS
2090 015532 114000 1$: POINT
2091 015534 000200 200
2092 015536 000700 700
2093 015540 100000 CHAR
2094 015542 044103 051101 041501 .ASCIZ /CHARACTER TERMINATE FAILED TO CAUSE A POP AND RESTORE/
2095 015630 160000 DJMP
2096 015632 015532 1$
2097
2098 .SBTTL DASH LINE SUB-PICTURE
2099
2100 015634 117000 FRME5: POINT!INT4
2101 015636 000000 0
2102 015640 001000 1000
2103 015642 154240 CHARS1
2104 015644 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2105 015646 100004 CHAR!LINE0
2106 015650 047523 044514 020104 .ASCII /SOLID /
```

```
2107 015662 110004 LONGV!LINE0
2108 015664 040400 40400
2109 015666 000000 0
2110 015670 000400 400
2111 015672 000000 0
2112 015674 110030 LONGV!BLKON
2113 015676 040400 40400
2114 015700 000000 0
2115 015702 100020 CHAR!BLKOFF
2116 015704 015 012 012 .BYTE 15,12,12,12,12,12
2117 015712 040504 044123 044440 .ASCII /DASH I /
2118 015724 110005 LONGV!LINE1
2119 015726 040400 40400
2120 015730 000000 0
2121 015732 000400 400
2122 015734 000000 0
2123 015736 110030 LONGV!BLKON
2124 015740 040400 40400
2125 015742 000000 0
2126 015744 100020 CHAR!BLKOFF
2127 015746 015 012 012 .BYTE 15,12,12,12,12,12
2128 015754 040504 044123 044440 .ASCII /DASH II /
2129 015766 110006 LONGV!LINE2
2130 015770 040400 40400
2131 015772 000000 0
2132 015774 000400 400
2133 015776 000000 0
2134 016000 110030 LONGV!BLKON
2135 016002 040400 40400
2136 016004 000000 0
2137 016006 100020 CHAR!BLKOFF
2138 016010 015 012 012 .BYTE 15,12,12,12,12,12
2139 016016 040504 044123 044440 .ASCII /DASH III /
2140 016030 110007 LONGV!LINE3
2141 016032 040400 40400
2142 016034 000000 0
2143 016036 000400 400
2144 016040 000000 0
2145 016042 110030 LONGV!BLKON
2146 016044 040400 40400
2147 016046 000000 0
2148 016050 110024 LONGV!BLKOFF!LINE0
2149 016052 000000 0
2150 016054 000000 0
2151 016056 173400 DSTOP
2160 016060 161665 DJMPR!BIT8!WHERE1 ;DJMP RELATIVE TO THE TAG 'FRME5'
2161
2162 .SBTTL VECTOR LENGTH SUB-PICTURE
2163
2164 016062 154024 FRME6: VCTR00!4 ;NORMAL VECTOR
2165 016064 114000 POINT
2166 016066 001777 MAXX
2167 016070 000000 0
2168 016072 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2169 016074 113600 LONGV!INT7
2170 016076 040000 INTX
```

2171 016100 001777
2172 016102 114000
2173 016104 000000
2174 016106 001777
2175 016110 110000
2176 016112 041777
2177 016114 000000
2178 016116 114000
2179 016120 000000
2180 016122 000000
2181 016124 154037
2182 016126 144000
2183 016130 160000
2184 016132 026246
2185
2186
2187
2188

MAXY
POINT
0
MAXY
LONGV
INTX!MAXX
0
PCINT
0
0
VCTROO!17
ABSVCT
DJMP
BUFFER

:MAX LENGTH VECTOR
:ABSOLUTE VECTOR

.SBTTL HORIZONTAL PHOSPHOR SUB-PICTURE

| | | | | |
|------|--------|--------|---------------------|---------------------|
| 2190 | | | | |
| 2191 | 016134 | 114000 | FRME10: POINT | |
| 2192 | 016136 | 000000 | DELTX7: 0 | |
| 2193 | 016140 | 000000 | | |
| 2194 | 016142 | 123600 | DFI10A: BASICV!INT7 | |
| 2200 | 016144 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016146 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016150 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016152 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016154 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016156 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016160 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016162 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016164 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016166 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016170 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016172 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016174 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016176 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016200 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016202 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016204 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016206 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016210 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016212 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016214 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016216 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016220 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016222 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016224 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016226 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016230 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016232 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016234 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016236 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016240 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016242 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016244 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016246 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016250 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016252 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016254 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016256 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016260 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016262 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016264 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016266 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016270 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016272 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016274 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016276 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016300 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016302 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016304 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016306 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016310 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |

| | | | | |
|------|--------|--------|---|--------------------------|
| (1) | 016312 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016314 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016316 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016320 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016322 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016324 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016326 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016330 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016332 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016334 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016336 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016340 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016342 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016344 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016346 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016350 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016352 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016354 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016356 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016360 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016362 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016364 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016366 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016370 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016372 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016374 | 053777 | INTX!PATH2!MAXY | :VECTOR STRAIGHT UP |
| (1) | 016376 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| (1) | 016400 | 073777 | INTX!PATH6!MAXY | :VECTOR DOWN |
| (1) | 016402 | 002002 | PATH0!2 | :MOVE RIGHT 2 UNITS |
| 2201 | 016404 | 173400 | DSTOP | |
| 2202 | 016406 | 160000 | DJMP | |
| 2203 | 016410 | 016142 | DFI10A | |
| 2204 | | | | |
| 2205 | | | .SBTTL MAIN VERTICAL PHOSPHOR SUB-PICTURE | |
| 2206 | | | | |
| 2207 | 016412 | 114000 | FRME11: POINT | |
| 2208 | 016414 | 000000 | 0 | |
| 2209 | 016416 | 000000 | DELTY7: 0 | |
| 2210 | 016420 | 170002 | DMENU0 | |
| 2211 | 016422 | 123600 | DFI11C: BASICV!INT7 | |
| 2217 | 016424 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016426 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016430 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016432 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016434 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016436 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016440 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016442 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016444 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016446 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016450 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016452 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016454 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016456 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016460 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016462 | 012002 | PATH2!2 | :MOVE UP 2-UNITS |

| | | | | |
|-----|--------|--------|-----------------|--------------------------|
| (1) | 016464 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016466 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016470 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016472 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016474 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016476 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016500 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016502 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016504 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016506 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016510 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016512 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016514 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016516 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016520 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016522 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016524 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016526 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016530 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016532 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016534 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016536 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016540 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016542 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016544 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016546 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016550 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016552 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016554 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016556 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016560 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016562 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016564 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016566 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016570 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016572 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016574 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016576 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016600 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016602 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016604 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016606 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016610 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016612 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016614 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016616 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016620 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016622 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016624 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016626 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016630 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016632 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016634 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016636 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016640 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016642 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |

| | | | | |
|------|--------|--------|-----------------|--------------------------|
| (1) | 016644 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016646 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016650 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016652 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016654 | 043777 | INTX!PATH0!MAXX | :VECTOR RIGHT FULL WIDTH |
| (1) | 016656 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016660 | 063777 | INTX!PATH4!MAXX | :VECTOR LEFT FULL SCREEN |
| (1) | 016662 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| 2218 | 016664 | 173400 | DSTOP | |
| 2219 | 016666 | 160000 | DJMP | |
| 2220 | 016670 | 016422 | DFI11C | |
| 2221 | | | | |
| 2222 | | | | |
| 2223 | | | | |

.SBTTL MENU VERTICAL PHOSPHOR SUB-PICTURE

| | | | | |
|------|--------|--------|---------------------|-----------------------------------|
| 2224 | 016672 | 114000 | FRM11S: POINT | |
| 2225 | 016674 | 000000 | 0 | |
| 2226 | 016676 | 000000 | DELT11: 0 | |
| 2227 | 016700 | 123600 | FRM11D: BASICV!INT7 | |
| 2233 | 016702 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016704 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016706 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016710 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016712 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016714 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016716 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016720 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016722 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016724 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016726 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016730 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016732 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016734 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016736 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016740 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016742 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016744 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016746 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016750 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016752 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016754 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016756 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016760 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016762 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016764 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016766 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 016770 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016772 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 016774 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 016776 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 017000 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 017002 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 017004 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 017006 | 062177 | INTX!PATH4!MAXMUX | :VECTOR LEFT FULL MENU SCREEN |
| (1) | 017010 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |
| (1) | 017012 | 042177 | INTX!PATH0!MAXMUX | :VECTOR RIGHT FULL SCREEN IN MENU |
| (1) | 017014 | 012002 | PATH2!2 | :MOVE UP 2 UNITS |

```
(1) 017016 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017020 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017022 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017024 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017026 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017030 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017032 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017034 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017036 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017040 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017042 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017044 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017046 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017050 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017052 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017054 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017056 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017060 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017062 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017064 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017066 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017070 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017072 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017074 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017076 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017100 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017102 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017104 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017106 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017110 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017112 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017114 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017116 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017120 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017122 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017124 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017126 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017130 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017132 042177 INTX!PATH0!MAXMUX ;VECTOR RIGHT FULL SCREEN IN MENU
(1) 017134 012002 PATH2!2 ;MOVE UP 2 UNITS
(1) 017136 062177 INTX!PATH4!MAXMUX ;VECTOR LEFT FULL MENU SCREEN
(1) 017140 012002 PATH2!2 ;MOVE UP 2 UNITS
2234 017142 173400 DSTOP
2235 017144 160000 DJMP
2236 017146 016700 FRM11D
2237
2238 017150 117600 FRM10: POINT!INT7
2239 017152 000000 0
2240 017154 000000 0
2241 017156 164700 CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
2242 017160 110000 LONGV
2243 017162 041777 INTX!MAXX
2244 017164 000000 0
2245 017166 040000 INTX
2246 017170 001777 MAXY
2247 017172 061777 INTX!MINUSX!MAXX
```

2248 017174 000000
2249 017176 040000
2250 017200 021777
2251 017202 173400
2252 017204 160000
2253 017206 017150
2254
2255 017210 170003
2256 017212 117600
2257 017214 000000
2258 017216 000000
2259 017220 110000
2260 017222 040177
2261 017224 000000
2262 017226 040000
2263 017230 001777
2264 017232 060177
2265 017234 000000
2266 017236 040000
2267 017240 021777
2268 017242 173400
2269 017244 160000
2270 017246 017210
2271

0
INTX
MINUSX!MAXY
DSTOP
DJMP
FRM10

FRM11M: DMENU1 ;ENABLE MENU
POINT!INT7
0
0
LONGV
INTX!MAXMUX
0
INTX
MAXY
INTX!MINUSX!MAXMUX
0
INTX
MINUSX!MAXY
DSTOP
DJMP
FRM11M

```
2273  
2285 .SBTTL SHORT VECTOR AND RELATIVE POINT SUB-PICTURE  
2286  
2287  
2288 017250 164700 FRME14: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1  
2289 017252 114000 POINT  
2290 017254 000000 FRM14A: 0  
2291 017256 000000 FRM14B: 0  
2292 017260 104000 SHORTV  
2293 017262 056200 INTX+16200  
(1) 017264 056271 INTX+16200+71  
(1) 017266 040071 INTX+71  
(1) 017270 076271 INTX!MINUSX+16200+71  
(1) 017272 076200 INTX!MINUSX+16200  
(1) 017274 076371 INTX!MINUSX+16200+MINSUY+71  
(1) 017276 040171 INTX+MINSUY+71  
(1) 017300 056371 INTX+16200+MINSUY+71  
(1) 017302 020504 20504  
2294 017304 130000 RELATP  
2295 017306 057000 INTX+17000  
(1) 017310 057074 INTX+17000+74  
(1) 017312 040074 INTX+74  
(1) 017314 077074 INTX!MINUSX+17000+74  
(1) 017316 077000 INTX!MINUSX+17000  
(1) 017320 077174 INTX!MINUSX+17000+MINSUY+74  
(1) 017322 040174 INTX+MINSUY+74  
(1) 017324 057174 INTX+17000+MINSUY+74  
(1) 017326 020504 20504  
2296 017330 104000 SHORTV  
2297 017332 057600 INTX+17600  
(1) 017334 057677 INTX+17600+77  
(1) 017336 040077 INTX+77  
(1) 017340 077677 INTX!MINUSX+17600+77  
(1) 017342 077600 INTX!MINUSX+17600  
(1) 017344 077777 INTX!MINUSX+17600+MINSUY+77  
(1) 017346 040177 INTX+MINSUY+77  
(1) 017350 057777 INTX+17600+MINSUY+77  
(1) 017352 020504 20504  
2298 017354 173400 DSTOP  
2299 017356 160000 DJMP  
2300 017360 017250 FRME14
```

2302
2303
2304 017362 114004
2305 017364 000400
2306 017366 000200
2307 017370 164700
2308 017372 110000
2309 017374 041200
2310 017376 000000
2311
2312 017400 114000
2313 017402 000440
2314 017404 000200
2315 017406 174104
2316 017410 124000
2317 017412 162000
2318 017414 017464
2319 017416 162000
2320 017420 017464
2321
2322 017422 114000
2323 017424 000200
2324 017426 000040
2325 017430 110000
2326 017432 040000
2327 017434 001200
2328
2329 017436 114000
2330 017440 000200
2331 017442 000100
2332 017444 120000
2333 017446 162000
2334 017450 017464
2335 017452 162000
2336 017454 017464
2337 017456 173400
2338 017460 160000
2339 017462 017362
2340

```
.SBTTL GRAPHPLOT INCREMENT SUB-PICTURE  
GRAPH: POINT!LINE0  
400  
200  
CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1  
LONGV ;DRAW BASE REF. VECTOR FOR GRAPH Y  
INTX+1200  
0  
  
POINT  
440  
200  
GRPINC: STATS!INCR+4 ;LOAD GRAPHPLOT INCR. REGISTER  
GRAPHY ;DJSR TO 'SINE DATA'  
DJSR ;DJSR TO SINE DATE  
SINE  
DJSR  
SINE  
  
POINT  
200  
40  
LONGV ;DRAW BASE REF. VECTOR FOR GRAPH X  
INTX  
1200  
  
POINT  
200  
100  
GRAPHX ;DJSR TO 'SINE DATA'  
DJSR ;DJSR TO 'SINE DATA'  
SINE  
DJSR  
SINE  
DSTOP  
DJMP  
GRAPH
```

```
2342
2343
2344
2345 017464 000200 000205 000212 SINE: .SBTTL DATA STRING FOR A SINE WAVE
2346 017510 000257 000262 000265 .WORD 0200,0205,0212,0217,0224,0231,0236,0243,0247,0253
2347 017534 000277 000276 000275 .WORD 0257,0262,0265,0270,0272,0274,0276,0277,0277,0277
2348 017560 000246 000241 000235 .WORD 0277,0276,0275,0274,0272,0267,0264,0261,0256,0252
2349 017604 000163 000156 000151 .WORD 0246,0241,0235,0230,0223,0216,0211,0203,0176,0171
2350 017630 000111 000106 000104 .WORD 0163,0156,0151,0144,0137,0133,0127,0123,0117,0114
2351 017654 000102 000104 000106 .WORD 0111,0106,0104,0102,0101,0100,0100,0100,0100,0101
2352 017700 000144 000151 000156 .WORD 0102,0104,0106,0111,0113,0117,0122,0126,0132,0137
2353
2354 017714 166000 DPOP ;DISPLAY POP AND RESTORE
2355
2356 .SBTTL SHORT TERM DRIFT SUB-PICTURE
```


2358 017716 164700
2359 017720 117000
2360 017722 000000
2361 017724 000000
2362 017726 173400
2363 017730 160000
2364 017732 017716
2365
2366 017734 114000
2367 017736 000000
2368 017740 000000
2369 017742 110000
2370 017744 040000
2371 017746 000000
2372 017750 173400
2373 017752 160000
2374 017754 017734
2375
2376
2377
2378 017756 164700
2379 017760 154024
2380 017762 114000
2381 017764 000000
2382 017766 000000
2383 017770 110000
2384 017772 040000
2385 017774 001777
2386 017776 041777
2387 020000 000000
2388 020002 040000
2389 020004 021777
2390 020006 061777
2391 020010 000000
2392 020012 114000
2393 020014 000040
2394 020016 000000
2395 020020 110000
2401 020022 060100
(1) 020024 000200
(1) 020026 040100
(1) 020030 000200
(1) 020032 060100
(1) 020034 000200
(1) 020036 040100
(1) 020040 000200
(1) 020042 060100
(1) 020044 000200
(1) 020046 040100
(1) 020050 000200
(1) 020052 060100
(1) 020054 000200
(1) 020056 040100
(1) 020060 000200
2402 020062 114000
2403 020064 000000

STDPIC: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
POINT!INT4
STDRA: 0
STDRB: 0
DSTOP
DJMP
STDPIC
PICVTL: POINT
PICVTA: 0
PICVTB: 0
PICVTC: LONGV
INTX
0
PICVTE: DSTOP
DJMP
PICVTL
.SBTTL SCREEN SCISSORING SUB-PICTURE
PICSCS: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
VCTRO0!4
POINT
0
0
LONGV ;BOX
INTX
MAXY
INTX!MAXX
0
INTX
MINUSX!MAXY
INTX!MINUSX!MAXX
0
POINT
40
0
LONGV
INTX!MINUSX!100
200
INTX!100
200
INTX!MINUSX!100
200
INTX!100
200
INTX!MINUSX!100
200
INTX!100
200
INTX!MINUSX!100
200
INTX!100
200
POINT
0

| | | | |
|------|--------|--------|-----------------|
| 2404 | 020066 | 001737 | MAXY-40 |
| 2405 | 020070 | 110000 | LONGV |
| 2411 | 020072 | 040200 | INTX!200 |
| (1) | 020074 | 000100 | 100 |
| (1) | 020076 | 040200 | INTX!200 |
| (1) | 020100 | 020100 | MINUSX!100 |
| (1) | 020102 | 040200 | INTX!200 |
| (1) | 020104 | 000100 | 100 |
| (1) | 020106 | 040200 | INTX!200 |
| (1) | 020110 | 020100 | MINUSX!100 |
| (1) | 020112 | 040200 | INTX!200 |
| (1) | 020114 | 000100 | 100 |
| (1) | 020116 | 040200 | INTX!200 |
| (1) | 020120 | 020100 | MINUSX!100 |
| (1) | 020122 | 040200 | INTX!200 |
| (1) | 020124 | 000100 | 100 |
| (1) | 020126 | 040200 | INTX!200 |
| (1) | 020130 | 020100 | MINUSX!100 |
| 2412 | 020132 | 114000 | POINT |
| 2413 | 020134 | 001737 | MAXX-40 |
| 2414 | 020136 | 001777 | MAXY |
| 2415 | 020140 | 110000 | LONGV |
| 2421 | 020142 | 040100 | INTX!100 |
| (1) | 020144 | 020200 | MINUSX!200 |
| (1) | 020146 | 060100 | INTX!MINUSX!100 |
| (1) | 020150 | 020200 | MINUSX!200 |
| (1) | 020152 | 040100 | INTX!100 |
| (1) | 020154 | 020200 | MINUSX!200 |
| (1) | 020156 | 060100 | INTX!MINUSX!100 |
| (1) | 020160 | 020200 | MINUSX!200 |
| (1) | 020162 | 040100 | INTX!100 |
| (1) | 020164 | 020200 | MINUSX!200 |
| (1) | 020166 | 060100 | INTX!MINUSX!100 |
| (1) | 020170 | 020200 | MINUSX!200 |
| (1) | 020172 | 040100 | INTX!100 |
| (1) | 020174 | 020200 | MINUSX!200 |
| (1) | 020176 | 060100 | INTX!MINUSX!100 |
| (1) | 020200 | 020200 | MINUSX!200 |

```
2423 020202 114000 POINT
2424 020204 001777 MAXX
2425 020206 000040 40
2426 020210 110000 LONGV
2432 020212 060200 INTX!MINUSX!200
(1) 020214 020100 MINUSX!100
(1) 020216 060200 INTX!MINUSX!200
(1) 020220 000100 100
(1) 020222 060200 INTX!MINUSX!200
(1) 020224 020100 MINUSX!100
(1) 020226 060200 INTX!MINUSX!200
(1) 020230 000100 100
(1) 020232 060200 INTX!MINUSX!200
(1) 020234 020100 MINUSX!100
(1) 020236 060200 INTX!MINUSX!200
(1) 020240 000100 100
(1) 020242 060200 INTX!MINUSX!200
(1) 020244 020100 MINUSX!100
(1) 020246 060200 INTX!MINUSX!200
(1) 020250 000100 100
2433 ;POSITION THE STARTING POINT OFF OF THE VIEWING SCREENE
2434 020252 114000 POINT
2435 020254 000777 MAXX/2
2436 020256 000000 0
2437 020260 110000 LONGV
2438 020262 000000 0
2439 020264 020200 MINUSX!200
2440 ;NOW DRAW AN DIAMOND THAT INTERSECTS EACH OF THE FOUR EDGES
2441 020266 110000 LONGV
2442 020270 041200 INTX!1200
2443 020272 001200 1200
2444 020274 061200 INTX!MINUSX!1200
2445 020276 001200 1200
2446 020300 061200 INTX!MINUSX!1200
2447 020302 021200 MINUSX!1200
2448 020304 041200 INTX!1200
2449 020306 021200 MINUSX!1200
2450
2451 .SBTTL VECTOR SCALE SUB-PICTURE
2452
2453 020310 154024 VCTROO!4
2454 020312 117600 POINT!INT7
2455 020314 000777 MAXX/2
2456 020316 000777 MAXY/2
2457 020320 154024 PICSCA: VCTROO!4
2458 020322 110000 LONGV
2459 020324 020150 MINUSX!150
2460 020326 020150 MINUSY!150
2461 020330 040320 INTX!320
2462 020332 000000 0
2463 020334 040000 INTX
2464 020336 000320 320
2465 020340 060320 INTX!MINUSX!320
2466 020342 000000 0
2467 020344 040000 INTX
2468 020346 020320 MINUSX!320
```

| | | | | |
|------|--------|--------|------------------|-----------------------------|
| 2469 | 020350 | 154024 | VCTR00!4 | |
| 2470 | 020352 | 173400 | DSTOP | |
| 2471 | 020354 | 160000 | DJMP | |
| 2472 | 020356 | 017756 | PICSCS | |
| 2473 | | | .SBTTL | VECTOR STARTING SUB-PICTURE |
| 2474 | | | | |
| 2475 | 020360 | 114000 | VSTRT: POINT | |
| 2476 | 020362 | 001003 | 1003 | |
| 2477 | 020364 | 001200 | 640. | |
| 2478 | 020366 | 110000 | LONGV | :VECTOR 1 |
| 2479 | 020370 | 040000 | INTX | |
| 2480 | 020372 | 000577 | 383. | |
| 2481 | 020374 | 114000 | POINT | |
| 2482 | 020376 | 001003 | 1003 | |
| 2483 | 020400 | 001400 | 768. | |
| 2484 | 020402 | 110000 | LONGV | :VECTOR 2 |
| 2485 | 020404 | 040200 | INTX!128. | |
| 2486 | 020406 | 000000 | 0 | |
| 2487 | 020410 | 114000 | POINT | |
| 2488 | 020412 | 001004 | 1004 | |
| 2489 | 020414 | 001366 | 758. | |
| 2490 | 020416 | 110000 | LONGV | :VECTOR 3 |
| 2491 | 020420 | 040177 | INTX!127. | |
| 2492 | 020422 | 000000 | 0 | |
| 2493 | 020424 | 114000 | POINT | |
| 2494 | 020426 | 001003 | 1003 | |
| 2495 | 020430 | 001400 | 768. | |
| 2496 | 020432 | 110000 | LONGV | :VECTOR 4 |
| 2497 | 020434 | 060200 | INTX!MINUSX!128. | |
| 2498 | 020436 | 000000 | 0 | |
| 2499 | 020440 | 114000 | POINT | |
| 2500 | 020442 | 001002 | 1002 | |
| 2501 | 020444 | 001366 | 758. | |
| 2502 | 020446 | 110000 | LONGV | :VECTOR 5 |
| 2503 | 020450 | 060177 | INTX!MINUSX!127. | |
| 2504 | 020452 | 000000 | 0 | |
| 2505 | 020454 | 114000 | POINT | |
| 2506 | 020456 | 001003 | 1003 | |
| 2507 | 020460 | 001200 | 640. | |
| 2508 | 020462 | 110000 | LONGV | :VECTOR 6 |
| 2509 | 020464 | 040200 | INTX!128. | |
| 2510 | 020466 | 000000 | 0 | |
| 2511 | 020470 | 114000 | POINT | |
| 2512 | 020472 | 001003 | 1003 | |
| 2513 | 020474 | 001200 | 640. | |
| 2514 | 020476 | 110000 | LONGV | :VECTOR 7 |
| 2515 | 020500 | 040000 | INTX | |
| 2516 | 020502 | 020200 | MINUSX!128. | |
| 2517 | 020504 | 114000 | POINT | |
| 2518 | 020506 | 001003 | 1003 | |
| 2519 | 020510 | 001200 | 640. | |
| 2520 | 020512 | 110000 | LONGV | :VECTOR 8 |
| 2521 | 020514 | 060200 | INTX!MINUSX!128. | |
| 2522 | 020516 | 000000 | 0 | |
| 2523 | | | | |
| 2524 | | | | |

```
2525 .SBTTL MAJOR AXIS OFFSET SUB-PICTURE
2526 : +X +Y
2527 020520 114000 POINT
2528 020522 001000 1000
2529 020524 000400 400
2530 020526 110000 LONGV
2531 020530 040177 INTX!177 ;X MINOR
2532 020532 000200 200
2533 020534 114000 POINT
2534 020536 001000 1000
2535 020540 000400 400
2536 020542 110000 LONGV
2537 020544 040200 INTX!200 ;Y MINOR
2538 020546 000177 177
2539 : +X -Y
2540
2541 020550 114000 POINT
2542 020552 001000 1000
2543 020554 000400 400
2544 020556 110000 LONGV
2545 020560 040177 INTX!177 ;X MINOR
2546 020562 020200 MINUSY!200
2547 020564 114000 POINT
2548 020566 001000 1000
2549 020570 000400 400
2550 020572 110000 LONGV
2551 020574 040200 INTX!200 ;Y MINOR
2552 020576 020177 MINUSX!177
2553
2554 : -X -Y
2555 020600 114000 POINT
2556 020602 001000 1000
2557 020604 000400 400
2558 020606 110000 LONGV
2559 020610 060177 INTX!MINUSX!177 ;X MINOR
2560 020612 020200 MINUSY!200
2561 020614 114000 POINT
2562 020616 001000 1000
2563 020620 000400 400
2564 020622 110000 LONGV
2565 020624 060200 INTX!MINUSX!200 ;Y MINOR
2566 020626 020177 MINUSX!177
2567
2568 : -X +Y
2569 020630 114000 POINT
2570 020632 001000 1000
2571 020634 000400 400
2572 020636 110000 LONGV
2573 020640 060177 INTX!MINUSX!177 ;X MINOR
2574 020642 000200 200
2575 020644 114000 POINT
2576 020646 001000 1000
2577 020650 000400 400
2578 020652 110000 LONGV
2579 020654 060200 INTX!MINUSX!200
2580 020656 000177 177
```

```
2581
2582 020660 173400          DSTOP
2583 020662 160000          DJMP
2584 020664 020360          VSTRT
2585
2586
2587          .SBTTL CHARACTER SCALE SUB-PICTURE
2588
2589 ;'A' CHARACTER
2590
2591 020666 164700          CHAQU: CONSL1!BIT7!BIT6          ;ENABLE CONSOLE #1
2592 020670 114000          POINT
2593 020672 000700          700
2594 020674 001400          1400
2595 020676 110000          LONGV          ;DRAW REF. LINE
2596 020700 040400          INTX!400
2597 020702 000000          0
2598 020704 114000          POINT
2599 020706 000700          700
2600 020710 001400          1400
2601 020712 154340          CHARS3          ;CHAR SIZE 3 ( X2)
(1) 020714 100000          CHAR          ;CHARACTER MODE
(1) 020716 101          .BYTE 101
(1) 020717 000          .BYTE 0
(1) 020720 154300          CHARS2          ;CHAR SIZE 2 ( 1 1/2 X)
(1) 020722 100000          CHAR          ;CHAR MODE
(1) 020724 101          .BYTE 101
(1) 020725 000          .BYTE 0
(1) 020726 154240          CHARS1          ;CHAR SIZE 1 ( 1X)
(1) 020730 100000          CHAR
(1) 020732 101          .BYTE 101
(1) 020733 000          .BYTE 0
(1) 020734 154200          CHARS0          ;CHAR SIZE ( 1/2)
(1) 020736 100000          CHAR
(1) 020740 101          .BYTE 101
(1) 020741 000          .BYTE 0
2602
2603 ;'B' CHARACTER
2604
2605 020742 114000          POINT
2606 020744 000700          700
2607 020746 001200          1200
2608 020750 110000          LONGV          ;DRAW REF. LINE
2609 020752 040400          INTX!400
2610 020754 000000          0
2611 020756 114000          POINT
2612 020760 000700          700
2613 020762 001200          1200
2614 020764 154340          CHARS3          ;CHAR SIZE 3 ( X2)
(1) 020766 100000          CHAR          ;CHARACTER MODE
(1) 020770 102          .BYTE 102
(1) 020771 000          .BYTE 0
(1) 020772 154300          CHARS2          ;CHAR SIZE 2 ( 1 1/2 X)
(1) 020774 100000          CHAR          ;CHAR MODE
(1) 020776 102          .BYTE 102
(1) 020777 000          .BYTE 0
```

| | | | | | |
|------|--------|--------|------------|-----|-------------------------|
| (1) | 021000 | 154240 | CHARS1 | | :CHAR SIZE 1 (1X) |
| (1) | 021002 | 100000 | CHAR | | |
| (1) | 021004 | 102 | .BYTE | 102 | |
| (1) | 021005 | 000 | .BYTE | 0 | |
| (1) | 021006 | 154200 | CHARS0 | | :CHAR SIZE (1/2) |
| (1) | 021010 | 100000 | CHAR | | |
| (1) | 021012 | 102 | .BYTE | 102 | |
| (1) | 021013 | 000 | .BYTE | 0 | |
| 2615 | | | | | |
| 2616 | | | | | |
| 2617 | | | | | |
| 2618 | 021014 | 114000 | POINT | | |
| 2619 | 021016 | 000700 | 700 | | |
| 2620 | 021020 | 001000 | 1000 | | |
| 2621 | 021022 | 110000 | LONGV | | :DRAW REF. LINE |
| 2622 | 021024 | 040400 | INTX!400 | | |
| 2623 | 021026 | 000000 | 0 | | |
| 2624 | 021030 | 114000 | POINT | | |
| 2625 | 021032 | 000700 | 700 | | |
| 2626 | 021034 | 001000 | 1000 | | |
| 2627 | 021036 | 154340 | CHARS3 | | :CHAR SIZE 3 (X2) |
| (1) | 021040 | 100000 | CHAR | | :CHARACTER MODE |
| (1) | 021042 | 106 | .BYTE | 106 | |
| (1) | 021043 | 000 | .BYTE | 0 | |
| (1) | 021044 | 154300 | CHARS2 | | :CHAR SIZE 2 (1 1/2 X) |
| (1) | 021046 | 100000 | CHAR | | :CHAR MODE |
| (1) | 021050 | 106 | .BYTE | 106 | |
| (1) | 021051 | 000 | .BYTE | 0 | |
| (1) | 021052 | 154240 | CHARS1 | | :CHAR SIZE 1 (1X) |
| (1) | 021054 | 100000 | CHAR | | |
| (1) | 021056 | 106 | .BYTE | 106 | |
| (1) | 021057 | 000 | .BYTE | 0 | |
| (1) | 021060 | 154200 | CHARS0 | | :CHAR SIZE (1/2) |
| (1) | 021062 | 100000 | CHAR | | |
| (1) | 021064 | 106 | .BYTE | 106 | |
| (1) | 021065 | 000 | .BYTE | 0 | |
| 2628 | | | | | |
| 2629 | | | | | |
| 2630 | | | | | |
| 2631 | 021066 | 117000 | POINT!INT4 | | |
| 2632 | 021070 | 000700 | 700 | | |
| 2633 | 021072 | 000600 | 600 | | |
| 2634 | | | | | |
| 2635 | 021074 | 154340 | CHARS3 | | :CHAR SIZE 3 (X2) |
| (1) | 021076 | 100000 | CHAR | | :CHARACTER MODE |
| (1) | 021100 | 117 | .BYTE | 117 | |
| (1) | 021101 | 000 | .BYTE | 0 | |
| (1) | 021102 | 154300 | CHARS2 | | :CHAR SIZE 2 (1 1/2 X) |
| (1) | 021104 | 100000 | CHAR | | :CHAR MODE |
| (1) | 021106 | 117 | .BYTE | 117 | |
| (1) | 021107 | 000 | .BYTE | 0 | |
| (1) | 021110 | 154240 | CHARS1 | | :CHAR SIZE 1 (1X) |
| (1) | 021112 | 100000 | CHAR | | |
| (1) | 021114 | 117 | .BYTE | 117 | |
| (1) | 021115 | 000 | .BYTE | 0 | |
| (1) | 021116 | 154200 | CHARS0 | | :CHAR SIZE (1/2) |

```
(1) 021120 100000 CHAR
(1) 021122 117 .BYTE 117
(1) 021123 000 .BYTE 0
2636
2637 021124 154024 VCTR00!4 ;LOAD VECTOR SCALE TO NORMAL SIZE
2638 021126 114000 POINT
2639 021130 000700 700
2640 021132 000600 600
2641 021134 154030 VCTR00!10 ;LOAD 2X VECTOR SIZE
2642 021136 162000 DJSR ;DJSR TO DISPLAY SCALED POINTS AROUND THE 'O'
2643 021140 021340 ORELPT
2644 021142 154026 VCTR00!6 ;LOAD VECTOR SCALE TO 1 1/2 SIZE
2645 021144 162000 DJSR ;DJSR TO DISPLAY SCALED POINTS
2646 021146 021340 ORELPT
2647 021150 154024 VCTR00!4 ;LOAD VECTOR SCALE TO 1 SIZE
2648 021152 162000 DJSR ;DJSR TO DISPLAY POINTS
2649 021154 021340 ORELPT
2650 021156 154022 VCTR00!2 ;LOAD VECTOR SCALE TO 1/2 SIZE
2651 021160 162000 DJSR ;DJSR TO DISPLAY RELATIVE POINTS
2652 021162 021340 ORELPT
2653 021164 154024 VCTR00!4 ;RETURN TO NORMAL SIZE
2654 021166 164000 DNOP
2655 021170 164000 DNOP
2656 021172 164000 DNOP
2657 021174 164000 DNOP
2658 021176 164000 DNOP
2659 021200 164000 DNOP
2660 021202 164000 DNOP
2661
2662 ;'T' CHARACTER
2663
2664 021204 114000 POINT
2665 021206 000700 700
2666 021210 000400 400
2667 021212 110000 LONGV
2668 021214 040400 INTX!400
2669 021216 000000 0
2670 021220 114000 POINT
2671 021222 000700 700
2672 021224 000400 400
2673 021226 154340 CHARS3 ;CHAR SIZE 3 ( X2)
(1) 021230 100000 CHAR ;CHARACTER MODE
(1) 021232 124 .BYTE 124
(1) 021233 000 .BYTE 0
(1) 021234 154300 CHARS2 ;CHAR SIZE 2 ( 1 1/2 X)
(1) 021236 100000 CHAR ;CHAR MODE
(1) 021240 124 .BYTE 124
(1) 021241 000 .BYTE 0
(1) 021242 154240 CHARS1 ;CHAR SIZE 1 ( 1X)
(1) 021244 100000 CHAR
(1) 021246 124 .BYTE 124
(1) 021247 000 .BYTE 0
(1) 021250 154200 CHARSO ;CHAR SIZE ( 1/2)
(1) 021252 100000 CHAR
(1) 021254 124 .BYTE 124
(1) 021255 000 .BYTE 0
```


2674
2675 021256 114000
2676 021260 000700
2677 021262 000200
2678 021264 110000
2679 021266 040400
2680 021270 000000
2681 021272 114000
2682 021274 000700
2683 021276 000200
2684 021300 154340
(1) 021302 100000
(1) 021304 130
(1) 021305 000
(1) 021306 154300
(1) 021310 100000
(1) 021312 130
(1) 021313 000
(1) 021314 154240
(1) 021316 100000
(1) 021320 130
(1) 021321 000
(1) 021322 154200
(1) 021324 100000
(1) 021326 130
(1) 021327 000
2685 021330 154240
2686 021332 173400
2687 021334 160000
2688 021336 020666
2689
2690 021340 130000
2691 021342 041600
2692 021344 040013
2693 021346 061600
2694 021350 040113
2695 021352 003400
2696 021354 166000
2697
2698
2699
2700

;'X' CHARACTER

POINT
700
200
LONGV
INTX!400
0
POINT
700
200
CHARS3
CHAR
.BYTE 130
.BYTE 0
CHARS2
CHAR
.BYTE 130
.BYTE 0
CHARS1
CHAR
.BYTE 130
.BYTE 0
CHARS0
CHAR
.BYTE 130
.BYTE 0
CHARS1
DSTOP
DJMP
CHAQU

;CHAR SIZE 3 (X2)
;CHARACTER MODE

;CHAR SIZE 2 (1 1/2 X)
;CHAR MODE

;CHAR SIZE 1 (1X)

;CHAR SIZE (1/2)

ORELPT: RELATP
INTX!1600
INTX!13
INTX!MINUSX!1600
INTX!113
3400
DPOP

;ENABLE RELATIVE POINT MODE

.SBTTL ROTATE CHARACTERS SUBPICTURE

2702 021356 170003
2703 021360 114000
2704 021362 000000
2705 021364 000000
2706 021366 120000
2707 021370 042177
2708 021372 053777
2709 021374 062177
2710 021376 073777
2711 021400 114000
2712 021402 000050
2713 021404 000000
2714 021406 155400
2721 021410 163005
2722 021412 155000
2723 021414 170002
2724 021416 173400
2725 021420 160000
2726 021422 026246

ROTCHR: DMENU1 ;ENABLE MENU
POINT
0
0
BASICV ;DRAW REF. BOX
INTX!PATH0!177
INTX!PATH2!MAXY
INTX!PATH4!177
INTX!PATH6!MAXY
POINT
50
0
CHRRT1 ;ENABLE CHAR ROTATION
DJSRR!WHERE2 ;DJSR RELATIVE TO THE TAG "CHARQA"
CHRRTO ;DISABLE ROTATION
DMENU0 ;RETURN TO MAIN SCREEN
DSTOP
DJMP ;JUMP BACK TO MAIN TEXT
BUFFER

2727
2728
2729
2730 021424 170040
2731 021426 100000
2737 021430 044124 020105 052521
(2) 021505 124 042510 050440
2738 021562 015 012
2739 021564 170060
2740 021566 044124 020105 052521
(2) 021643 124 042510 050440
2741 021720 015 012

;TWO COPIES OF THE 'QUICK BROWN FOX' MESSAGE
CHARQA: STATSA!ITAL0 ;NON ITALIC
CHAR
.ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
.ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
.BYTE 15,12
STATSA!ITAL1
.ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
.ASCII /THE QUICK BROWN FOX JUMPS OVER THE LAZY DOGS /
.BYTE 15,12

2742
2743
2744
2745 021722 170040
2746 021724 100000
2755 021726 164 150 145
(2) 021745 040 146 157
(2) 021764 040 164 150
(2) 021776 144 157 147
(2) 022003 164 150 145
(2) 022022 040 146 157
(2) 022041 040 164 150
(2) 022053 144 157 147
2756 022060 015 012
2757 022062 170060
2758 022064 164 150 145
(2) 022103 040 146 157
(2) 022122 040 164 150
(2) 022134 144 157 147
(2) 022141 164 150 145
(2) 022160 040 146 157
(2) 022177 040 164 150
(2) 022211 144 157 147
2759 022216 015 012
2760 022220 170040

;LOWER CASE ASCII MESSAGES
CHARQD: STATSA!ITAL0
CHAR
.BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
.BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
.BYTE 40,164,150,145,40,154,141,172,171,40
.BYTE 144,157,147,163,40
.BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
.BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
.BYTE 40,164,150,145,40,154,141,172,171,40
.BYTE 144,157,147,163,40
.BYTE 15,12
STATSA!ITAL1 ;SET ITALICS
.BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
.BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
.BYTE 40,164,150,145,40,154,141,172,171,40
.BYTE 144,157,147,163,40
.BYTE 164,150,145,40,161,165,151,143,153,40,142,162,157,167,156
.BYTE 40,146,157,170,40,152,165,155,160,163,40,157,166,145,162
.BYTE 40,164,150,145,40,154,141,172,171,40
.BYTE 144,157,147,163,40
.BYTE 15,12
STATSA!ITAL0

2761 022222 166000
2762
2763
2764
2765
2766
2767
2768
2769 022224 164774
2770 022226 164374
2771 022230 114140
2772 022232 001400
2773 022234 001200
2774 022236 110000
(1) 022240 040137
(1) 022242 000000
(1) 022244 040137
(1) 022246 000137
(1) 022250 040000
(1) 022252 000137
(1) 022254 060137
(1) 022256 000137
(1) 022260 060137
(1) 022262 000000
(1) 022264 060137
(1) 022266 020137
(1) 022270 040000
(1) 022272 020137
(1) 022274 040137
(1) 022276 020137
2775
2776 022300 164640
2777 022302 114000
2778 022304 001300
2779 022306 001500
2780 022310 100000
2781 022312 036530
2782 022314 030061 030060
2783 022320 040 040 040
2784 022323 131 020075
2785 022326 030061 030060
2786 022332 114000
2787 022334 001250
2788 022336 001340
2789 022340 100000
2790 022342 160000
2791 022344 022454
2792
2793
2794
2795 022346 164760
2796 022350 164240
2797 022352 114000
2798 022354 001300
2799 022356 001500
2800 022360 100000

DPOP
.SBTTL
.SBTTL LIGHT-PEN SUBPICTURE
.SBTTL
.SBTTL POSITION THE OCTAGON
FRME16: CONSL1!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2 ;ENABLE CONSOLE #1
CONSLO!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2
POINT!LPON
1400
1200
LONGV ;OCTOGON BY LENGTH OF 137
INTX+137
0
INTX+137
137
INTX
137
INTX!MINUSX+137
137
INTX!MINUSX+137
0
INTX!MINUSX+137
MINUSX+137
INTX
MINUSX+137
INTX+137
MINUSX+137
.SBTTL DISPLAY ON CONSOLE #0 THE X-Y READOUT VALUE
CONSL1!BIT7!BIT5 ;DISABLE CONSOLE #1
POINT
1300
1500
CHAR
.ASCII /X=/
DLT14A: .ASCII /1000/
.BYTE 40,40,40
040
.ASCII /Y= /
DLT14B: .ASCII /1000/
POINT
1250
1340
CHAR
DJMP
MSOPEN: PENOF0 ;JUMP TO PEN SWITCH MESSAGE FOR CONSOLE #0
.SBTTL DISPLAY ON CONSOLE #1 THE X-Y READOUT VALUE
LPRTA: CONSL1!BIT7!BIT6!BIT5!BIT4 ;ENABLE CONSOLE #1
CONSLO!BIT7!BIT5 ;DISABLE CONSOLE #0
POINT
1300
1500 ;POSITION THE X-Y MESSAGE
CHAR

```
2801 022362 036530
2802 022364 030061 030060
2803 022370 040 040 040
2804 022373 131 020075
2805 022376 030061 030060
2806 022402 114000
2807 022404 001250
2808 022406 001340
2809 022410 100000
2810 022412 160000
2811 022414 022534
2812
2813
2814 022416 117140
2815 022420 001300
2816 022422 000200
2817 022424 164360
2818 022426 100000
2819 022430 044510 020124 047503
2820 022450 160000
2821 022452 022614
2822
2823 022454 042520 020116 053523
2824 022500 160000
2825 022502 022346
2826 022504 042520 020116 053523
2827 022530 160000
2828 022532 022346
2829 022534 042520 020116 053523
2830 022560 160000
2831 022562 022416
2832 022564 042520 020116 053523
2833 022610 160000
2834 022612 022416
2835
2836
2845
2846 022614 114000
2847 022616 000000
2848 022620 000700
2849 022622 110000
2850 022624 041777
2851 022626 000000
2852
2853 022630 114000
(1) 022632 000000
(1) 022634 000640
(1) 022636 110000
(1) 022640 040000
(1) 022642 000030
2854 022644 114000
(1) 022646 000200
(1) 022650 000640
(1) 022652 110000
(1) 022654 040000
(1) 022656 000030

DLT14C: .ASCII /X=/
         .ASCII /1000/
         .BYTE 40,40,40
DLT14D: .ASCII /Y= /
         .ASCII /1000/
         POINT
         1250
         1340
         CHAR
         DJMP
MS1PEN: PENOF1
.SBTTL DISPLAY HIT COUNT MESSAGE

LPRTC: POINT!INT4!LPON
       1300
       200
CONSLO!BIT7!BIT6!BIT5!BIT4
CHAR
.FRMT16B: .ASCII /HIT COUNT = 0000/
          DJMP
          FRM16C

PENOF0: .ASCII /PEN SWITCH #0 IS OFF/
        DJMP
        LPRTA
PENON0: .ASCII /PEN SWITCH #0 IS ON /
        DJMP
        LPRTA
PENOF1: .ASCII /PEN SWITCH #1 IS OFF/
        DJMP
        LPRTC
PENON1: .ASCII /PEN SWITCH #1 IS ON /
        DJMP
        LPRTC

.SBTTL HORIZONTAL REF. LINE SECTION

FRM16C: POINT
        0
        700
        LONGV
        INTX!MAXX
        0

POINT
0
640
LONGV
INTX
30
POINT
200
640
LONGV
INTX
30

;POSITION THE PEN SWITCH MESSAGE FOR CONSOLE #1
;JUMP TO MESSAGE FOR #1
;ENABLE CONSOLE #0
;POINT TO X CORDINATE '0'
;Y CORD. = 640
;DRAW 30 UNIT VERTICAL LINE
;POINT TO X CORDINATE '200'
;Y CORD. = 640
;DRAW 30 UNIT VERTICAL LINE
```

| | | | | |
|------|--------|--------|-------|------------------------------|
| 2855 | 022660 | 114000 | POINT | :POINT TO X CORDINATE '400' |
| (1) | 022662 | 000400 | 400 | |
| (1) | 022664 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022666 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022670 | 040000 | INTX | |
| (1) | 022672 | 000030 | 30 | |
| 2856 | 022674 | 114000 | POINT | :POINT TO X CORDINATE '600' |
| (1) | 022676 | 000600 | 600 | |
| (1) | 022700 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022702 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022704 | 040000 | INTX | |
| (1) | 022706 | 000030 | 30 | |
| 2857 | 022710 | 114000 | POINT | :POINT TO X CORDINATE '1000' |
| (1) | 022712 | 001000 | 1000 | |
| (1) | 022714 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022716 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022720 | 040000 | INTX | |
| (1) | 022722 | 000030 | 30 | |
| 2858 | 022724 | 114000 | POINT | :POINT TO X CORDINATE '1200' |
| (1) | 022726 | 001200 | 1200 | |
| (1) | 022730 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022732 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022734 | 040000 | INTX | |
| (1) | 022736 | 000030 | 30 | |
| 2859 | 022740 | 114000 | POINT | :POINT TO X CORDINATE '1400' |
| (1) | 022742 | 001400 | 1400 | |
| (1) | 022744 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022746 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022750 | 040000 | INTX | |
| (1) | 022752 | 000030 | 30 | |
| 2860 | 022754 | 114000 | POINT | :POINT TO X CORDINATE '1600' |
| (1) | 022756 | 001600 | 1600 | |
| (1) | 022760 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022762 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 022764 | 040000 | INTX | |
| (1) | 022766 | 000030 | 30 | |
| 2861 | 022770 | 114000 | POINT | :POINT TO X CORDINATE '1777' |
| (1) | 022772 | 001777 | 1777 | |
| (1) | 022774 | 000640 | 640 | :Y CORD. = 640 |
| (1) | 022776 | 110000 | LONGV | :DRAW 30 UNIT VERTICAL LINE |
| (1) | 023000 | 040000 | INTX | |
| (1) | 023002 | 000030 | 30 | |

2862
2882 .SBTTL VERTICAL SPACEING SECTION
2883

| | | | | |
|------|--------|--------|-------------------------|------------------|
| 2884 | 023004 | 114000 | POINT | |
| 2885 | 023006 | 000200 | 200 | |
| 2886 | 023010 | 000010 | 10 | |
| 2887 | 023012 | 100000 | CHAR | |
| 2888 | 023014 | 020130 | .ASCII /X COORD = 200 / | |
| 2889 | 023032 | 114000 | POINT | |
| 2890 | 023034 | 000200 | 200 | |
| 2891 | 023036 | 000060 | 60 | |
| 2892 | | | | |
| 2893 | 023040 | 110000 | LONGV | :DRAW LOWER LINE |
| (1) | 023042 | 040200 | INTX:200 | |

047503 051117

| | | | | |
|------|--------|--------|-----------------|------------------------|
| (1) | 023044 | 000000 | 0 | |
| (1) | 023046 | 000000 | 0 | |
| (1) | 023050 | 000011 | 9. | |
| (1) | 023052 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023054 | 000000 | 0 | |
| (1) | 023056 | 000000 | 0 | |
| (1) | 023060 | 000011 | 9. | |
| (1) | 023062 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023064 | 000000 | 0 | |
| (1) | 023066 | 000000 | 0 | |
| (1) | 023070 | 000011 | 9. | |
| (1) | 023072 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023074 | 000000 | 0 | |
| (1) | 023076 | 000000 | 0 | |
| (1) | 023100 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2894 | 023102 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023104 | 040200 | INTX!200 | |
| (1) | 023106 | 000000 | 0 | |
| (1) | 023110 | 000000 | 0 | |
| (1) | 023112 | 000010 | 8. | |
| (1) | 023114 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023116 | 000000 | 0 | |
| (1) | 023120 | 000000 | 0 | |
| (1) | 023122 | 000010 | 8. | |
| (1) | 023124 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023126 | 000000 | 0 | |
| (1) | 023130 | 000000 | 0 | |
| (1) | 023132 | 000010 | 8. | |
| (1) | 023134 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023136 | 000000 | 0 | |
| (1) | 023140 | 000000 | 0 | |
| (1) | 023142 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2895 | 023144 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023146 | 040200 | INTX!200 | |
| (1) | 023150 | 000000 | 0 | |
| (1) | 023152 | 000000 | 0 | |
| (1) | 023154 | 000007 | 7 | |
| (1) | 023156 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023160 | 000000 | 0 | |
| (1) | 023162 | 000000 | 0 | |
| (1) | 023164 | 000007 | 7 | |
| (1) | 023166 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023170 | 000000 | 0 | |
| (1) | 023172 | 000000 | 0 | |
| (1) | 023174 | 000007 | 7 | |
| (1) | 023176 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023200 | 000000 | 0 | |
| (1) | 023202 | 000000 | 0 | |
| (1) | 023204 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2896 | 023206 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023210 | 040200 | INTX!200 | |
| (1) | 023212 | 000000 | 0 | |
| (1) | 023214 | 000000 | 0 | |
| (1) | 023216 | 000006 | 6 | |
| (1) | 023220 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023222 | 000000 | 0 | |

| | | | | |
|------|--------|--------|-----------------|------------------------|
| (1) | 023224 | 000000 | 0 | |
| (1) | 023226 | 000006 | 6 | |
| (1) | 023230 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023232 | 000000 | 0 | |
| (1) | 023234 | 000000 | 0 | |
| (1) | 023236 | 000006 | 6 | |
| (1) | 023240 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023242 | 000000 | 0 | |
| (1) | 023244 | 000000 | 0 | |
| (1) | 023246 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2897 | 023250 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023252 | 040200 | INTX!200 | |
| (1) | 023254 | 000000 | 0 | |
| (1) | 023256 | 000000 | 0 | |
| (1) | 023260 | 000005 | 5 | |
| (1) | 023262 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023264 | 000000 | 0 | |
| (1) | 023266 | 000000 | 0 | |
| (1) | 023270 | 000005 | 5 | |
| (1) | 023272 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023274 | 000000 | 0 | |
| (1) | 023276 | 000000 | 0 | |
| (1) | 023300 | 000005 | 5 | |
| (1) | 023302 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023304 | 000000 | 0 | |
| (1) | 023306 | 000000 | 0 | |
| (1) | 023310 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2898 | 023312 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023314 | 040200 | INTX!200 | |
| (1) | 023316 | 000000 | 0 | |
| (1) | 023320 | 000000 | 0 | |
| (1) | 023322 | 000004 | 4 | |
| (1) | 023324 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023326 | 000000 | 0 | |
| (1) | 023330 | 000000 | 0 | |
| (1) | 023332 | 000004 | 4 | |
| (1) | 023334 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023336 | 000000 | 0 | |
| (1) | 023340 | 000000 | 0 | |
| (1) | 023342 | 000004 | 4 | |
| (1) | 023344 | 060200 | INTX!MINUSX!200 | ;DRAW UPPER LINE |
| (1) | 023346 | 000000 | 0 | |
| (1) | 023350 | 000000 | 0 | |
| (1) | 023352 | 000040 | 40 | ;OFFSET FOR NEXT LINE |
| 2899 | 023354 | 110000 | LONGV | ;DRAW LOWER LINE |
| (1) | 023356 | 040200 | INTX!200 | |
| (1) | 023360 | 000000 | 0 | |
| (1) | 023362 | 000000 | 0 | |
| (1) | 023364 | 000003 | 3 | |
| (1) | 023366 | 060200 | INTX!MINUSX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023370 | 000000 | 0 | |
| (1) | 023372 | 000000 | 0 | |
| (1) | 023374 | 000003 | 3 | |
| (1) | 023376 | 040200 | INTX!200 | ;DRAW NEXT HIGHER LINE |
| (1) | 023400 | 000000 | 0 | |
| (1) | 023402 | 000000 | 0 | |

```
(1) 023404 000003 3  
(1) 023406 060200 INTX!MINUSX!200 ;DRAW UPPER LINE  
(1) 023410 000000 0  
(1) 023412 000000 0  
(1) 023414 000040 40 ;OFFSET FOR NEXT LINE  
2900 023416 110000 LONGV ;DRAW LOWER LINE  
(1) 023420 040200 INTX!200  
(1) 023422 000000 0  
(1) 023424 000000 0  
(1) 023426 000002 2  
(1) 023430 060200 INTX!MINUSX!200 ;DRAW NEXT HIGHER LINE  
(1) 023432 000000 0  
(1) 023434 000000 0  
(1) 023436 000002 2  
(1) 023440 040200 INTX!200 ;DRAW NEXT HIGHER LINE  
(1) 023442 000000 0  
(1) 023444 000000 0  
(1) 023446 000002 2  
(1) 023450 060200 INTX!MINUSX!200 ;DRAW UPPER LINE  
(1) 023452 000000 0  
(1) 023454 000000 0  
(1) 023456 000040 40 ;OFFSET FOR NEXT LINE  
2901 .SBTTL VARIABLE HORIZ. LINE LENGTH  
2902  
2903 023460 114000 POINT  
2904 023462 001000 1000  
2905 023464 000020 20  
2906 023466 100000 CHAR  
2907 023470 020130 047503 051117 .ASCII /X COORDINATE = 1000 /  
2908 000001 L=1  
2909 000600 M=600  
2921 023514 114000 POINT ;POINT TO Y CORD. " M "  
(1) 023516 001000 1000  
(1) 023520 000600 M  
(1) 023522 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG  
(1) 023524 040001 INTX! L  
(1) 023526 000000 0  
(1) 023530 114000 POINT ;POINT TO Y CORD. " M "  
(1) 023532 001000 1000  
(1) 023534 000560 M  
(1) 023536 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG  
(1) 023540 040002 INTX! L  
(1) 023542 000000 0  
(1) 023544 114000 POINT ;POINT TO Y CORD. " M "  
(1) 023546 001000 1000  
(1) 023550 000540 M  
(1) 023552 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG  
(1) 023554 040003 INTX! L  
(1) 023556 000000 0  
(1) 023560 114000 POINT ;POINT TO Y CORD. " M "  
(1) 023562 001000 1000  
(1) 023564 000520 M  
(1) 023566 110000 LONGV ;DRAW A VECTOR " L " UNITS LONG  
(1) 023570 040004 INTX! L  
(1) 023572 000000 0  
(1) 023574 114000 POINT ;POINT TO Y CORD. " M "
```


| | | | | |
|-----|--------|--------|---------|-----------------------------------|
| (1) | 023576 | 001000 | 1000 | |
| (1) | 023600 | 000500 | M | |
| (1) | 023602 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023604 | 040005 | INTX! L | |
| (1) | 023606 | 000000 | 0 | |
| (1) | 023610 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023612 | 001000 | 1000 | |
| (1) | 023614 | 000460 | M | |
| (1) | 023616 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023620 | 040006 | INTX! L | |
| (1) | 023622 | 000000 | 0 | |
| (1) | 023624 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023626 | 001000 | 1000 | |
| (1) | 023630 | 000440 | M | |
| (1) | 023632 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023634 | 040007 | INTX! L | |
| (1) | 023636 | 000000 | 0 | |
| (1) | 023640 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023642 | 001000 | 1000 | |
| (1) | 023644 | 000420 | M | |
| (1) | 023646 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023650 | 040010 | INTX! L | |
| (1) | 023652 | 000000 | 0 | |
| (1) | 023654 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023656 | 001000 | 1000 | |
| (1) | 023660 | 000400 | M | |
| (1) | 023662 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023664 | 040011 | INTX! L | |
| (1) | 023666 | 000000 | 0 | |
| (1) | 023670 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023672 | 001000 | 1000 | |
| (1) | 023674 | 000360 | M | |
| (1) | 023676 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023700 | 040012 | INTX! L | |
| (1) | 023702 | 000000 | 0 | |
| (1) | 023704 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023706 | 001000 | 1000 | |
| (1) | 023710 | 000340 | M | |
| (1) | 023712 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023714 | 040013 | INTX! L | |
| (1) | 023716 | 000000 | 0 | |
| (1) | 023720 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023722 | 001000 | 1000 | |
| (1) | 023724 | 000320 | M | |
| (1) | 023726 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023730 | 040014 | INTX! L | |
| (1) | 023732 | 000000 | 0 | |
| (1) | 023734 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023736 | 001000 | 1000 | |
| (1) | 023740 | 000300 | M | |
| (1) | 023742 | 110000 | LONGV | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023744 | 040015 | INTX! L | |
| (1) | 023746 | 000000 | 0 | |
| (1) | 023750 | 114000 | POINT | ;POINT TO Y CORD. '' M '' |
| (1) | 023752 | 001000 | 1000 | |
| (1) | 023754 | 000260 | M | |

| | | | | | |
|-----|--------|--------|---------|--|-----------------------------------|
| (1) | 023756 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023760 | 040016 | INTX! L | | |
| (1) | 023762 | 000000 | 0 | | |
| (1) | 023764 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 023766 | 001000 | 1000 | | |
| (1) | 023770 | 000240 | M | | |
| (1) | 023772 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 023774 | 040017 | INTX! L | | |
| (1) | 023776 | 000000 | 0 | | |
| (1) | 024000 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 024002 | 001000 | 1000 | | |
| (1) | 024004 | 000220 | M | | |
| (1) | 024006 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 024010 | 040020 | INTX! L | | |
| (1) | 024012 | 000000 | 0 | | |
| (1) | 024014 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 024016 | 001000 | 1000 | | |
| (1) | 024020 | 000200 | M | | |
| (1) | 024022 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 024024 | 040021 | INTX! L | | |
| (1) | 024026 | 000000 | 0 | | |
| (1) | 024030 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 024032 | 001000 | 1000 | | |
| (1) | 024034 | 000160 | M | | |
| (1) | 024036 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 024040 | 040022 | INTX! L | | |
| (1) | 024042 | 000000 | 0 | | |
| (1) | 024044 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 024046 | 001000 | 1000 | | |
| (1) | 024050 | 000140 | M | | |
| (1) | 024052 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 024054 | 040023 | INTX! L | | |
| (1) | 024056 | 000000 | 0 | | |
| (1) | 024060 | 114000 | POINT | | ;POINT TO Y CORD. '' M '' |
| (1) | 024062 | 001000 | 1000 | | |
| (1) | 024064 | 000120 | M | | |
| (1) | 024066 | 110000 | LONGV | | ;DRAW A VECTOR '' L '' UNITS LONG |
| (1) | 024070 | 040024 | INTX! L | | |
| (1) | 024072 | 000000 | 0 | | |

2922 .SBTTL INTENSITY LEVEL SECTION OF LIGHT PEN TEST

| | | | | | |
|------|--------|--------|--------|----------|------------|
| 2923 | | | | | |
| 2924 | 024074 | 114000 | POINT | | |
| 2925 | 024076 | 000200 | 200 | | |
| 2926 | 024100 | 001740 | 1740 | | |
| 2927 | 024102 | 100000 | CHAR | | ;CHAR MODE |
| 2928 | 024104 | 036530 | .ASCII | /X= 200/ | |
| 2929 | 024112 | 114000 | POINT | | |
| 2930 | 024114 | 000700 | 700 | | |
| 2931 | 024116 | 001740 | 1740 | | |
| 2932 | 024120 | 100000 | CHAR | | |
| 2933 | 024122 | 036530 | .ASCII | /X=1000/ | |
| 2934 | 024130 | 114000 | POINT | | |
| 2935 | 024132 | 001100 | 1100 | | |
| 2936 | 024134 | 001740 | 1740 | | |
| 2937 | 024136 | 100000 | CHAR | | |
| 2938 | 024140 | 036530 | .ASCII | /X=1077/ | |

```
2939
2940      001000      J=1000
2941      024146      114000      POINT
2942      024150      000020      20
2943      024152      001700      1700
2944      024154      103600      CHAR!INT7      ;CHAR MODE
2945      024156      036531      033461 030060      .ASCII /Y=1700/
2946      024164      114000      POINT
2947      024166      000200      200
2948      024170      001700      1700
2949      024172      110000      LONGV      ;DRAW HORIZ. LINE
2950      024174      040600      INTX!600
2951      024176      000000      0
2952      024200      130000      RELATP
2953      024202      057600      57600
2954      024204      114000      POINT
2955      024206      000020      20
2956      024210      001600      1600
2957      024212      103400      CHAR!INT6      ;CHAR MODE
2958      024214      036531      033061 030060      .ASCII /Y=1600/
2959      024222      114000      POINT
2960      024224      000200      200
2961      024226      001600      1600
2962      024230      110000      LONGV      ;DRAW HORIZ. LINE
2963      024232      040600      INTX!600
2964      024234      000000      0
2965      024236      130000      RELATP
2966      024240      057600      57600
2967      024242      114000      POINT
2968      024244      000020      20
2969      024246      001500      1500
2970      024250      103200      CHAR!INT5      ;CHAR MODE
2971      024252      036531      032461 030060      .ASCII /Y=1500/
2972      024260      114000      POINT
2973      024262      000200      200
2974      024264      001500      1500
2975      024266      110000      LONGV      ;DRAW HORIZ. LINE
2976      024270      040600      INTX!600
2977      024272      000000      0
2978      024274      130000      RELATP
2979      024276      057600      57600
2980      024300      114000      POINT
2981      024302      000020      20
2982      024304      001400      1400
2983      024306      103000      CHAR!INT4      ;CHAR MODE
2984      024310      036531      032061 030060      .ASCII /Y=1400/
2985      024316      114000      POINT
2986      024320      000200      200
2987      024322      001400      1400
2988      024324      110000      LONGV      ;DRAW HORIZ. LINE
2989      024326      040600      INTX!600
2990      024330      000000      0
2991      024332      130000      RELATP
2992      024334      057600      57600
2993      024336      114000      POINT
2994      024340      000020      20
```

| | | | | | | |
|------|--------|--------|--------|--------|---------------------------------|-------------------|
| 2995 | 024342 | 001300 | | | 1300 | |
| 2996 | 024344 | 102600 | | | CHAR!INT3 | ;CHAR MODE |
| 2997 | 024346 | 036531 | 031461 | 030060 | .ASCII /Y=1300/ | |
| 2998 | 024354 | 114000 | | | POINT | |
| 2999 | 024356 | 000200 | | | 200 | |
| 3000 | 024360 | 001300 | | | 1300 | |
| 3001 | 024362 | 110000 | | | LONGV | ;DRAW HORIZ. LINE |
| 3002 | 024364 | 040600 | | | INTX!600 | |
| 3003 | 024366 | 000000 | | | 0 | |
| 3004 | 024370 | 130000 | | | RELATP | |
| 3005 | 024372 | 057600 | | | 57600 | |
| 3006 | 024374 | 114000 | | | POINT | |
| 3007 | 024376 | 000020 | | | 20 | |
| 3008 | 024400 | 001200 | | | 1200 | |
| 3009 | 024402 | 102400 | | | CHAR!INT2 | ;CHAR MODE |
| 3010 | 024404 | 036531 | 031061 | 030060 | .ASCII /Y=1200/ | |
| 3011 | 024412 | 114000 | | | POINT | |
| 3012 | 024414 | 000200 | | | 200 | |
| 3013 | 024416 | 001200 | | | 1200 | |
| 3014 | 024420 | 110000 | | | LONGV | ;DRAW HORIZ. LINE |
| 3015 | 024422 | 040600 | | | INTX!600 | |
| 3016 | 024424 | 000000 | | | 0 | |
| 3017 | 024426 | 130000 | | | RELATP | |
| 3018 | 024430 | 057600 | | | 57600 | |
| 3019 | 024432 | 114000 | | | POINT | |
| 3020 | 024434 | 000020 | | | 20 | |
| 3021 | 024436 | 001100 | | | 1100 | |
| 3022 | 024440 | 102200 | | | CHAR!INT1 | ;CHAR MODE |
| 3023 | 024442 | 036531 | 030461 | 030060 | .ASCII /Y=1100/ | |
| 3024 | 024450 | 114000 | | | POINT | |
| 3025 | 024452 | 000200 | | | 200 | |
| 3026 | 024454 | 001100 | | | 1100 | |
| 3027 | 024456 | 110000 | | | LONGV | ;DRAW HORIZ. LINE |
| 3028 | 024460 | 040600 | | | INTX!600 | |
| 3029 | 024462 | 000000 | | | 0 | |
| 3030 | 024464 | 130000 | | | RELATP | |
| 3031 | 024466 | 057600 | | | 57600 | |
| 3032 | 024470 | 114000 | | | POINT | |
| 3033 | 024472 | 000020 | | | 20 | |
| 3034 | 024474 | 001000 | | | 1000 | |
| 3035 | 024476 | 102000 | | | CHAR!INT0 | ;CHAR MODE |
| 3036 | 024500 | 036531 | 030061 | 030060 | .ASCII /Y=1000/ | |
| 3037 | 024506 | 114000 | | | POINT | |
| 3038 | 024510 | 000200 | | | 200 | |
| 3039 | 024512 | 001000 | | | 1000 | |
| 3040 | 024514 | 110000 | | | LONGV | ;DRAW HORIZ. LINE |
| 3041 | 024516 | 040600 | | | INTX!600 | |
| 3042 | 024520 | 000000 | | | 0 | |
| 3043 | 024522 | 130000 | | | RELATP | |
| 3044 | 024524 | 057600 | | | 57600 | |
| 3045 | | | | | | |
| 3046 | | | | | .SBTTL DRAW OUTER REFERENCE BOX | |
| 3047 | | | | | | |
| 3048 | 024526 | 117000 | | | POINT!INT4 | |
| 3049 | 024530 | 000000 | | | 0 | |
| 3050 | 024532 | 000000 | | | 0 | |

| | | | | | | | |
|------|--------|--------|--------|--------|--|--|---|
| 3051 | 024534 | 110000 | | | | LONGV | |
| 3052 | 024536 | 041777 | | | | INTX!MAXX | |
| 3053 | 024540 | 000000 | | | | 0 | |
| 3054 | 024542 | 040000 | | | | INTX | |
| 3055 | 024544 | 001777 | | | | MAXY | |
| 3056 | 024546 | 061777 | | | | INTX!MINUSX!MAXX | |
| 3057 | 024550 | 000000 | | | | 0 | |
| 3058 | 024552 | 040000 | | | | INTX | |
| 3059 | 024554 | 021777 | | | | MINUSX!MAXY | |
| 3060 | 024556 | 173400 | | | | DSTOP | |
| 3061 | 024560 | 160000 | | | | DJMP | |
| 3062 | 024562 | 022224 | | | | FRME16 | |
| 3063 | | | | | | | |
| 3064 | | | | | | .SBTTL | |
| 3065 | | | | | | .SBTTL | KEYBOARD CHARACTER ECHO SUB-PICTURE |
| 3066 | 024564 | 114000 | | | | ECHOFR: POINT | |
| 3067 | 024566 | 000000 | | | | 0 | |
| 3068 | 024570 | 001577 | | | | MAXY-200 | |
| 3069 | 024572 | 170010 | | | | STATSA!SYNC40 | :ENABLE SYNC |
| 3070 | 024574 | 154240 | | | | CHARS1 | :ENABLE NORMAL CHAR. SIZE |
| 3071 | 024576 | 100000 | | | | CHAR | |
| 3072 | 024600 | 017 | 017 | | | .BYTE 17,17 | |
| 3073 | 024602 | 042513 | 041131 | 040517 | | .ASCII /KEYBOARD CHARACTER ECHO LOOP/<15><12> | |
| 3074 | 024640 | 020040 | 052103 | 046122 | | .ASCII / CTRL C TO EXIT LOOP/<15><12><12> | |
| 3075 | 024670 | 044103 | 051101 | 041501 | | .ASCII /CHARACTER CODE IS = / | |
| 3076 | 024716 | 000 | 000 | 000 | | .BYTE 0,0,0,0 | :OCTAL VALUE CODE IS LOADED HERE |
| 3077 | 024722 | 015 | 012 | | | ECODEV: .BYTE 15,12 | |
| 3078 | 024724 | 161010 | | | | ECHJMP: DJMPR!10 | :BR OVER IF NOT "SHIFTOUT" MODE |
| 3079 | 024726 | 100000 | | | | CHAR | |
| 3080 | 024730 | 044123 | 043111 | 026524 | | .ASCII /SHIFT-OUT MODE/ | |
| 3081 | 024746 | 015 | 012 | | | .BYTE 15,12 | |
| 3082 | 024750 | 160000 | | | | DJMP | |
| 3083 | 024752 | 026246 | | | | BUFFER | |
| 3084 | | | | | | | |
| 3085 | | | | | | .SBTTL | |
| 3086 | | | | | | .SBTTL | DYNAMIC EXT. STOP FRAME |
| 3087 | | | | | | .SBTTL | |
| 3088 | | | | | | | |
| 3089 | | | | | | :DISPLAY A BOX AROUND THE SCREEN | |
| 3090 | | | | | | : EACH LINE IS A DIFFERENT LINE TYPE AND INTENSITY LEVEL | |
| 3091 | | | | | | | |
| 3092 | 024754 | 164300 | | | | FRME17: CONSLO!BIT7!BIT6 | :ENABLE CONSOLE #0 |
| 3093 | 024756 | 164700 | | | | CONSL1!BIT7!BIT6 | :ENABLE CONSOLE #1 |
| 3094 | 024760 | 114000 | | | | POINT | :POINT |
| 3095 | 024762 | 000000 | | | | 0 | |
| 3096 | 024764 | 001777 | | | | MAXY | :TO TOP LEFT CORNOR |
| 3097 | | | | | | | |
| 3098 | 024766 | 150001 | | | | DNAME!BIT0 | :LOAD NAME REG. WITH #1 |
| 3099 | | | | | | | |
| 3100 | 024770 | 113407 | | | | LONGV!INT6!LINE3 | :LONG VECTOR WITH INTENS. 6 AND LINE TYPE 3 |
| 3101 | 024772 | 041777 | | | | INTX!MAXX | |
| 3102 | 024774 | 000000 | | | | 0 | |
| 3103 | | | | | | | |
| 3104 | 024776 | 113006 | | | | LONGV!INT4!LINE2 | :LONG VECTOR WITH INTENS. 4 AND LINE TYPE 2 |
| 3105 | 025000 | 040000 | | | | INTX | |
| 3106 | 025002 | 021777 | | | | MINUSX!MAXY | |

| | | | | |
|------|--------|--------|------------------|---|
| 3107 | | | | |
| 3108 | 025004 | 112405 | LONGV!INT2!LINE1 | ;LONG VECTOR WITH INTENS. 2 AND LINE TYPE 1 |
| 3109 | 025006 | 061777 | INTX!MINUSX!MAXX | |
| 3110 | 025010 | 000000 | 0 | |
| 3111 | | | | |
| 3112 | 025012 | 113604 | LONGV!INT7!LINE0 | ;LONG VECTOR WITH INTENS. 7 AND LINE TYPE 0 |
| 3113 | 025014 | 040000 | INTX | |
| 3114 | 025016 | 001777 | MAXY | |
| 3115 | | | | |

```
3117 ;DISPLAY A DIAMOND -- WITH SHORT VECTORS AND DIFFERENT INTENSITY LEVELS
3118
3119 025020 150004 DNAME!BIT2 ;LOAD NAME REG. WITH BIT 2
3120 025022 114000 POINT
3121 025024 001000 1000
3122 025026 001500 1500
3123 025030 106200 SHORTV!INT1
3124 025032 057677 57677 ;+X +Y
3125 025034 106600 SHORTV!INT3
3126 025036 077677 77677 ;+X -Y
3127 025040 107200 SHORTV!INT5
3128 025042 077777 77777 ;-X -Y
3129 025044 107600 SHORTV!INT7
3130 025046 057777 57777 ;-X +Y
3131
3132 ;DISPLAY FOUR BLINKING POINTS -- WITH RELATIVE POINT AND BLINK ENABLED
3133
3134 025050 150010 DNAME!BIT3 ;LOAD NAME REG. WITH #10
3135 025052 114000 POINT
3136 025054 001000 1000
3137 025056 000700 700
3138 025060 133030 RELATP!INT4!BLKON ;RELATIVE POINT AND BLINK ON
3139 025062 057677 57677 ;+X +Y
3140 025064 077677 77677 ;+X -Y
3141 025066 077777 77777 ;-X -Y
3142 025070 057777 57777 ;-X +Y
3143
3144 ;DISPLAY FIVE GRAPH PLOT X DATA POINTS
3145
3146 025072 150020 DNAME!BIT4 ;LOAD NAME REG. WITH BIT4
3147 025074 174110 STATSB!INCR+10 ;LOAD GRAPH INCREMENT
3148 025076 114020 POINT!BLKOFF
3149 025100 001000 1000
3150 025102 001600 1600
3151
3152 025104 120000 GRAPHX
3153 025106 001500 001510 001520 1500, 1510, 1520, 1530, 1540
3154
3155 ;DISPLAY FIVE GRAPH PLOT Y DATA POINTS
3156
3157 025120 150040 DNAME!BIT5 ;LOAD NAME REG. WITH BIT5
3158 025122 114000 POINT
3159 025124 001540 1540
3160 025126 001200 1200
3161
3162 025130 124000 GRAPHY
3163 025132 001640 001630 001620 1640, 1630, 1620, 1610
3164
```

```
3166  
3167  
3168 025142 150060  
3169 025144 114000  
3170 025146 001540  
3171 025150 000640  
3172  
3173 025152 120000  
3174 025154 042100  
3175 025156 046100  
3176 025160 052100  
3177 025162 056100  
3178 025164 062100  
3179 025166 066100  
3180 025170 072100  
3181 025172 076100  
3182  
3183  
3184  
3185 025174 150100  
3186 025176 114000  
3187 025200 000400  
3188 025202 000400  
3189  
3190 025204 144000  
3191 025206 041400  
3192 025210 000400  
3193  
3194 025212 041400  
3195 025214 001400  
3196  
3197 025216 040400  
3198 025220 001400  
3199  
3200 025222 040400  
3201 025224 000400  
3202
```

```
      ;DISPLAY AN OCTOGON -- USING BASIC VECTOR'S  
      DNAME!BIT5!BIT4  
      POINT  
      1540  
      640  
      BASICV  
      INTX!PATH0!100  
      INTX!PATH1!100  
      INTX!PATH2!100  
      INTX!PATH3!100  
      INTX!PATH4!100  
      INTX!PATH5!100  
      INTX!PATH6!100  
      INTX!PATH7!100  
      ;DISPLAY A LARGE SQUARE IN THE CENTER -- USING ABSOLUTE VECTORS  
      DNAME!BIT6  
      POINT  
      400  
      400  
      ABSVCT  
      INTX!1400  
      400  
      INTX!1400  
      1400  
      INTX!400  
      1400  
      INTX!400  
      400
```



```
3204 ;NOW USE CHAR MODE AND DISP. JSR'S, CHAR. ROTATE, CHAR ITALICS
3205
3206 025226 150400 DNAME!BIT8 ;LOAD NAME REG. WITH BIT8
3207 025230 170060 STATSA!ITAL1 ;ITALICS ON
3208 025232 155400 CHRRT1 ;CHAR. ROTATE ON
3209
3210 025234 162000 DJSR ;ABSOLUTE JSR TO CHAR. FRAME
3211 025236 025252 SHOWCH
3212
3213 ;NOW USE CHAR MODE, DISP. JSR'S
3214
3215 025240 151000 DNAME!BIT9 ;LOAD NAME REG. WITH BIT9
3216 025242 170040 STATSA!ITAL0 ;ITALICS OFF
3217 025244 155000 CHRRT0 ;CHAR. ROTATE OFF
3218
3219 025246 163001 DJSRR!1 ;RELATIVE DJSR TO CHAR. FRAME
3220 025250 161036 DJMPR!36 ;RELATIVE DJUMP OVER CHAR. SUBROUTINE
3221
3222 025252 114000 SHOWCH: PCINT
3223 025254 000200 200
3224 025256 000200 200
3225
3226 025260 154200 CHARSO ;SET CHAR. SIZE TO 00
3227 025262 100000 CHAR
3228 025264 020040 027060 020065 .ASCII '' 0.5 SIZE''
3229
3230 025276 154240 CHARS1 ;SET CHAR. SIZE TO 01
3231 025300 020040 027061 020060 .ASCII '' 1.0 SIZE''
3232
3233 025312 154300 CHARS2 ;SET CHAR. SIZE TO 10
3234 025314 020040 027061 020065 .ASCII '' 1.5 SIZE''
3235
3236 025326 154340 CHARS3 ;SET CHAR. SIZE TO 11
3237 025330 020040 027062 020060 .ASCII '' 2.0 SIZE''
3238 025342 154240 CHARS1 ;RESET CHAR. SIZE TO NORMAL
3239 025344 166000 DPOP ;EXIT
3240
```

```

3242 ;DISPLAY A RECTANGLE IN THE MENU AREA -- USE DIFFERENT VECTOR SCALES
3243
3244 025346 151400 $FILE2: DNAME!BIT9!BIT8 ;LOAD NAME REG. WITH #1400
3245 025350 170003 DMENU1 ;ENABLE THE MENU AREA
3246 025352 114000 POINT
3247 025354 000000 0
3248 025356 000040 40
3249
3250 025360 154037 VCTR00!17 ;LOAD VECTOR SCALE
3251 025362 110000 LONGV
3252 025364 040000 INTX ;DRAW VERT. LINE
3253 025366 000400 400
3254
3255 025370 154021 VCTR00!1 ;LOAD VECTOR SCALE
3256 025372 040700 INTX!700
3257 025374 000000 0
3258
3259 025376 154037 VCTR00!17 ;LOAD VECTOR SCALE
3260 025400 040000 INTX
3261 025402 020400 MINUSX!400 ;DRAW VERT. LINE
3262
3263 025404 154021 VCTR00!1 ;LOAD VECTOR SCALE
3264 025406 060700 INTX!MINUSX!700
3265 025410 000000 0
3266
3267 025412 170040 STATSA!ITAL0 ;DISABLE ITALICS
3268 025414 154024 VCTR00!4 ;RETURN TO NORMAL SCALE
3269 025416 170002 DMENU0 ;EXIT MENU AREA
3270
3271 025420 173400 DSTOP
3272
3273 025422 160000 DJMP ;JUMP TO START OF FILE
3274 025424 024754 FRM17F: FRME17
3275
3276
3277 025426 164700 FRM17E: CONSL1!BIT7!BIT6 ;ENABLE CONSOLE #1
3278 025430 117030 POINT!INT4!BLKON
3279 025432 000000 0
3280 025434 001000 1000
3281 025436 170040 STATSA!ITAL0 ;ITALICS OFF
3282 025440 155000 CHRRT0 ;CHAR. ROT. OFF
3283 025442 154340 CHAR S3
3284
3285 025444 100000 CHAR
3286 025446 162000 DJSR ;JSR TO ASCII ERROR MESSAGE
3287 025450 025462 WHY: WHY0 ;ADDRESS OF ERROR TYPE
3288 025452 100020 CHAR!BLKOFF
3289 025454 173400 DSTOP
3290 025456 160000 DJMP
3291 025460 025426 FRM17E
3292

```

| | | | | | | |
|------|--------|--------|--------|--------|-------|---|
| 3294 | 025462 | 047516 | 042440 | 052130 | WHY0: | .ASCII /NO EXTERNAL STOP INTERRUPT/ DPOP |
| 3295 | 025514 | 166000 | | | | |
| 3296 | | | | | | |
| 3297 | 025516 | 047125 | 054105 | 042520 | WHY1: | .ASCII /UNEXPECTED INTERRUPT TO VECTOR +4 / DPOP |
| 3298 | 025560 | 166000 | | | | |
| 3299 | | | | | | |
| 3300 | 025562 | 047125 | 054105 | 042520 | WHY2: | .ASCII /UNEXPECTED INTERRUPT TO VECTOR +10/ DPOP |
| 3301 | 025624 | 166000 | | | | |
| 3302 | | | | | | |
| 3303 | 025626 | 047125 | 054105 | 042520 | WHY3: | .ASCII /UNEXPECTED INTERRUPT TO VECTOR +14/ DPOP |
| 3304 | 025670 | 166000 | | | | |
| 3305 | | | | | | |
| 3306 | 025672 | 027104 | 027120 | 027103 | WHY4: | .ASCII /D.P.C. TOO LOW/ DPOP |
| 3307 | 025710 | 166000 | | | | |
| 3308 | | | | | | |
| 3309 | 025712 | 027104 | 027120 | 027103 | WHY5: | .ASCII /D.P.C. TOO HIGH / DPOP |
| 3310 | 025732 | 166000 | | | | |
| 3311 | | | | | | |
| 3312 | 025734 | 052123 | 050117 | 044440 | WHY6: | .ASCII /STOP INTERRUPT BUT NO STOP FLAGS/ DPOP |
| 3313 | 025774 | 166000 | | | | |

```
3321          .SBTTL   SCOPE HANDLER ROUTINE
(1)
(2)          :*****
(1)          :*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
(1)          :*AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
(1)          :*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
(1)          :*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
(1)          :*SW14=1         LOOP ON TEST
(1)          :*SW08=1         LOOP ON TEST IN SWR<7:0>
(1)          :*CALL
(1)          :*         SCOPE            ;;SCOPE=IOT
(1)
(1) 025776          $SCOPE:
(3) 025776 032777 040000 153134          BIT      #BIT14,@SWR            ;TEST IF SW14 = 1
(3) 026004 001047          BNE      $OVER             ;BR IF SET
(3) 026006 005737 007664          TST      HOLD             ;TEST IF LOOP ON PICTURE ?
(3) 026012 001044          BNE      $OVER             ;BR IF LOOP ON THIS TEST
(1) 026014 032777 040000 153116  1$:    BIT      #BIT14,@SWR            ;;LOOP ON PRESENT TEST?
(1) 026022 001040          BNE      $OVER             ;;YES IF SW14=1
(1)          :####START OF CODE FOR THE XOR TESTER####
(1) 026024 000416          $XTSTR: BR      6$           ;;IF RUNNING ON THE 'XOR' TESTER CHANGE
(1)          :THIS INSTRUCTION TO A 'NOP' (NOP=240)
(1) 026026 013746 000004          MOV     @#ERRVEC,-(SP)    ;;SAVE THE CONTENTS OF THE ERROR VECTOR
(1) 026032 012737 026052 000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
(1) 026040 005737 177060          TST     @#177060        ;;TIME OUT ON XOR?
(1) 026044 012637 000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1) 026050 000414          BR      $$SVLAD         ;;GO TO THE NEXT TEST
(1) 026052 022626          5$:    CMP     (SP)+,(SP)+  ;;CLEAR THE STACK AFTER A TIME OUT
(1) 026054 012637 000004          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
(1) 026060 000421          BR      $OVER          ;;LOOP ON THE PRESENT TEST
(1) 026062          6$:;####END OF CODE FOR THE XOR TESTER####
(1) 026062 032777 000400 153050          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
(1) 026070 001404          BEQ     $$SVLAD         ;;BR IF NO
(1) 026072 127737 153042 001102          CMPB   @SWR,$STNM      ;;ON THE RIGHT TEST? SWR<7:0>
(1) 026100 001411          BEQ     $OVER          ;;BR IF YES
(1) 026102 105237 001102          $SVLAD: INCB   $STNM     ;;COUNT TEST NUMBERS
(1) 026106 113737 001102 001176          MOVB   $STNM,$STEN     ;;SET TEST NUMBER IN APT MAILBOX
(1) 026114 011637 001106          MOV     (SP),$LPADR    ;;SAVE SCOPE LOOP ADDRESS
(1) 026120 105037 001103          CLRB   $ERFLG         ;;ZERO THE ERROR FLAG
(1) 026124 013777 001102 153010 $OVER:  MOV     $STNM,@DISPLAY  ;;DISPLAY TEST NUMBER
(1) 026132 013716 001106          MOV     $LPADR,(SP)    ;;FUDGE RETURN ADDRESS
(1) 026136 000002          RTI                     ;;FIXES PS
3322          .=-2
3323 026136 000005          RESET
3324 026140 005737 002246          TST     KRBD           ;TEST IF KEYBOARD CONTROL
3325 026144 001403          BEQ     1$            ;BR IF NOT
3326 026146 052777 000100 152770  1$:    BIS     #BIT6,@$TKS    ;ENABLE KEYBOARD INTR.
3327 026154 000002          RTI
3328 026156 000240          NOP
3329 026160 000240          NOP
3330          APTSIZE=200
```

;DISPATCH TABLE OF THE STARTING ADDRESSES OF EACH TEST

```
3332  
3333  
3334 026162 002250  
3335 026164 002262  
3336 026166 002274  
3337 026170 002470  
3338 026172 002516  
3339 026174 002554  
3340 026176 003156  
3341 026200 003372  
3342 026202 003404  
3343 026204 003460  
3344 026206 004070  
3345 026210 004116  
3346 026212 004202  
3347 026214 004712  
3348 026216 005054  
3349 026220 005066  
3350 026222 005232  
3351 026224 005316  
3352 026226 005442  
3353 026230 005714  
3354 026232 005750  
3355 026234 006700  
3356 026236 006162  
3357 026240 000000 000000 000000  
3358  
3359 026246 000000  
3360  
3361 000001
```

```
DISPTC: TST1  
TST2  
TST3  
TST4  
TST5  
TST6  
TST7  
TST10  
TST11  
TST12  
TST13  
TST14  
TST15  
TST16  
TST17  
TST20  
TST21  
TST22  
TST23  
TST24  
TST25  
TST27  
TST26  
0, 0, 0
```

; X, Y, AND Z UNUSED.

BUFFER: 0 ; USE REMAINING CORE UNDER 12K FOR BUFFER.

.END

| | | | | | | | |
|----------------|------|-------|------|------|------|------|-------|
| ABASE = 172000 | 20# | 30 | | | | | |
| ABSVCT= 144000 | 169# | 1778 | 1834 | 1889 | 1955 | 2182 | 3190 |
| ACDW1 = 000000 | 30 | | | | | | |
| ACDW2 = 000000 | 30 | | | | | | |
| ACPUOP= 000000 | 30 | | | | | | |
| ACRLF 004440 | 668 | 676# | | | | | |
| ADDW0 = 000000 | 30 | | | | | | |
| ADDW1 = 000000 | 30 | | | | | | |
| ADDW10= 000000 | 30 | | | | | | |
| ADDW11= 000000 | 30 | | | | | | |
| ADDW12= 000000 | 30 | | | | | | |
| ADDW13= 000000 | 30 | | | | | | |
| ADDW14= 000000 | 30 | | | | | | |
| ADDW15= 000000 | 30 | | | | | | |
| ADDW2 = 000000 | 30 | | | | | | |
| ADDW3 = 000000 | 30 | | | | | | |
| ADDW4 = 000000 | 30 | | | | | | |
| ADDW5 = 000000 | 30 | | | | | | |
| ADDW6 = 000000 | 30 | | | | | | |
| ADDW7 = 000000 | 30 | | | | | | |
| ADDW8 = 000000 | 30 | | | | | | |
| ADDW9 = 000000 | 30 | | | | | | |
| ADEVCT= 000000 | 30 | | | | | | |
| ADEVM = 000000 | 30 | | | | | | |
| AENV = 000000 | 30 | | | | | | |
| AENVM = 000000 | 30 | | | | | | |
| AFATAL= 000000 | 30 | | | | | | |
| AMADR1= 000000 | 30 | | | | | | |
| AMADR2= 000000 | 30 | | | | | | |
| AMADR3= 000000 | 30 | | | | | | |
| AMADR4= 000000 | 30 | | | | | | |
| AMAMS1= 000000 | 30 | | | | | | |
| AMAMS2= 000000 | 30 | | | | | | |
| AMAMS3= 000000 | 30 | | | | | | |
| AMAMS4= 000000 | 30 | | | | | | |
| AMSGAD= 000000 | 30 | | | | | | |
| AMSGLG= 000000 | 30 | | | | | | |
| AMSGTY= 000000 | 30 | | | | | | |
| AMTYP1= 000000 | 30 | | | | | | |
| AMTYP2= 000000 | 30 | | | | | | |
| AMTYP3= 000000 | 30 | | | | | | |
| AMTYP4= 000000 | 30 | | | | | | |
| APASS = 000000 | 30 | | | | | | |
| APRIOR= 000200 | 22# | 30 | | | | | |
| APTSIZ= 000200 | 69 | 3330# | | | | | |
| ASWREG= 000000 | 30 | | | | | | |
| ATESTN= 000000 | 30 | | | | | | |
| AUNIT = 000000 | 30 | | | | | | |
| AUSWR = 000000 | 30 | | | | | | |
| AVECT1= 100320 | 21# | 30 | | | | | |
| AVECT2= 000000 | 30 | | | | | | |
| BADDON 006542 | 1010 | 1013 | 1016 | 1019 | 1021 | 1023 | 1027# |
| BAD0 006446 | 992 | 1009# | | | | | |
| BAD1 006456 | 960 | 1011# | | | | | |
| BAD2 006470 | 962 | 1014# | | | | | |
| BAD3 006502 | 964 | 1017# | | | | | |

| | | | | | | | | | | | | | | | | |
|----------------|-------|-------|-------|-------|-------|------|------|------|------|------|------|------|------|--|--|--|
| DSREL 001274 | 47# | | | | | | | | | | | | | | | |
| DSR1 001276 | 48# | 979* | 993 | 1029 | | | | | | | | | | | | |
| DSTOP = 173400 | 216# | 425 | 464 | 645 | 801 | 866 | 930 | 1395 | 1482 | 1741 | 1990 | 2007 | 2042 | | | |
| | 2080 | 2151 | 2201 | 2218 | 2234 | 2251 | 2268 | 2298 | 2337 | 2362 | 2372 | 2470 | 2582 | | | |
| | 2686 | 2724 | 3060 | 3271 | 3289 | | | | | | | | | | | |
| DSWR = 177570 | 18# | 30 | 69 | | | | | | | | | | | | | |
| ECHJMP 024724 | 1064* | 1086* | 1092* | 3078# | | | | | | | | | | | | |
| ECHOFR 024564 | 1063 | 1073 | 3066# | | | | | | | | | | | | | |
| ECODEV 024722 | 1055* | 1056* | 1097 | 3077# | | | | | | | | | | | | |
| EDGE0 = 176040 | 247# | 248 | | | | | | | | | | | | | | |
| EDGE1 = 176060 | 248# | | | | | | | | | | | | | | | |
| EMTVEC= 000030 | 18# | | | | | | | | | | | | | | | |
| ERRVEC= 000004 | 18# | 69* | 3321* | | | | | | | | | | | | | |
| FILE 010010 | 1223* | 1224 | 1249# | | | | | | | | | | | | | |
| FILE4A 004516 | 648 | 702# | 721 | | | | | | | | | | | | | |
| FILLIT 004462 | 663 | 667 | 684# | | | | | | | | | | | | | |
| FIL14A 005550 | 869 | 879# | | | | | | | | | | | | | | |
| FIXVCT 001666 | 95 | 97# | 139 | | | | | | | | | | | | | |
| FRME0 010176 | 279 | 1321# | 1397 | | | | | | | | | | | | | |
| FRME10 016134 | 819 | 2191# | | | | | | | | | | | | | | |
| FRME11 016412 | 836 | 2207# | | | | | | | | | | | | | | |
| FRME14 017250 | 884 | 889 | 894 | 899 | 2288# | 2300 | | | | | | | | | | |
| FRME16 022224 | 940 | 2769# | 3062 | | | | | | | | | | | | | |
| FRME17 024754 | 968 | 982 | 997 | 3092# | 3274 | | | | | | | | | | | |
| FRME2 012472 | 344 | 353 | 1489# | 1743 | | | | | | | | | | | | |
| FRME3 013522 | 483 | 1748# | 1992 | | | | | | | | | | | | | |
| FRME5 015634 | 784 | 2100# | 2154 | | | | | | | | | | | | | |
| FRME6 016062 | 803 | 806 | 2164# | | | | | | | | | | | | | |
| FRM10 017150 | 822 | 839 | 2238# | 2253 | | | | | | | | | | | | |
| FRM11D 016700 | 2227# | 2236 | | | | | | | | | | | | | | |
| FRM11M 017210 | 842 | 2255# | 2270 | | | | | | | | | | | | | |
| FRM11S 016672 | 845 | 2224# | | | | | | | | | | | | | | |
| FRM14A 017254 | 880* | 885* | 890* | 895* | 2290# | | | | | | | | | | | |
| FRM14B 017256 | 881* | 886* | 891* | 896* | 2291# | | | | | | | | | | | |
| FRM16B 022450 | 934* | 935* | 1176 | 2820# | | | | | | | | | | | | |
| FRM16C 022614 | 2821 | 2846# | | | | | | | | | | | | | | |
| FRM17E 025426 | 1035 | 3277# | 3291 | | | | | | | | | | | | | |
| FRM17F 025424 | 985 | 3274# | | | | | | | | | | | | | | |
| GNS = ***** U | 25 | | | | | | | | | | | | | | | |
| GRAPH 017362 | 912 | 2304# | 2339 | | | | | | | | | | | | | |
| GRAPHX= 120000 | 164# | 166 | 2332 | 3152 | | | | | | | | | | | | |
| GRAPHY= 124000 | 165# | 2316 | 3162 | | | | | | | | | | | | | |
| GRPINC 017406 | 909* | 913* | 914 | 2315# | | | | | | | | | | | | |
| HALFX = 000777 | 257# | 417 | 418 | 419 | 420 | 421 | 422 | 423 | 424 | | | | | | | |
| HITCNT 007652 | 933* | 1174* | 1175 | 1204# | | | | | | | | | | | | |
| HOLD 007664 | 84* | 135* | 148* | 150* | 1209# | 3321 | | | | | | | | | | |
| HT = 000011 | 18# | | | | | | | | | | | | | | | |
| INCR = 000100 | 229# | 909 | 914 | 2315 | 3147 | | | | | | | | | | | |
| INTX = 040000 | 253# | 301 | 372 | 386 | 399 | 410 | 417 | 418 | 419 | 420 | 421 | 422 | 423 | | | |
| | 424 | 442 | 450 | 454 | 462 | 788 | 797 | 872 | 1111 | 1196 | 1403 | 1405 | 1407 | | | |
| | 1409 | 1411 | 1413 | 1415 | 1417 | 1419 | 1421 | 1423 | 1426 | 1428 | 1430 | 1432 | 1434 | | | |
| | 1436 | 1438 | 1440 | 1442 | 1445 | 1447 | 1449 | 1451 | 1453 | 1455 | 1457 | 1459 | 1461 | | | |
| | 1464 | 1466 | 1468 | 1470 | 1472 | 1474 | 1476 | 1478 | 1480 | 1492 | 1494 | 1496 | 1498 | | | |
| | 1500 | 1502 | 1504 | 1506 | 1508 | 1510 | 1512 | 1514 | 1521 | 1523 | 1525 | 1527 | 1530 | | | |
| | 1532 | 1534 | 1536 | 1539 | 1541 | 1547 | 1549 | 1558 | 1560 | 1562 | 1564 | 1567 | 1569 | | | |
| | 1571 | 1573 | 1576 | 1578 | 1582 | 1584 | 1593 | 1595 | 1597 | 1599 | 1601 | 1603 | 1605 | | | |

| | | | | | | | | | | | | |
|-------|-------|------|------|------|------|-------|------|------|------|------|------|------|
| 1607 | 1609 | 1611 | 1615 | 1617 | 1624 | 1626 | 1628 | 1630 | 1632 | 1634 | 1636 | 1638 |
| 1640 | 1642 | 1646 | 1648 | 1656 | 1662 | 1668 | 1674 | 1691 | 1704 | 1709 | 1714 | 1719 |
| 1724 | 1729 | 1734 | 1739 | 1771 | 1779 | 1780 | 1781 | 1782 | 1783 | 1784 | 1785 | 1786 |
| 1787 | 1788 | 1789 | 1790 | 1791 | 1792 | 1793 | 1794 | 1795 | 1796 | 1797 | 1798 | 1799 |
| 1800 | 1801 | 1802 | 1803 | 1804 | 1805 | 1806 | 1807 | 1808 | 1809 | 1810 | 1811 | 1812 |
| 1813 | 1814 | 1815 | 1816 | 1817 | 1818 | 1819 | 1820 | 1821 | 1822 | 1823 | 1835 | 1836 |
| 1837 | 1838 | 1839 | 1840 | 1841 | 1842 | 1843 | 1844 | 1845 | 1846 | 1847 | 1848 | 1849 |
| 1850 | 1851 | 1852 | 1853 | 1854 | 1855 | 1856 | 1857 | 1858 | 1859 | 1860 | 1861 | 1862 |
| 1863 | 1864 | 1865 | 1866 | 1867 | 1868 | 1869 | 1870 | 1871 | 1872 | 1873 | 1874 | 1875 |
| 1876 | 1877 | 1878 | 1879 | 1890 | 1891 | 1892 | 1893 | 1894 | 1895 | 1896 | 1897 | 1898 |
| 1899 | 1900 | 1901 | 1902 | 1903 | 1904 | 1905 | 1906 | 1907 | 1908 | 1909 | 1910 | 1911 |
| 1912 | 1913 | 1914 | 1915 | 1916 | 1917 | 1918 | 1919 | 1920 | 1921 | 1922 | 1923 | 1924 |
| 1925 | 1926 | 1927 | 1928 | 1929 | 1930 | 1931 | 1932 | 1933 | 1934 | 1942 | 1946 | 1950 |
| 1956 | 1958 | 1960 | 1962 | 1964 | 1966 | 1968 | 1970 | 2003 | 2004 | 2005 | 2006 | 2017 |
| 2028 | 2068 | 2069 | 2070 | 2071 | 2170 | 2176 | 2200 | 2217 | 2233 | 2243 | 2245 | 2247 |
| 2249 | 2260 | 2262 | 2264 | 2266 | 2293 | 2295 | 2297 | 2309 | 2326 | 2370 | 2384 | 2386 |
| 2388 | 2390 | 2401 | 2411 | 2421 | 2432 | 2442 | 2444 | 2446 | 2448 | 2461 | 2463 | 2465 |
| 2467 | 2479 | 2485 | 2491 | 2497 | 2503 | 2509 | 2515 | 2521 | 2531 | 2537 | 2545 | 2551 |
| 2559 | 2565 | 2573 | 2579 | 2596 | 2609 | 2622 | 2668 | 2679 | 2691 | 2692 | 2693 | 2694 |
| 2707 | 2708 | 2709 | 2710 | 2774 | 2850 | 2853 | 2854 | 2855 | 2856 | 2857 | 2858 | 2859 |
| 2860 | 2861 | 2893 | 2894 | 2895 | 2896 | 2897 | 2898 | 2899 | 2900 | 2921 | 2950 | 2963 |
| 2976 | 2989 | 3002 | 3015 | 3028 | 3041 | 3052 | 3054 | 3056 | 3058 | 3101 | 3105 | 3109 |
| 3113 | 3174 | 3175 | 3176 | 3177 | 3178 | 3179 | 3180 | 3181 | 3191 | 3194 | 3197 | 3200 |
| 3252 | 3256 | 3260 | 3264 | | | | | | | | | |
| 176# | 3035 | | | | | | | | | | | |
| 177# | 3022 | 3123 | | | | | | | | | | |
| 178# | 3009 | 3108 | | | | | | | | | | |
| 179# | 2996 | 3125 | | | | | | | | | | |
| 180# | 469 | 1321 | 1999 | 2100 | 2359 | 2631 | 2814 | 2983 | 3048 | 3104 | 3138 | 3278 |
| 181# | 2970 | 3127 | | | | | | | | | | |
| 182# | 2957 | 3100 | | | | | | | | | | |
| 183# | 926 | 2169 | 2194 | 2211 | 2227 | 2238 | 2256 | 2454 | 2944 | 3112 | 3129 | |
| 18# | 69* | | | | | | | | | | | |
| 222# | 640 | 661 | 2034 | 2730 | 2745 | 2760 | 3216 | 3267 | 3281 | | | |
| 223# | 643 | 665 | 2038 | 2739 | 2757 | 3207 | | | | | | |
| 2940# | | | | | | | | | | | | |
| 1098 | 1157 | 1160 | 1165 | 1168 | 1177 | 1258# | | | | | | |
| 89* | 92* | 156# | 3324 | | | | | | | | | |
| 2908# | 2921# | | | | | | | | | | | |
| 18# | | | | | | | | | | | | |
| 190# | 2105 | 2107 | 2148 | 2304 | 3112 | | | | | | | |
| 191# | 2118 | 3108 | | | | | | | | | | |
| 192# | 2129 | 3104 | | | | | | | | | | |
| 193# | 2140 | 3100 | | | | | | | | | | |
| 1109# | 1118 | | | | | | | | | | | |
| 635 | 637 | 639 | 661# | | | | | | | | | |
| 641 | 644 | 650# | | | | | | | | | | |
| 929 | 1117# | | | | | | | | | | | |
| 863 | 865 | 871# | | | | | | | | | | |
| 162# | 409 | 472 | 1119 | 1520 | 1557 | 1592 | 1623 | 1655 | 1661 | 1667 | 1673 | 1771 |
| 1942 | 1946 | 1950 | 2016 | 2027 | 2107 | 2112 | 2118 | 2123 | 2129 | 2134 | 2140 | 2145 |
| 2148 | 2169 | 2175 | 2242 | 2259 | 2308 | 2325 | 2369 | 2383 | 2395 | 2405 | 2415 | 2426 |
| 2437 | 2441 | 2458 | 2478 | 2484 | 2490 | 2496 | 2502 | 2508 | 2514 | 2520 | 2530 | 2536 |
| 2544 | 2550 | 2558 | 2564 | 2572 | 2578 | 2595 | 2608 | 2621 | 2667 | 2678 | 2774 | 2849 |
| 2853 | 2854 | 2855 | 2856 | 2857 | 2858 | 2859 | 2860 | 2861 | 2893 | 2894 | 2895 | 2896 |
| 2897 | 2898 | 2899 | 2900 | 2921 | 2949 | 2962 | 2975 | 2988 | 3001 | 3014 | 3027 | 3040 |

INT0 = 002000
INT1 = 002200
INT2 = 002400
INT3 = 002600
INT4 = 003000
INT5 = 003200
INT6 = 003400
INT7 = 003600
IOTVEC = 000020
ITAL0 = 000040
ITAL1 = 000060
J = 001000
KBCHR 010014
KRBD 002246
L = 000025
LF = 000012
LINE0 = 000004
LINE1 = 000005
LINE2 = 000006
LINE3 = 000007
LOADAC 007162
LOADBF 004374
LOADSP 004340
LOADUP 007204
LOADVT 005522
LONGV = 110000

CZVSDC VS60 VISUAL DISPLAY TEST
CZVSDC.P11 11-SEP-79 09:01

E 10
MACY11 30G(1063) 17-SEP-79 08:50 PAGE 46-1
CROSS REFERENCE TABLE -- MACRO NAMES

SEQ 0121

.\$EOP 7# 1046
.\$SPARM 8#
.\$SPOWE 8#
.\$SSCOP 8# 3321
.\$SWDO 8#
.\$STRAP 9#
.\$STRPT 9#
.\$STYPD 9#
.\$STYPE 9#

. ABS. 026250 000 CON RW REL LCL D

ERRORS DETECTED: 0

CZVSDC,CZVSDC/CRF=CZVSDC
RUN-TIME: 27 13 1 SECONDS
RUN-TIME RATIO: 111/43=2.5
CORE USED: 26K (51 PAGES)