```
        .REM _




                              IDENTIFICATION
                              --------------

                    PRODUCT CODE:          AC-E872C-MC

                    PRODUCT NAME:          CXVSAC0 VS60 MOD

                    PRODUCT DATE:          SEPTEMBER 1978

                    MAINTAINER:            DEC/X11 SUPPORT GROUP

                    THE INFORMATION  IN  THIS  DOCUMENT  IS  SUBJECT  TO  CHANGE
                    WITHOUT  NOTICE  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
                    BY  DIGITAL  EQUIPMENT   CORPORATION.   DIGITAL   EQUIPMENT
                    CORPORATION  ASSUMES  NO  RESPONSIBILITY FOR ANY ERRORS THAT
                    MAY APPEAR IN THIS MANUAL.

                    THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO  THE
                    PURCHASER  UNDER  A  LICENSE  FOR  USE  ON A SINGLE COMPUTER
                    SYSTEM  AND  CAN  BE  COPIED  (WITH  INCLUSION  OF  DIGITALS
                    COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
                    OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

                    DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY  FOR
                    THE  USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
                    NOT SUPPLIED BY DIGITAL.

                  COPYRIGHT (C)  1976,1978 DIGITAL EQUIPMENT CORPORATION
```

1. ABSTRACT

"VSA" IS AN "IOMODP" THAT EXERCISES ONE DECGRAPHIC-11 DISPLAY SYSTEM AND VS60 ADDITIONAL CONSOLE. THE MODULE DISPLAYS A SPECIAL TEST PATTERN THAT CONSISTS OF FOUR FRAMES. THE MODULE, BY EXECUTING ALL OF THE VS-60'S DISPLAY INSTRUCTIONS, WILL VERIFY THE VS-60 OPERATIONS AND PROVIDE A HIGH DEVICE ACTIVITY RATE TO THE UNIBUS. INCLUDED IN EACH FRAME IS A DESCRIPTION OF THE SUB-PICTURES. EACH SUB-PICTURE DESCRIPTION INCLUDES A LIST OF THE DISPLAY INSTRUCTIONS USED FOR THE SUB-PICTURE. BEFORE THE VS-60 IS ENABLED TO DISPLAY THE TEST PATTERN, A READ/WRITE REGISTER TEST IS PERFORMED TO ENSURE SOME OPERATING CONFIDENCE IN THE BASIC HARDWARE INTERFACE. IF BIT 0 OF SR1 IS CLEARED, THE FOUR SUB-PICTURES WILL BE DISPLAYED. AS EACH UNIQUE SUB-PICTURE IS ENTERED, THE DISPLAY NAME REGISTERS IS LOADED WITH A UNIQUE VALUE. IF AN ERROR IS DETECTED, THE VALUE OF THE DISPLAY NAME REGISTER IS TYPED AS THE "STATUS REGISTER". THE NAME REGISTER CAN BE READ TO DETERMINE THE CURRENT SUB-PICTURE IF A PROBLEM ARRISES.

THE MODULE ALSO PROVIDES A MEANS TO VERIFY THE OPERATION OF THE LIGHT-PEN HIT AND LIGHT-PEN SWITCH LOGIC. UPON A LIGHT-PEN HIT, ON EITHER CONSOLE, THE OPERATOR IS INFORMED ON THE SCREEN OF THE HIT. WHEN A LIGHT-PEN SWITCH CONDITION HAS CHANGED, THE OPERATOR IS ALSO INFORMED OF THE CHANGE.

2. REQUIREMENTS

```
HARDWARE:  VS-60 ALPHAGRAPHIC DISPLAY SYSTEM
STORAGE:: VSA REQUIRES:
                1. DECIMAL WORDS: 1613
                2. OCTAL WORDS: 03115
                3. OCTAL BYTES: 6232
```

3. PASS DEFINITION

ONE PASS OF VSA MODULE CONSISTS OF ONE ITERATION OF THE FOUR SUB-PICTURES, WHICH RESULTS IN:

12 THOUSAND PROGRAM INTERRUPTS, 11 MILLION NON-PROCESSOR REQUESTS.

4. EXECUTION TIME

VSA RUNNING ALONE ON PDP-11/05 TAKES APPROXIMATELY 60 SECONDS.
WHEN RUNNING WITH "RELOCATION" ENABLED, THIS MODULE WILL ONLY RUN ON THE 32K BOUNDRY.
THE VISUAL EFFECT IS THAT NOTHING WILL BE SEEN ON THE 8, 16 AND 24K RELOCATION PASSES (REF. 5.).

```
                5.      CONFIGURATION REQUIREMENTS
                        --------------------------

                        DEFAULT PARAMETERS:

                        DEVADR: 172000, VECTOR: 320, BR1: 4, DEVCNT: 1, SR1: 0

                        REQUIRED PARAMETERS:

                        NONE, HOWEVER IF THIS MODULE IS NOT CONFIGURED ACROSS A 8, 16 AND 24K BOUNDRY,
                        THE OPERATOR MAY MODIFY THE "STAT" LOCATION AND CLEAR BIT 10 (2000).
                        THE RESULT WILL CHANGE THIS MODULE FROM A "IOMODP" TO AND "IOMOD".
                        WITH THE RESULT BEING THE MODULE WILL RUN WHEN RELOCATED TO A 8, 16 AND 24K BOUNDRY.
                6.      DEVICE/OPTION SETUP
                        -------------------

                        THE VS-60 MUST HAVE THE POWER ON.


                7.      MODULE OPERATION
                        ----------------

                        THE MODULE WILL BEGIN BY TESTING THE ABLILTY OF THE BUS READ/WRITE
                        REGISTERS TO FUNCTION PROPERLY.  THE REGISTERS VERIFIED ARE:

                                X DYNAMIC OFFSET
                                Y DYNAMIC OFFSET
                                RELOCATE
                                DISPLAY P.C.

                        IF ANY ERRORS ARE DETECTED, THE MODULE WILL BE DROPPED.

                        UPON COMPLETION, THE VALUE IN "SR1" IS TESTED. IF BIT 0 OF SR1
                        IS SET, THE MODULE WILL DESTROY THE ENTIRE VISUAL PICTURE BY LOADING
                        DISPLAY NOP'S THRU THE PICTURE BUFFER.  SETTING OF BIT 0 IN SR1 WILL ENABLE
                        THE VS-60 TO ACCESS THE UNIBUS AT THE VS-60 WORST CASE RATE.
                        RESETTING OF BIT 0 IN SR1 WILL NOT RESTORE THE PICTURE BUFFER.

                        IF BIT 0 OF SR1 IS CLEARED, THE VISUAL PICTURE DISPLAYED CONSISTS
                        OF FOUR FRAMES. THE VIEWING AREA IS DYNAMICLY MOVED ACROSS THE
                        FOUR FRAMES.  THE MOVEMENT IS A FUNCTION OF THE NUMBER OF DISPLAY STOP INTERRUPTS.
```

```
                          TEST PATTERN DESCRIPTION:

                  FRAME0  A.  LINE TYPE TEST:

                               TO TEST THE ABILITY OF THE VS-60 TO DISPLAY
                          EACH OF THE FOUR POSSIBLE LINE TYPES, THE OUTER
                          PERIMETER OF THE TEST PATTERN CONSISTS OF A LARGE
                          RECTANGLE.  EACH SIDE OF THE RECTANGLE IS DISPLAYED
                          USING A DIFFERENT LINE TYPE IE:  SOLID, DASH, DOT,
                          -DASH AND DOT.  (POINT AND LONG VECTOR MODE ARE USED)

                          B.  GRAPHPLOT DISPLAY TEST:

                               TO TEST THE ABILITY OF THE VS-60 TO DISPLAY
                          A GRAPHPLOT PATTERN, TWO EXPANDING SINE WAVE PATTERNS
                          ARE DISPLAYED.  THE FIRST SINE WAVE APPEARS
                          SUPERIMPOSED ON A HORIZONTAL LINE ACROSS THE BOTTOM
                          OF THE SCREEN AND EXPANDS FROM LEFT TO RIGHT.
                          THE SECOND SINE WAVE APPEARS SUPERIMPOSED ON A
                          VERTICAL LINE AT THE LEFT OF THE SCREEN AND EXPANDS
                          FROM BOTTOM TO TOP.  THE EXPANSION OF THE SINE
                          WAVES IS A FUNCTION OF THE DISPLAY INTERRUPT RATE.
                          NO SINE WAVE EXPANSION WOULD INDICATE THAT THE VS-60
                          IS NOT INTERRUPTING THE CPU. THE FOLLOWING MODES ARE USED:

                                  POINT    LONGV
                                  STATSB   GRAPHY
                                  GRAPHX   DJSRR
                                  DJMPR    DPOP

                          C.  VECTOR/RELATIVE POINT AND BLINK TEST:

                               TO TEST THE ABILITY OF THE VS-60 TO DISPLAY
                          VECTORS IN THE LONG, SHORT, AND RELATIVE POINT
                          MODE AND TO BLINK A SELECTED PORTION OF THE
                          DISPLAY, A SET OF SIX NESTED OCTAGONS IS DISPLAYED
                          IN THE UPPER RIGHT QUADRANT OF THE SCREEN.
                          THE TWO OUTERMOST OCTAGONS ARE DISPLAYED USING
                          LONG VECTOR MODE, THE TWO MIDDLE ONES USING
                          SHORT VECTOR MODE, AND THE INNERMOST TWO USING
                          RELATIVE POINT MODE.  THE USE OF RELATIVE POINT
                          POINT MODE CAUSES THE TWO INNERMOST OCTAGONS
                          TO BE DISPLAYED AS EIGHT INTENSIFIED POINTS
                          FOR EACH ONE.  ALTERNATE OCTAGONS STARTING
                          WITH THE INNERMOST ONE ARE BLINKED TO TEST
                          THE OPERATION OF THE BLINK MODE. THE FOLLOWING MODES ARE USED
                          IN THE SUB-PICTURE:

                                  POINT
                                  RELATP
                                  SHORTV
                                  LONGV
```

D. CHARACTER GENERATOR TEST:

TO TEST THE ABILITY OF THE VS-60 TO DISPLAY EACH MEMBER OF ITS CHARACTER SET, THREE PAIRS OF LINES ARE DISPLAYED NEAR THE TOP OF THE SCREEN. THE FIRST LINE IN EACH PAIR DISPLAYS THE CHARACTERS IN NORMAL FONT WHILE THE SECOND LINE DISPLAYS THE SAME CHARACTERS IN ITALIC FONT. THE FIRST PAIR OF LINES DISPLAYS THE 64 ASCII UPPERCASE CHARACTERS (OCTAL CODES 100-137 AND 40-77 DISPLAYED LEFT TO RIGHT). THE SECOND PAIR DISPLAYS THE 32 LOWER CASE ASCII CHARACTERS (OCTAL CODES 140-177 DISPLAYED LEFT TO RIGHT). THE THIRD PAIR DISPLAYS THE 31 SPECIAL CHARACTERS (OCTAL CODES 0-37 DISPLAYED LEFT TO RIGHT) THAT APPEAR AS APL - GREEK - SPECIAL CHARACTERS.

E. INTENSITY LEVEL TEST:

TO TEST THE ABILITY OF THE VS-60 TO VARY THE INTENSITY LEVEL OF THE DISPLAY, EIGHT HORIZONTAL PARALLEL LINES ARE DISPLAYED TO THE LEFT OF CENTER OF THE TEST PATTERN. EACH LINE IS DISPLAYED WITH A DIFFERENT INTENSITY LEVEL STARTING WITH THE TOP LINE AT LEVEL 7 (THE BRIGHTEST) AND PROCEEDING DOWN TO THE BOTTOM LINE AT LEVEL 0 (THE DIMMEST). ALL LINES ARE DISPLAYED IN LONG VECTOR MODE.

F. MENU TEST:

A PERIMETER REFERENCE BOX IS DRAWN USING LONG VECTOR MODE. THE BOX IS QUARTERED BY TWO VECTORS. THE FIRST STARTS AT THE LOWER LEFT TO UPPEN RIGHT CORNER. THE SECOND STARTS FROM THE LOWER RIGHT TO UPPER LEFT CORNER. THE FOLLOWING MODES ARE USED IN THE SUB-PICTURE.

DMENU1	ENABLE MENU
POINT	POINT TO X-Y POSITION
LONGV	DRAW IN LONG VECTOR MODE
DMENU0	DISABLE MENU

G. EDGE SCISSORING TEST:

THE TEST CONSISTS OF DRAWING EIGHT PAIRS OF VECTORS. THE SEQUENCE IS TO DRAW A VECTOR FROM AND "ON-SCREEN" POSITION TO AN "OFF-SCREEN" POSITION AND THEN BACK TO AN "ON-SCREEN" POSITION. THE SEQUENCE IS REPEATED EIGHT TIMES. THE PATTERN WILL APPEAR AT THE TOP OF FRAME 0 WHEN VIEWED. THE FOLOWING MODES ARE USED IN THE SUB-PICTURE: POINT   LONGV

```
        H.  SUPER AND SUB-SCRIPT CHARACTER TEST:

                IN THE UPPER CENTER OF FRAME 0, TWELVE CHARACTERS
        WILL BE DISPLAYED.  THE FIRST CHARACTER BYTE IS THE LETTER
        "B". THE CODE FOR "SUPER-SCRIPT ON" IS THE NEXT BYTE.  THE
        NUMBERS 2 AND 5 ARE THE NEXT BYTES.  WITH "SUPER-SCRIPT ON"
        THE NUMBERS SHOULD REDUCE, BY ONE SIZE, AND ASCEND VERTICALLY
        BY HALF THE SIZE OF THE LETTER "B".  THE NEXT BYTE IS THE CODE
        FOR "SUPER-SCRIPT OFF".  THE RESULT SHOULD BE A RETURN TO THE
        PREVIOUS SIZE AND "Y" POSITION.  THE NEXT BYTE IS ANOTHER
        LETTER "B", APPEARING THE SAME SIZE AND "Y" POSITION
        AS THE INITIAL "B".  THE NEXT BYTE IS THE CODE FOR A
        "SUB-SCRIPT ON", THEN FOLLOWED BY THE NUMBERS 2 AND 5.
        WITH "SUB-SCRIPT ON" THE NUMBERS REDUCE, BY ONE SIZE, IN SIZE
        AND DECEND VERTICALLY BY HALF THE SIZE OF THE LETTER "B".
        THE FOLLOWING BYTE IS THE CODE FOR "SUB-SCRIPT OFF" WHICH WILL
        RETURN TO THE ORIDIGINAL SIZE AND "Y" POSITION.
        THE FOLLOWING MODES USED ARE USED IN THE SUB-PICTURE:

                POINT           POINT TO AN X,Y POSITION
                CHAR            DISPLAY IN CHARACTER MODE
                CHARS1          LOAD CHAR. SCALE TO NORMAL
                SUPON           ENABLE SUPER-SCRIPT ASCII MODE
                SUPOFF          DISABLE SUPER-SCRIPT ASCII MODE
                SUBON           ENABLE SUB-SCRIPT ASCII MODE
                SUBOFF          DISABLE SUPER-SCRIPT ASCII MODE


        I.  CHARACTER SCALE AND ROTATE TEST:

                THE LETTER "B" IS USED TO VERIFY THE OPERATION OF THE
        CHARACTER SCALE LOGIC.  IN THE UPPER RIGHT CORNER OF FRAME 0,
        FOUR "B"'S ARE DISPLAYED. EACH OF THE LETTERS SHOULD BE FOUR
        DIFFERENT SIZES STARTING FROM THE SMALLEST TO LARGEST.
        THE PATTERN IS REPEATED WITH THE CHARACTER 'ITALIC' ENABLED.
        TO VERIFY 'CHARACTER ROTATE', THE SAME PROCEDURE IS REPEATED
        IN THE LOWER LEFT CORNER OF FRAME 0.  THE FOUR LETTERS SHOULD
        APPEAR THESAME AS ABOVE WITH THE EXCEPTION THE CHARACTERS
        SHOULD BE ROTATED BY 90 DEGREES.
        THE FOLOWING MODES ARE USED IN THE SUB-PICTURE:

                POINT           POINT TO X,Y POSITION
                STATSA          LOAD STATUS REG. A
                CHRRT1          ENABLE CHARACTER ROTATE
                DJSRR           DISPLAY JSR RELATIVE TO A SUB-ROUTINE
                CHRRT0          DISABLE CHARACTER ROTATE
                DJMPR           DISPLAY JMP RELATIVE TO A SUB-PICTURE
                DPOPNR          DISPLAY POP AND NO RESTORE
                CHARS0          ENABLE CHARACTER SIZE 0
                CHARS1          ENABLE CHARACTER SIZE 1
                CHARS2          ENABLE CHARACTER SIZE 2
                CHARS3          ENABLE CHARACTER SIZE 3

FRAME1  J.  VECTOR SCALE TEST:
```

```
                THE SUB-PICTURE CONSCISTS OF SIXTEEN DIFFERENT SIZE
        SQUARES STARTING FROM A COMMON POINT.  THE COMMON POINT IS
        RELATIVE 0,0 FROM FRAME 1.  THE VECTOR SCALE IS LOADED WITH
        THE LARGEST VALUE AND A "DJSR" TO A SUB-PICTURE TO DISPLAY
        A 200 UNIT SQUARE.  THE VECTOR SCALE VALUE IS REDICED BY
        ONE AND THEN THE SQUARE IS DRAWN AGAIN.  THE PROCEDURE IS
        REPEATED UNTIL ALL VALUES OF VECTOR SCALE HAVE BEEN LOADED.
        THE FRAME IS BEING DISPLAYED AT PLUS 1X PLUS 1Y SECTOR.
        THE FOLLOWING MODES ARE USED IN THE SUB-PICTURE:

                POINT           POINT TO X,Y CORDINATE
                DJSRR           DISPLAY JSR TO A SUB-ROUTINE
                DJMPR           DISPLAY JMP RELATIVE TO MORE DISPLAY CODE
                LONGV           DISPLAY IN LONG VECTOR MODE
                VCTR00-17       ENABLE VECTOR SCALE 00 THRU 17
                DPOP            DISPLAY POP AND RESTORE THE D.P.U. STATUS

FRAME2  K.  BASIC VECTOR TEST:

                THE SUB-PICTURE DISPLAYS THE EIGHT BASIC VECTOR
        PATHS FROM THE CENTER OF THE FRAME.  A HALF SCREEN LENGTH
        VECTOR IS DRAWN AWAY FROM THE CENTER.  UPON COMPLETION
        OF THE VECTOR, THE OPPOSITE PATH VECTOR IS DRAWN RETURNING
        TO THE CENTER OF THE FRAME.
        THE FOLLOWING MODES ARE USED IN THE SUB-PICTURE:

                POINT           POINT TO X,Y CORDINATE
                BASICV          DISPLAY IN BASIC VECTOR MODE
                PATH0-7         ENABLE PATH (DIRECTION) TO BE DRAWN

FRAME3  L.   STACK LEVEL TEST:

                THE SUB-PICTURE DISPLAYS EIGHT STATEMENTS INDICATING THE
        EIGHT DIFFERENT STACK LEVELS.  THE FRAME USES ALL STACK LEVELS BY
        "NESTING DISPLAY JSR'S". EACH STACK LEVEL WILL LOAD A  DIFFERENT
        VALUE INTO THE DISPLAY NAME REGISTER.  THE SUB-ROUTINE WILL EXECUTE
        AN "DJSR" TO ANOTHER SUBROUTINE UNTIL ALL STACK LEVELS HAVE BEEN
        LOADED. UPON REACHING THE LOWEST STACK LEVEL, A "POP" (RETURN FROM
        SUB-ROUTINE) IS EXECUTED.  THE "POP" SHOULD RETURN TO THE CALLING
        SUB-ROUTINE.
        8.      OPERATION OPTIONS
                -----------------

                BIT 0 OF SR1 CONTROLS WORST CASE BUS RATE.
                WITH BIT 0 CLEARED, THE VS60 WILL BE EXECUTING THE VS60 INSTRUCTION SET.
                WITH BIT 0 SET, THE DISPLAY BUFFER IS LOADED WITH VS60 NOP'S.
                THIS ACTION RESULTS IN THE HIGHEST POSSIBLE UNIBUS 'NPR' REQUEST FOR THE VS60.

                WHEN DEC/X11 RELOCATES TO A 8, 16 AND 24K BOUNDRY, THIS MODULE WILL NOT BE RUN.
                THIS CAN BE DEFEATED BY THE OPERATOR NOT CONFIGURING THE MODULE ACROSS
                A 8, 16 AND 24K BOUNDRY AND CLEARING BIT 10 (2000) OF LOCATION "STAT" OF THIS MODULE.

                THE MOTION OF THE TEST PATTERN CAN BE STOPPED BY DEPRESSING THE
                "LIGHT-PEN" SWITCH ONCE.  RELEASING THE SWITCH WILL RESUME THE TEST
                PATTERN MOVEMENT.

        9.      NON STANDARD PRINTOUTS
                ----------------------

                "STATC" IS THE CONTENTS OF THE DISPLAY NAME REGISTER.
                THE DISPLAY NAME REGISTER CONTAINS A UNIQUE VALUE FOR EACH SUB-PICTURE.
                ALL OTHER PRINTOUTS HAVE STANDARD MEANINGS AS REPRESENTED IN
                DEC/X11 DOCUMENTATION.

        10.     ENVIROMENT
                ----------

                #1      11/10 WITH 16K OF MEMORY
                        RK-11-D DISK CONTROLLER WITH 1 DRIVE
                        VS-60 DISPLAY SYSTEM WITH ADDITIONAL CONSOLE

                #2      11/45 WITH 24K OF MEMORY  (16K CORE + 8K MOS)
                        KT-11-D MEMORY MANAGEMENT
                        RK-11-D DISK CONTROLLER WITH 1 DRIVE
                        VS-60 DISPLAY SYSTEM WITH ADDITIONAL CONSOLE

                #3      11/40 WITH 64K OF MEMORY
                        EIS/FIS
                        RK-11-D DISK CONTROLLER WITH 2 DRIVES
                        VS-60 DISPLAY SYSTEM WITH ONE CONSOLE

                -                       .LIST   SEQ,BIN
```

```
                        ;VS-60 INSTRUCTION SET

100000                           CHAR=100000                      ;DISPLAY IN CHARACTER MODE
104000                           SHORTV=104000                    ;               SHORT VECTOR
110000                           LONGV=110000                     ;               LONG VECTOR MODE
114000                           POINT=114000                     ;               POINT MODE
120000                           GRAPHX=120000                    ;               GRAPHPLOT X MODE
124000                           GRAPHY=124000                    ;               GRAPHPLOT Y MODE
120000                           BASICV=GRAPHX                    ;               BASIC VECTOR MODE
130000                           RELATP=130000                    ;               RELATIVE POINT MODE
134000                           BASICS=RELATP!4000               ;               BASIC SHORT VECTOR MODE
144000                           ABSVCT=144000                    ;               ABSOLUTE VECTOR MODE

010000                           OFFST0=10000                     ;               ENABLE OFFSET OF 0
012000                           OFFST1=12000                     ;               ENABLE OFFSET OF 1
014000                           OFFST2=14000                     ;               ENABLE OFFSET OF 2
016000                           OFFST3=16000                     ;               ENABLE OFFSET OF 3

002000                           INT0=2000                        ;ENABLE INTENSITY LEVEL 0
002200                           INT1=2200                        ;                       1
002400                           INT2=2400                        ;                       2
002600                           INT3=2600                        ;                       3
003000                           INT4=3000                        ;                       4
003200                           INT5=3200                        ;                       5
003400                           INT6=3400                        ;                       6
003600                           INT7=3600                        ;LEVEL 7

000100                           LPOFF=100
000140                           LPON=140
000020                           BLKOFF=20                        ;DISABLE BLINK
000030                           BLKON=30                         ;ENABLE BLINK

000004                           LINE0=4                          ;ENABLE LINE TYPE 0
000005                           LINE1=5                          ;ENABLE LINE TYPE 1
000006                           LINE2=6                          ;ENABLE LINE TYPE 2
000007                           LINE3=7                          ;ENABLE LINE TYPE 3

002000                           PATH0=2000                       ;DIRECTION 0
006000                           PATH1=6000                       ;DIRECTION 1
012000                           PATH2=12000                      ;DIRECTION 2
016000                           PATH3=16000                      ;DIRECTION 3
022000                           PATH4=22000                      ;        4
026000                           PATH5=26000                      ;        5
032000                           PATH6=32000                      ;        6
036000                           PATH7=36000                      ;        7

160000                           DJMP=160000                      ;DISPLAY ABSOLUTE JUMP
161000                           DJMPR=DJMP!BIT9                  ;DISPLAY RELATIVE JUMP
162000                           DJSR=DJMP!BIT10                  ;DISPLAY JSR ABSOLUTE
163000                           DJSRR=DJSR!BIT9                  ;DISPLAY JSR RELATIVE

164000                           DNOP=164000
166000                           DPOP=DNOP!BIT10                  ;POP AND RESTORE
165000                           DPOPNR=DNOP!BIT9                 ;POP AND NO RESTORE
164000                           CONSL0=DNOP                      ;CONSOLE 0
164400                           CONSL1=DNOP!BIT8                 ;CONSOLE 1
```

```

170000                           STATSA=170000
173400                           DSTOP=173400
170002                           DMENU0=STATSA!BIT1               ;DISABLE MENU
170003                           DMENU1=DMENU0!BIT0

000200                           LPLITE=200
000300                           LPDARK=300
000040                           ITAL0=40                         ;DISABLE ITALIC CHARACTERS
000060                           ITAL1=60

174000                           STATSB=174000

000100                           INCR=100                         ;ENABLE "GRAPHPLOT INCREMENT REG. CHANGE"

154000                           STATSC=154000
155000                           CHRRT0=STATSC!BIT9               ;DISABLE CHAR ROTATE
155400                           CHRRT1=CHRRT0!BIT8

154200                           CHARS0=STATSC!BIT7               ;LOAD CHARACTER SCALE TO 1/2
154240                           CHARS1=CHARS0!BIT5               ;                       1
154300                           CHARS2=CHARS0!BIT6               ;                       1 1/2
154340                           CHARS3=CHARS0!BIT6!BIT5          ;                       2

154020                           VCTR00=STATSC!BIT4               ;LOAD VECTOR SCALE REGISTER

176000                           STATE=STATSB!BIT10

176002                           STRNG0=STATE!BIT1                ;DISABLE CHARACTER STRING TERMINATE
176003                           STRNG1=STRNG0!BIT0

176040                           EDGE0=STATE!BIT5                 ;DISABLE EDGE INTERRUPT
176060                           EDGE1=EDGE0!BIT4
150000                           DNAME=150000                     ;LOAD DISPLAY NAME REGISTER

                        ;MORE EQUATES

040000                           INTX=BIT14                       ;INTENSIFY
000177                           MAXMUX=177                       ;MAX. MENU  X WIDTH
001777                           MAXX=1777                        ;MAX. X AXIS LENGTH
001777                           MAXY=1777                        ;MAX. Y AXIS LENGTH
000777                           HALFX=MAXX/2                     ;HALF OF MAXIUM LENGTH
020000                           MINUSX=20000                     ;NEGATIVE SIGN BIT
020000                           MINUSY=20000                     ;NEGATIVE SIGN BIT
000100                           MINSUY=100                       ;NEGATIVE SIGN BIT <SHORT VECTOR MODE>

000021                           SUPON=21                         ;SUPER-SCRIPT ENABLE
000023                           SUPOFF=23                        ;SUPER-SCRIPT DISABLE
000022                           SUBON=22                         ;SUB-SCRIPT ENABLE
000024                           SUBOFF=24                        ;SUB-SCRIPT DISABLE
```

```
000000'                            IOMODP  <VSAC >,172000,320,4,,,0,75
000000'                            MODULE  142000,VSAC ,172000,320,4,,,0,75,,,
                                   .TITLE  VSAC DEC/X11 SYSTEM EXERCISER MODULE
                          ;        DDXCOM  VERSION  6      23-MAY-78
                                   .LIST   BIN
                          ;******************************************************************
000000'                   BEGIN:
000000' 051526  041501     040  MODNAM: .ASCII  /VSAC / ;MODULE NAME.
000005'    000            XFLAG:   .BYTE   OPEN             ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 172000            ADDR:    172000+0                 ;1ST DEVICE ADDR.
000010' 000320            VECTOR:  320+0                    ;1ST DEVICE VECTOR.
000012'    200            BR1:     .BYTE   PRTY4+0          ;1ST BR LEVEL.
000013'    000            BR2:     .BYTE   PRTY+0           ;2ND BR LEVEL.
000014' 000001            DVID1:   +1                       ;DEVICE INDICATOR 1.
000016' 000000            SR1:     OPEN                     ;SWITCH REGISTER 1
000020' 000000            SR2:     OPEN                     ;SWITCH REGISTER 2
000022' 000000            SR3:     OPEN                     ;SWITCH REGISTER 3
000024' 000000            SR4:     OPEN                     ;SWITCH REGISTER 4
                          ;******************************************************************
000026' 142000            STAT:    142000                   ;STATUS WORD.
000030' 000316'           INIT:    START                    ;MODULE START ADDR.
000032' 000224'           SPOINT:  MODSP                    ;MODULE STACK POINTER.
000034' 000000            PASCNT:  0                        ;PASS COUNTER.
000036' 000000            ICONT:   0                        ;# OF ITERATIONS PER PASS=0
000040' 000000            ICOUNT:  0                        ;LOC TO COUNT ITERATIONS
000042' 000000            SOFCNT:  0                        ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000            HRDCNT:  0                        ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000            SOFPAS:  0                        ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000            HRDPAS:  0                        ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000            SYSCNT:  0                        ;# OF SYS ERRORS ACCUMULATED
000054' 000000            RANNUM:  0                        ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                   CONFIG:                           ;RESERVED FOR MONITOR USE
000056' 000000            RES1:    0                        ;RESERVED FOR MONITOR USE
000060' 000000            RES2:    0                        ;RESERVED FOR MONITOR USE
000062' 000000            SVR0:    OPEN                     ;LOC TO SAVE R0.
000064' 000000            SVR1:    OPEN                     ;LOC TO SAVE R1.
000066' 000000            SVR2:    OPEN                     ;LOC TO SAVE R2.
000070' 000000            SVR3:    OPEN                     ;LOC TO SAVE R3.
000072' 000000            SVR4:    OPEN                     ;LOC TO SAVE R4.
000074' 000000            SVR5:    OPEN                     ;LOC TO SAVE R5.
000076' 000000            SVR6:    OPEN                     ;LOC TO SAVE R6.
000100' 000000            CSRA:    OPEN                     ;ADDR OF CURRENT CSR.
000102'                   SBADR:                            ;ADDR OF GOOD DATA, OR
000102' 000000            ACSR:    OPEN                     ;CONTENTS OF CSR.
000104'                   WASADR:                           ;ADDR OF BAD DATA, OR
000104' 000000            ASTAT:   OPEN                     ;STATUS REG CONTENTS.
000106'                   ERRTYP:                           ;TYPE OF ERROR
000106' 000000            ASB:     OPEN                     ;EXPECTED DATA.
000110' 000000            AWAS:    OPEN                     ;ACTUAL DATA.
000112' 000430'           RSTRT:   RESTRT                   ;RESTART ADDRESS AFTER END OF PASS
000114' 000000            WDTO:    OPEN                     ;WORDS TO MEMORY PER ITERATION
000116' 000000            WDFR:    OPEN                     ;WORDS FROM MEMORY PER ITERATION
000120' 000000            INTR:    OPEN                     ;# OF INTERRUPTS PER ITERATION
000122' 000075            IDNUM:   75                       ;MODULE IDENTIFICATION NUMBER=75
        000040                     .REPT   SPSIZ            ;MODULE STACK STARTS HERE.
                                   .NLIST
```

```
                                   .WORD   0
                                   .LIST
                                   .ENDR
000224'                   MODSP:
                          ;******************************************************************


000224' 003144'           RBUFVA:  FRAME0
000226' 000000            RBUFPA:  OPEN
000230' 000000            RBUFEA:  OPEN
```

```
                                     ;GTPASS:        1                              ;PASS COUNTER
000232' 000003                       DELAY:  3                              ;PICTURE MOTION DELAY FACTOR
000234' 000020                       DELAY1: 20                             ;GRAPHPLOT MOTION DELAY FACTOR
000236' 172000                       GTPC:   172000                         ;DISPLAY PC (D.P.C.)
000240' 172002                       GTSR:   172002                         ;DISPLAY STATUS REG.
000242' 172004                       GTXPOS: 172004                         ;DISPLAY X REGISTER
000244' 172006                       GTYPOS: 172006                         ;DISPLAY Y REGISTER
000246' 172010                       GTREL:  172010                         ;DISPLAY RELOCATE REG.
000250' 172012                       GTSR1:  172012                         ;MISC. STATUS REG. #1
000252' 172014                       GTXOFF: 172014                         ;DISPLAY X POS OFFSET REG
000254' 172016                       GTYOFF: 172016                         ;DISPLAY Y POS OFFSET REG
000256' 172020                       GTASNA: 172020                         ;DISPLAY ASSOC. NAME REG.
000260' 172022                       GTCONS: 172022                         ;DISPLAY CONSOLE STATUS REGISTER
000262' 172024                       GTNAME: 172024                         ;DISPLAY NAME REGISTER
000264' 172026                       GTSTAK: 172026                         ;DISPLAY STACK CONTENTS
000266' 172030                       GTTERM: 172030                         ;DISPLAY CHARACTER TERMINATE REG.
000270' 172032                       GTSPPT: 172032                         ;DISPLAY STACK POINTER REG.
000272' 172034                       GTZPOS: 172034                         ;DISPLAY Z POS REG.
000274' 172036                       GTZOFF: 172036                         ;DISPLAY Z OFFSET REG.
000276' 000320                       GTDONE: 320                            ;DISPLAY DONE VECTOR
000300' 000322                       GTDNE1: 322
000302' 000324                       GTLPH:  324                            ;DISPLAY LIGHT-PEN VECTOR
000304' 000326                       GTLPH1: 326
000306' 000330                       GTSOTM: 330                            ;DISPLAY SHIFT-OUT/ TIME-OUT VECTOR
000310' 000332                       GTSOT1: 332
000312' 000334                       GTNAMM: 334                            ;DISPLAY NAME MATCH VECTOR
000314' 000336                       GTNAM1: 336

                                     ;INITILIZE VS-60 ADDRESSES AND VECTORS

000316' 005767  177512               START:  TST     PASCNT          ;HAS A PASS BEEN MADE YET?
000322' 001042                               BNE     RESTRT          ;YES LEAVE
000324' 032767  000001  177464               BIT     #BIT0,SR1       ;NO - IS NOP OR INSTRUCT DESIRED
000332' 001021                               BNE     2$              ;BR IF NOP
000334' 012767  000707  177554               MOV     #455.,WDFR      ;455 WORDS FROM MEM/ITERATION
000342' 012767  000010  177550               MOV     #8.,INTR                ;8 INTERRUPTS/ITERATION
000350' 005767  177462                       TST     ICONT           ;IS ICONT ZERO
000354' 001404                               BEQ     1$              ;YES BR TO SET TO 1
000356' 026727  177454  005000               CMP     ICONT,#5000     ;IS IT LESS THAN 5000
000364' 002421                               BLT     RESTRT          ;OPERATOR MUST HAVE ALREADY SET IT UP
000366' 012767  000001  177442       1$:     MOV     #1,ICONT        ;IT WAS ZERO - SET TO 1
000374' 000415                               BR      RESTRT
000376' 012767  000001  177512       2$:     MOV     #1,WDFR ;1 WORD FR MEM/ITERATION
000404' 012767  000001  177506               MOV     #1,INTR         ;1 INTERRUPT/ITERATION
000412' 026727  177420  005000               CMP     ICONT,#5000     ;IS IT GREATER THAN ZERO
000420' 003003                               BGT     RESTRT
000422' 012767  060000  177406               MOV     #60000,ICONT    ;NO - USE 60000 TO START WITH
000430' 012701  000236'              RESTRT: MOV     #GTPC,R1                ;LOAD POINTER
000434' 016700  177346                       MOV     ADDR,R0                 ;LOAD VALUE
000440' 010021                       1$:     MOV     R0,(R1)+                ;LOAD VALUE INTO ADDRESS
000442' 005720                               TST     (R0)+                   ;TEST FOR BUS ERROR AND UPDATE R0
000444' 022701  000276'                      CMP     #GTDONE,R1              ;TEST IF DONE ADDRESS SETUP
000450' 001373                               BNE     1$                      ;BR IF NOT
000452' 016700  177332                       MOV     VECTOR,R0               ;LOAD VECTOR ADDRESS
000456' 010021                       2$:     MOV     R0,(R1)+                ;LOAD VECTOR VALUE
000460' 005720                               TST     (R0)+                   ;ADJUST R0
```

```
000462' 022701  000316'                      CMP     #START,R1               ;TEST IF DONE VECTOR SETUP
000466' 001373                               BNE     2$                      ;BR IF NOT
000470' 104415  000000' 000224'              GETPAS,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
000476' 016700  177526                       MOV     RBUFEA,R0               ;GET EA BITS
000502' 006000                               ROR     R0
000504' 006000                               ROR     R0
000506' 000300                               SWAB    R0                      ;MOVE BITS 4 + 5 INTO 10 AND 11
000510' 042700  171777                       BIC     #171777,R0              ;MASK
000514' 010067  002376                       MOV     R0,EABITS               ;SAVE MY EA BITS
000520' 016767  177502  005414               MOV     RBUFPA,FILEOD           ;LOAD PHYSICAL ADDRESS OF THE STARTING LOC. OF T
000526' 062767  000004  005406               ADD     #4,FILEOD               ;UPDATE ADDRESS
000534' 005067  002376                       CLR     ABORT                   ;CLEAR ABORT TESTING FLAG
```

```
                                          ;TEST THAT THE X DYNAMIC OFFSET REGISTER CAN BE LOADED

000540' 012767  004000  177336    XDOFF:  MOV     #BIT11,ASTAT            ;LOAD EXPECTED VALUE
000546' 016767  177500  177324            MOV     GTXOFF,CSRA             ;LOAD THE BUS ADDRESS
000554' 016777  177324  177470    1$:     MOV     ASTAT,@GTXOFF           ;LOAD THE REGISTER
000562' 017767  177464  177312            MOV     @GTXOFF,ACSR            ;READ THE REGISTER
000570' 042767  170000  177304            BIC     #170000,ACSR            ;MASK TO OTHER BITS
000576' 026767  177302  177276            CMP     ASTAT,ACSR              ;TEST IF EQUAL
000604' 001411                            BEQ     2$                      ;BR IF SAME
000606' 012767  000025  177272            MOV     #25,ERRTYP      ;BIT STUCK
                                          ;**************************************************************
000614' 104405  000000' 000000            HRDER$,BEGIN,NULL       ; X DYNAMIC OFFSET REGISTER FAILED TO LOAD PROPERLY
                                          ;**************************************************************
000622' 052767  000001  002306            BIS     #BIT0,ABORT             ;INDICATE "MAJOR ERROR"
000630'                           2$:
000630' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000634' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000640' 006267  177240                    ASR     ASTAT                   ;ADJUST DATA PATTERN
000644' 001343                            BNE     1$                      ;BR IF MORE BITS TO TEST
000646' 005077  177400                    CLR     @GTXOFF                 ;ENSURE CLEAR REGISTER

                                          ;TEST THAT THE Y DYNAMIC OFFSET REGISTER CAN BE LOADED

000652' 012767  004000  177224    YDOFF:  MOV     #BIT11,ASTAT            ;LOAD EXPECTED VALUE
000660' 016767  177370  177212            MOV     GTYOFF,CSRA             ;LOAD THE BUS ADDRESS
000666' 016777  177212  177360    1$:     MOV     ASTAT,@GTYOFF           ;LOAD THE REGISTER
000674' 017767  177354  177200            MOV     @GTYOFF,ACSR            ;READ THE REGISTER
000702' 042767  170000  177172            BIC     #170000,ACSR            ;MASK TO OTHER BITS
000710' 026767  177170  177164            CMP     ASTAT,ACSR              ;TEST IF EQUAL
000716' 001411                            BEQ     2$                      ;BR IF SAME
000720' 012767  000025  177160            MOV     #25,ERRTYP      ;BIT STUCK
                                          ;**************************************************************
000726' 104405  000000' 000000            HRDER$,BEGIN,NULL       ; Y DYNAMIC OFFSET REGISTER FAILED TO LOAD PROPERLY
                                          ;**************************************************************
000734' 012767  000002  002174            MOV     #BIT1,ABORT             ;INDICATE "MAJOR ERROR"
000742'                           2$:
000742' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000746' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000752' 006267  177126                    ASR     ASTAT                   ;ADJUST DATA PATTERN
000756' 001343                            BNE     1$                      ;BR IF MORE BITS TO TEST
000760' 005077  177270                    CLR     @GTYOFF                 ;ENSURE CLEAR REGISTER
```

```
                                          ;TEST THAT THE RELOCATE REGISTER CAN BE LOADED

000764' 012767  004000  177112    RELTST: MOV     #BIT11,ASTAT            ;LOAD EXPECTED VALUE
000772' 016767  177250  177100            MOV     GTREL,CSRA              ;LOAD THE BUS ADDRESS
001000' 016777  177100  177240    1$:     MOV     ASTAT,@GTREL            ;LOAD THE REGISTER
001006' 017767  177234  177066            MOV     @GTREL,ACSR             ;READ THE REGISTER
001014' 042767  170000  177060            BIC     #170000,ACSR            ;MASK TO OTHER BITS
001022' 026767  177056  177052            CMP     ASTAT,ACSR              ;TEST IF EQUAL
001030' 001411                            BEQ     2$                      ;BR IF SAME
001032' 012767  000025  177046            MOV     #25,ERRTYP      ;BIT STUCK
                                          ;**************************************************************
001040' 104405  000000' 000000            HRDER$,BEGIN,NULL       ; RELOCATE FAILED TO LOAD PROPERLY
                                          ;**************************************************************
001046' 052767  000004  002062            BIS     #BIT2,ABORT             ;INDICATE "MAJOR ERROR"
001054'                           2$:
001054' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001060' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001064' 006267  177014                    ASR     ASTAT                   ;ADJUST DATA PATTERN
001070' 001343                            BNE     1$                      ;BR IF MORE BITS TO TEST
001072' 005077  177150                    CLR     @GTREL                  ;ENSURE CLEAR REGISTER

                                          ;TEST THAT THE D.P.C. REGISTER CAN BE LOADED

001076' 012777  010000  177164    DPCTST: MOV     #BIT12,@GTSPPT          ;SET MAINT SW #1
001104' 012767  000002  176772            MOV     #BIT1,ASTAT             ;LOAD EXPECTED VALUE
001112' 016767  177120  176760            MOV     GTPC,CSRA               ;LOAD THE BUS ADDRESS
001120' 016777  176760  177110    1$:     MOV     ASTAT,@GTPC             ;LOAD THE REGISTER
001126' 017767  177104  176746            MOV     @GTPC,ACSR              ;READ THE REGISTER
001134' 026767  176744  176740            CMP     ASTAT,ACSR              ;TEST IF EQUAL
001142' 001411                            BEQ     2$                      ;BR IF SAME
001144' 012767  000025  176734            MOV     #25,ERRTYP      ;BIT STUCK
                                          ;**************************************************************
001152' 104405  000000' 000000            HRDER$,BEGIN,NULL       ; D.P.C. FAILED TO LOAD PROPERLY
                                          ;**************************************************************
001160' 052767  000010  001750            BIS     #BIT3,ABORT             ;INDICATE "MAJOR ERROR"
001166'                           2$:
001166' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001172' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001176' 006367  176702                    ASL     ASTAT                   ;ADJUST DATA PATTERN
001202' 001346                            BNE     1$                      ;BR IF MORE BITS TO TEST
001204' 005077  177026                    CLR     @GTPC                   ;ENSURE CLEAR REGISTER
001210' 005077  177054                    CLR     @GTSPPT                 ;CLEAR MAINT SW #1

                                  ;NOW TEST IF ANY "MAJOR ERRORS" HAVE BEEN FOUND
                                  ;       AND DROP MODULE IF ANY WERE PRESENT

001214' 005767  001716            DONTST: TST     ABORT                   ;TEST FOR "MAJOR ERRORS"
001220' 001402                            BEQ     BEGNA                   ;BR IF NONE AND START DISPLAY SECTION
001222' 104410  000000'                   END$,BEGIN              ;DROP MODULE BECAUSE OF A FATAL REGISTER ERROR
```

```
                                ;TEST IF "DISPLAY NOP MODE"
001226' 005767  001670          BEGNA:  TST     FAST                    ;TEST IF RUNNING "FAST" MODE
001232' 001015                          BNE     2$                      ;BR IF YES
001234' 032767  000001  176554          BIT     #BIT0,SR1               ;TEST IF BIT 0 OF SR1 = 1
001242' 001420                          BEQ     3$                      ;BR IF CLEARED
001244' 005267  001652                  INC     FAST                    ;SET FLAG
001250' 012700  003144'                 MOV     #FRAME0,R0              ;LOAD POINTER
001254' 012720  164000          1$:     MOV     #DNOP,(R0)+             ;LOAD DISPLAY NOP INTO BUFFER
001260' 022700  006136'                 CMP     #FILE0E,R0              ;TEST IF AT END
001264' 001373                          BNE     1$                      ;BR IF NOT
001266' 012777  002174' 177002  2$:     MOV     #NOPDON,@GTDONE         ;LOAD RETURN VECTOR ON STOP INTR.
001274' 116777  176512  176776          MOVB    BR1,@GTDNE1             ;LOAD RETURN BR LEVEL
001302' 000523                          BR      5$                      ;BR TO START DISPLAY

                                ;NORMAL INTERRUPT VECTOR SETUP

001304' 012777  001572' 176764  3$:     MOV     #GTSTOP,@GTDONE         ;LOAD STOP VECTOR
001312' 116777  176474  176760          MOVB    BR1,@GTDNE1
001320' 012777  002210' 176754          MOV     #GTLPEN,@GTLPH          ;LOAD LIGHT-PEN VECTOR
001326' 116777  176460  176750          MOVB    BR1,@GTLPH1
001334' 012777  002544' 176744          MOV     #GTSHIF,@GTSOTM         ;LOAD SHIFT-OUT VECTOR
001342' 116777  176444  176740          MOVB    BR1,@GTSOT1
001350' 012777  002710' 176734          MOV     #GTMACH,@GTNAMM         ;LOAD NAME MATCH VECTOR
001356' 116777  176430  176730          MOVB    BR1,@GTNAM1
001364' 016767  176642  001534          MOV     DELAY,GTDLY0            ;LOAD PICTURE MOTION DELAY FACTOR
001372' 016767  176636  001530          MOV     DELAY1,GTDLY1           ;LOAD GRAPHPLOT DELAY FACTOR
001400' 005067  001514                  CLR     MOTION                  ;CLEAR "STOP MOTION" FLAG
001404' 005767  001514                  TST     FIRST                   ;TEST IF FIRST TIME EVER
001410' 001011                          BNE     4$                      ;BR IF NOT
001412' 016767  004332  001512          MOV     FILE0A,JMPF0B           ;SAVE DISPLAY JUMP RELATIVE
001420' 016767  004362  001506          MOV     FILE0B,JMPF0C           ;       "
001426' 012767  000001  001470          MOV     #1,FIRST                ;ENSURE NOT THE FIRST TIME IS SET
001434' 016767  001472  004306  4$:     MOV     JMPF0B,FILE0A           ;LOAD RELATIVE JUMP OVER LIGHT PEN #0 MESSAGE
001442' 016767  001466  004336          MOV     JMPF0C,FILE0B           ;      "      "     "     "    " #1
001450' 012767  174104  001560          MOV     #STATSB!INCR+4,GRPINC   ;LOAD GRAPHPLOT INCREMENT VALUE
001456' 012777  000040  176604          MOV     #BIT5,@GTSPPT           ;RESET DPU STACK POINTER
001464' 016767  004456  003266          MOV     UPMSG,PENSW0            ;RESET PEN SWITCH MESSAGE FOR #0
001472' 016767  004450  003334          MOV     UPMSG,PENSW1            ;      "     "      "      "  #1
001500' 012767  002000  001432          MOV     #2000,DLTXRG            ;LOAD VALUE TO BE LOADED INTO X DYNAMIC OFFSET R
001506' 012767  002000  001426          MOV     #2000,DLTYRG            ;LOAD VALUE TO BE LOADED INTO Y DYNAMIC OFFSET R
001514' 016777  001420  176530          MOV     DLTXRG,@GTXOFF          ;LOAD X DYNAMIC OFFSET
001522' 016777  001414  176524          MOV     DLTYRG,@GTYOFF          ;LOAD Y DYNAMIC OFFSET
001530' 012767  002000  001354          MOV     #BIT10,ANAME            ;PRESET THE ASSOCITIVE NAME VALUE
001536' 012767  152000  003160          MOV     #DNAME!BIT10,NMATCH     ;PRESET THE ASSOCITIVE NAME MATCH
                                                                        ;DISPLAY INSTRUCTION
001544' 012777  056000  176504          MOV     #BIT14!BIT12!BIT11!BIT10,@GTASNA      ;ENABLE NAME MATCH INTERRUPT
001552' 016777  001340  176466  5$:     MOV     EABITS,@GTREL           ;LOAD D.P.U. RELOCATE REG.
001560' 016777  176442  176450          MOV     RBUFPA,@GTPC            ;START DISPLAY
001566' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                ;RETURN HERE IF INTERRUPT VIA DONE (STOP) FLAG AND BIT 0 OF SR1 IS CLEARED

001572'                         GTSTOP:
                                        ;----------------------------------------------------------------
001572' 000004  000000' 001600'         PIRQ$,BEGIN,STOPRT              ; QUEUE UP TO CONTINUE AT STOPRT AND RTI
                                        ;----------------------------------------------------------------
001600' 017767  176456  176276  STOPRT: MOV     @GTNAME,ASTAT           ;READ NAME REGISTER
001606' 016767  176426  176264          MOV     GTSR,CSRA               ;LOAD BUS ADDRESS OF STATUS REG #1
001614' 017767  176420  176260          MOV     @GTSR,ACSR              ;READ THE STATUS REGISTER
001622' 100133                          BPL     EXSTOP                  ;BR IF EXTERNAL STOP EVENT

                                ;TEST THAT A NAME MATCH INTERRUPT HAS OCCURRED BEFORE STOP INTERRUPT

001624' 005767  001264                  TST     NAMESW                  ;TEST NAME INTR. SOFT FLAG
001630' 001006                          BNE     1$                      ;BR IF NAME INTR. OCCURRED
001632' 012767  000023  176246          MOV     #23,ERRTYP      ;DEV FAILED TO INTERRUPT
                                        ;************************************************************
001640' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;NAME MATCH INTERRUPT FAILED TO OCCUR BEFORE STOP INTERR
                                        ;************************************************************

001646' 005067  001242          1$:     CLR     NAMESW                  ;CLEAR NAME INTR. SOFT FLAG
001652' 005767  001242                  TST     MOTION                  ;TEST IF "STOP MOTION" IS SET
001656' 001074                          BNE     RESTR                   ;BR IF YES
001660' 005367  001242                  DEC     GTDLY0                  ;DECREMENT PICTURE MOTION DELAY
001664' 001071                          BNE     RESTR
001666' 016767  176340  001232          MOV     DELAY,GTDLY0            ;RESET PICTURE MOTION DELAY
001674' 032767  020000  001236          BIT     #BIT13,DLTXRG           ;TEST IF NEGATIVE POLARITY ?
001702' 001020                          BNE     2$                      ;BR IF NEG.
001704' 162767  000001  001226          SUB     #1,DLTXRG               ;ADJUST X DYNAMIC OFFSET
001712' 162767  000001  001222          SUB     #1,DLTYRG               ;ADJUST Y DYNAMIC OFFSET
001720' 026727  001214  000000          CMP     DLTXRG,#0               ;TEST IF FINISHED ALL POS. OFFSETS ?
001726' 001031                          BNE     3$                      ;BR IF NOT
001730' 012767  020000  001202          MOV     #MINUSX,DLTXRG          ;PRESET X OFFSET VALUE
001736' 012767  020000  001176          MOV     #MINUSX,DLTYRG          ;PRESET Y OFFSET VALUE
001744' 062767  000001  001166  2$:     ADD     #1,DLTXRG               ;UPDATE X DYNAMIC OFFSET VALUE
001752' 062767  000001  001162          ADD     #1,DLTYRG               ;UPDATE Y DYNAMIC OFFSET VALUE
001760' 026727  001154  024000          CMP     DLTXRG,#MINUSX!4000     ;TEST IF FINISHED ALL NEG. OFFSETS ?
001766' 001011                          BNE     3$                      ;BR IF NOT
001770' 012767  002000  001142          MOV     #2000,DLTXRG            ;RELOAD X DYNAMIC OFFSET VALUE
001776' 012767  002000  001136          MOV     #2000,DLTYRG            ;RELOAD Y DYNAMIC OFFSET VALUE
002004' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
002010' 000417                          BR      RESTR
```

```


                                 ;TEST IF TIME TO MOVE THE "SINE WAVE"

002012' 005367  001112           3$:      DEC     GTDLY1                  ;DEC COUNTER
002016' 001014                            BNE     RESTR                   ;BR IF NOT TIME TO MOVE IT
002020' 016767  176210  001102            MOV     DELAY1,GTDLY1           ;RESET DELAY
002026' 005267  001204                    INC     GRPINC                  ;UPDATE GRAPH INCREMENT
002032' 022767  174110  001176            CMP     #STATSB!INCR+10,GRPINC  ;TEST FOR INCREMENT
002040' 001003                            BNE     RESTR                   ;BRANCH IF NOT
002042' 012767  174100  001166            MOV     #STATSB!INCR,GRPINC     ;RESET GRAPH INCREMENT
002050' 016767  001056  003672   RESTR:   MOV     JMPF0B,FILE0A           ;RESET LIGHT PEN #0 MESSAGE
002056' 016767  001052  003722            MOV     JMPF0C,FILE0B           ;RESET LIGHT PEN #1 MESSAGE
002064' 016777  001050  176160            MOV     DLTXRG,@GTXOFF          ;LOAD X DYNAMIC OFFSET
002072' 016777  001044  176154            MOV     DLTYRG,@GTYOFF          ;LOAD Y DYNAMIC OFFSET
002100' 012777  000001  176130   CONT:    MOV     #1,@GTPC                ;RESUME THE DISPLAY
002106' 104400  000000'                   EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

                                 ;RETURN HERE IS THE INTERNAL STOP FLAG WAS NOT SET

002112' 016767  176132  175760   EXSTOP:  MOV     GTSR1,CSRA              ;LOAD BUS ADDRESS
002120' 017767  176124  175754            MOV     @GTSR1,ACSR             ;READ REGISTER VALUE
002126' 105767  175750                    TSTB    ACSR                    ;TEST IF SET
002132' 100010                            BPL     1$                      ;BR IF NOT
002134' 012767  000011  175744            MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                          ;************************************************************
002142' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;UNEXPECTED EXTERNAL STOP  INTERRUPT
                                          ;************************************************************
002150' 000167  176142                    JMP     START                   ;START AGAIN
002154' 012767  000011  175724   1$:      MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                          ;************************************************************
002162' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;STOP INTERRPUT BUT NO FLAG WAS SET
                                          ;************************************************************
002170' 000167  176122                    JMP     START                   ;START AGAIN

                                 ;RETURN HERE IF INTERRUPT VIA DONE (STOP) FLAG AND BIT 0 OF SR1 IS SET
002174'                          NOPDON:
                                          ;-------------------------------------------------------------
002174' 000004  000000' 002202'           PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                          ;-------------------------------------------------------------
002202'                          1$:
002202' 104413  000000'                   ENDIT$,BEGIN            ;SIGNAL END OF ITERATION,
                                                                  ;MONITOR SHALL TEST END OF PASS
002206' 000734                            BR      CONT
```

```
                                 ;RETURN HERE IF INTERRUPT VIA LIGHT PEN FLAG, SWITCH, OR EDGE INTERRUPT
                                 ;TEST IF "EDGE INTERRUPT" FLAG IS SET
002210'                          GTLPEN:
                                          ;-------------------------------------------------------------
002210' 000004  000000' 002216'           PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                          ;-------------------------------------------------------------
002216' 017767  176016  175656   1$:      MOV     @GTSR,ACSR              ;READ STARUS REG FOR EDGE FRAL
002224' 017767  176032  175652            MOV     @GTNAME,ASTAT           ;READ DISPLAY NAME
002232' 016767  176002  175640            MOV     GTSR,CSRA               ;LOAD BUS ADDRESS OF STATUS REGISTER
002240' 005067  000276                    CLR     20$                     ;CLEAR FLAG HAPPENED  SWITCH
002244' 032767  000004  175630            BIT     #BIT2,ACSR              ;TEST IF "EDGE" IS SET
002252' 001410                            BEQ     2$                      ;BR IF NOT
002254' 012767  000011  175624            MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                          ;************************************************************
002262' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;UNEXPECTED EDGE FLAG INTERRUPT
                                          ;************************************************************
002270' 000167  176022                    JMP     START                   ;START AGAIN
                                 ;TEST IF LIGHT-PEN FLAG OR LIGHT PEN SWITCH FLAG
002274' 017767  175760  175600   2$:      MOV     @GTCONS,ACSR            ;READ CONSOLE STATUS REG
002302' 016767  175752  175570            MOV     GTCONS,CSRA             ;LOAD BUS ADDRESS
002310' 032767  040000  175564            BIT     #BIT14,ACSR             ;TEST IF LP FLAG #0 SET
002316' 001405                            BEQ     3$                      ;BR IF NTO
002320' 012767  164000  003422            MOV     #DNOP,FILE0A            ;ENABLE LP HIT MESSAGE ON CONSOLE #0
002326' 005267  000210                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002332' 032767  020000  175542   3$:      BIT     #BIT13,ACSR             ;TEST IF SW #0 DOWN
002340' 001410                            BEQ     4$                      ;BR IF NOT SET
002342' 016767  003576  002410            MOV     DWNMSG,PENSW0           ;CHANGE MESSAGE ON CONSOLE #0
002350' 005267  000544                    INC     MOTION                  ;SET "STOP MOTION" FLAG
002354' 005267  000162                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002360' 000413                            BR      5$
002362' 032767  010000  175512   4$:      BIT     #BIT12,ACSR             ;TEST IF SW #0 UP ON
002370' 001407                            BEQ     5$                      ;BR IF NOT
002372' 016767  003550  002360            MOV     UPMSG,PENSW0            ;CHANGE MESSAGE ON CONSOLE #0
002400' 005067  000514                    CLR     MOTION                  ;CLEAR "STOP MOTION" FLAG
002404' 005267  000132                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002410' 032767  000400  175464   5$:      BIT     #BIT8,ACSR              ;TEST IF LP FALG #1 SET
002416' 001405                            BEQ     6$                      ;BR IF NOT
002420' 012767  164000  003360            MOV     #DNOP,FILE0B            ;ENABLE LP HIT MESSAGE ON CONSOLE #1
002426' 005267  000110                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002432' 032767  000200  175442   6$:      BIT     #BIT7,ACSR              ;TEST IF SW #1 DOWN ON
002440' 001410                            BEQ     7$                      ;BR IF NOT
002442' 016767  003476  002364            MOV     DWNMSG,PENSW1           ;CHANGE MESSAGE ON CONSOLE #1
002450' 005267  000444                    INC     MOTION                  ;SET "STOP MOTION" FLAG
002454' 005267  000062                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002460' 000413                            BR      10$
002462' 032767  000100  175412   7$:      BIT     #BIT6,ACSR              ;TEST IF SW #1 UP ON
002470' 001407                            BEQ     10$                     ;BR IF NOT
002472' 016767  003450  002334            MOV     UPMSG,PENSW1            ;CHANGE MESSAGE ON CONSOLE #1
002500' 005067  000414                    CLR     MOTION                  ;CLEAR "STOP MOTION" FLAG
002504' 005267  000032                    INC     20$                     ;SET FLAG HAPPENED SWITCH
002510' 005767  000026           10$:     TST     20$                     ;TEST IF FLAG HAS BEEN SERVICED ?
002514' 001402                            BEQ     24$                     ;BR IF IT HAS NOT BEEN SENSED
002516' 000167  177356                    JMP     CONT
002522' 012767  000011  175356   24$:     MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                          ;************************************************************
002530' 104405  000000' 000000            HRDER$,BEGIN,NULL       ;INTERRUPT DETECTED BUT NO FLAG WAS SET
```

```
                                        ;***************************************************************
002536' 000167  175554                  JMP     START                   ;START AGAIN
002542' 000000                  20$:    0                               ;NON-ZERO IF THE FLAG WAS KNOWN

                                ;RETURN HERE IF A MISC. VS60 INTERRUPT

002544'                         GTSHIF:
                                        ;---------------------------------------------------------------
002544' 000004  000000' 002552'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;---------------------------------------------------------------
002552' 017767  175504  175324  1$:     MOV     @GTNAME,ASTAT           ;READ D.P.U. NAME REGISTER
002560' 016767  175454  175314          MOV     GTSR,ACSR               ;LOAD BUS ADDRESS
002566' 017767  175446  175306          MOV     @GTSR,ACSR              ;READ REGISTER
002574' 032767  000100  175300          BIT     #BIT6,ACSR              ;TEST IF "SHIFT-OUT" FLAG WAS SET
002602' 001410                          BEQ     2$                      ;BR IF NOT
002604' 012767  000044  175274          MOV     #44,ERRTYP      ;FLAG SHOULD NOT BE SET
                                        ;***************************************************************
002612' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;UNEXPECTED SHIFT-OUT FLAG SET
                                        ;***************************************************************
002620' 000167  175472                  JMP     START                   ;START AGAIN
002624' 016767  175420  175246  2$:     MOV     GTSR1,CSRA              ;LOAD BUS ADDRESS
002632' 017767  175412  175242          MOV     @GTSR1,ACSR             ;READ REGISTER
002640' 042767  003515  175234          BIC     #3515,ACSR              ;MASK TO UNWANTED BITS
002646' 001410                          BEQ     3$                      ;BR IF NONE
002650' 012767  000044  175230          MOV     #44,ERRTYP      ;FLAG SHOULD NOT BE SET
                                        ;***************************************************************
002656' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;VS-60 MAJOR ERROR FLAG WAS SET
                                        ;***************************************************************
002664' 000167  175426                  JMP     START                   ;START AGAIN
002670' 012767  000011  175210  3$:     MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;***************************************************************
002676' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INTERRUPT DETECTED NUT NO FLAG WAS SET
                                        ;***************************************************************
002704' 000167  175406                  JMP     START                   ;START AGAIN
```

```

                                ;RETURN HERE IF NAME MATCH INTERRUPT

002710'                         GTMACH:
                                        ;---------------------------------------------------------------
002710' 000004  000000' 002716'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;---------------------------------------------------------------
002716' 017767  175340  175160  1$:     MOV     @GTNAME,ASTAT           ;READ NAME REGISTER
002724' 016767  000162  175150          MOV     ANAME,ACSR              ;READ REGISTER AGAIN
002732' 016767  175324  175140          MOV     GTNAME,CSRA             ;LOAD BUS ADDRESS
002740' 005767  175140                  TST     ASTAT                   ;TEST IF NAME MATCH FLAG IS SET ?
002744' 100406                          BMI     4$                      ;BR IF YES
002746' 012767  000011  175132          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;***************************************************************
002754' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;INTERRUPT DETECTED BUT NO FLAG WAS SET
                                        ;***************************************************************
002762' 042767  170000  175114  4$:     BIC     #170000,ASTAT           ;MASK TO BITS
002770' 026767  175110  175104          CMP     ASTAT,ACSR              ;TEST IF EXPECTED
002776' 001410                          BEQ     2$                      ;BR IF SAME
003000' 012767  000011  175100          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;***************************************************************
003006' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;UNEXPECTED NAME MATCH INTERRUPT
                                        ;***************************************************************
003014' 000167  175276                  JMP     START                   ;START AGAIN
003020' 006267  000066          2$:     ASR     ANAME                   ;ADJUST THE EXPECTED NAME VALUE
003024' 022767  000040  000060          CMP     #BIT5,ANAME             ;TEST IF COMPLETE
003032' 001003                          BNE     3$                      ;BR IF NOT
003034' 012767  002000  000050          MOV     #BIT10,ANAME            ;YES, RESET EXPECTED NAME VALUE
003042' 016767  000044  000040  3$:     MOV     ANAME,10$               ;COPY NAME
003050' 052767  054000  000032          BIS     #BIT14!BIT12!BIT11,10$  ;ADD SEARCH CODE AND ENABLE
003056' 016777  000026  175172          MOV     10$,@GTASNA             ;LOAD ASSOCITIVE NAME REGISTER
003064' 016767  000022  001632          MOV     ANAME,NMATCH            ;LOAD THE DISPLAY BUFFER VALUE
003072' 052767  150000  001624          BIS     #DNAME,NMATCH           ; LOCATION
003100' 005267  000010                  INC     NAMESW                  ;SET SOFT SWITCH SAYING MATCH OCCURRED
003104' 000167  176770                  JMP     CONT                    ;CONTINUE THE SUP-PICTURE
003110' 056000                  10$:    BIT14!BIT12!BIT11!BIT10         ;SEARCH CODE AND DISPLAY NAME VALUE
003112' 002000                  ANAME:  BIT10
003114' 000000                  NAMESW: 0

003116' 000000                  EABITS: 0
003120' 000000                  MOTION: 0                               ;NON-ZERO IF STOP MOTION
003122' 000000                  FAST:   0
003124' 000000                  FIRST:  0
003126' 000100                  GTDLY0: 100
003130' 000100                  GTDLY1: 100
003132' 000000                  JMPF0B: 0
003134' 000000                  JMPF0C: 0
003136' 000000                  ABORT:  0
003140' 000000                  DLTXRG: 0
003142' 000000                  DLTYRG: 0
```

```
                                ;FRAME  CX,     OY
003144' 164374                  FRAME0: CONSL0!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2     ;ENABLE CONSOLE 0
003146' 164774                          CONSL1!BIT7!BIT6!BIT5!BIT4!BIT3!BIT2     ;ENABLE CONSOLE 1

                                ;DISPLAY OUTER REF. BOX WITH DIFFERENT LINE TYPES
003150' 150000                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003152' 114140                          POINT!LPON
003154' 000000                          0
003156' 001777                          MAXY
003160' 174200                          STATSB!LPLITE
003162' 113004                          LONGV!INT4!LINE0
003164' 041777                          INTX!MAXX                        ;TOP LINE
003166' 000000                          0
003170' 110005                          LONGV!LINE1                      ;RIGHT LINE
003172' 040000                          INTX
003174' 021777                          MINUSY!MAXY
003176' 110006                          LONGV!LINE2                      ;BOTTOM LINE
003200' 061777                          INTX!MINUSX!MAXX
003202' 000000                          0
003204' 110007                          LONGV!LINE3                      ;LEFT LINE
003206' 040000                          INTX
003210' 001777                          MAXY

003212' 150001                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003214' 114004                          POINT!LINE0
003216' 000400                          400
003220' 000200                          200
003222' 110000                          LONGV                            ;DRAW BASE REF. VECTOR
003224' 041200                          INTX+1200
003226' 000000                          0
003230' 114000                          POINT
003232' 000440                          440
003234' 000200                          200
003236' 174104                  GRPINC: STATSB!INCR+4
003240' 150002                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003242' 124000                          GRAPHY
003244' 163022                          DJSRR!XL                         ;DJSR RELATIVE TO THE TAG "SINE"
003246' 150003                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003250' 163020                          DJSRR!XL                         ;DJSR RELATIVE TO THE TAG "SINE"

003252' 114000                          POINT
003254' 000200                          200
003256' 000040                          40
003260' 150004                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003262' 110000                          LONGV
003264' 040000                          INTX
003266' 001200                          1200
003270' 114000                          POINT
003272' 000200                          200
003274' 000100                          100
003276' 150005                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003300' 120000                          GRAPHX
003302' 163003                          DJSRR!XL                         ;DJSR RELATIVE TO THE TAG "SINE"
003304' 150006                          DNAME!V0                         ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
003306' 163001                          DJSRR!XL                         ;DJSR RELATIVE TO THE TAG "SINE"
003310' 161115                          DJMPR!XL                         ;DJMP RELATIVE TO THE TAG "OCTPIC"
```

```

                                ;DATA STRING FOR A SINE WAVE

003312' 000200  000205  000212  SINE:   .WORD   0200,0205,0212,0217,0224,0231,0236,0243,0247,0253
003320' 000217  000224  000231
003326' 000236  000243  000247
003334' 000253
003336' 000257  000262  000265          .WORD   0257,0262,0265,0270,0272,0274,0276,0277,0277,0277
003344' 000270  000272  000274
003352' 000276  000277  000277
003360' 000277
003362' 000277  000276  000275          .WORD   0277,0276,0275,0274,0272,0267,0264,0261,0256,0252
003370' 000274  000272  000267
003376' 000264  000261  000256
003404' 000252
003406' 000246  000241  000235          .WORD   0246,0241,0235,0230,0223,0216,0211,0203,0176,0171
003414' 000230  000223  000216
003422' 000211  000203  000176
003430' 000171
003432' 000163  000156  000151          .WORD   0163,0156,0151,0144,0137,0133,0127,0123,0117,0114
003440' 000144  000137  000133
003446' 000127  000123  000117
003454' 000114
003456' 000111  000106  000104          .WORD   0111,0106,0104,0102,0101,0100,0100,0100,0100,0101
003464' 000102  000101  000100
003472' 000100  000100  000100
003500' 000101
003502' 000102  000104  000106          .WORD   0102,0104,0106,0111,0113,0117,0122,0126,0132,0137
003510' 000111  000113  000117
003516' 000122  000126  000132
003524' 000137
003526' 000144  000151  000156          .WORD   0144,0151,0156,0163,0170,0175
003534' 000163  000170  000175

003542' 166000                          DPOP                             ;DISPLAY POP AND RESTORE

```

```
  1086  003544'                 OCTPIC:
  1087  003544'  150007                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1088  003546'  114000                 POINT
  1089  003550'  001434                 1434
  1090  003552'  000724                 724
  1091  003554'  130030                 RELATP!BLKON
  1092  003556'  041600                 INTX+1600                       ;SHORT VECTOR OCTAGON
  1093  003560'  041607                 INTX+1600+7
  1094  003562'  040007                 INTX+7
  1095  003564'  061607                 INTX!MINUSX+1600+7
  1096  003566'  061600                 INTX!MINUSX+1600
  1097  003570'  061707                 INTX!MINUSX+1600+MINSUY+7
  1098  003572'  040107                 INTX+MINSUY+7
  1099  003574'  041707                 INTX+1600+MINSUY+7
  1100  003576'  150010                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1101  003600'  114000                 POINT
  1102  003602'  001430                 1430
  1103  003604'  000710                 710
  1104  003606'  130020                 RELATP!BLKOFF
  1105  003610'  043600                 INTX+3600                       ;SHORT VECTOR OCTAGON
  1106  003612'  043617                 INTX+3600+17
  1107  003614'  040017                 INTX+17
  1108  003616'  063617                 INTX!MINUSX+3600+17
  1109  003620'  063600                 INTX!MINUSX+3600
  1110  003622'  063717                 INTX!MINUSX+3600+MINSUY+17
  1111  003624'  040117                 INTX+MINSUY+17
  1112  003626'  043717                 INTX+3600+MINSUY+17
  1113  003630'  150011                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1114  003632'  114000                 POINT
  1115  003634'  001420                 1420
  1116  003636'  000660                 660
  1117  003640'  104030                 SHORTV!BLKON
  1118  003642'  047600                 INTX+7600                       ;SHORT VECTOR OCTAGON
  1119  003644'  047637                 INTX+7600+37
  1120  003646'  040037                 INTX+37
  1121  003650'  067637                 INTX!MINUSX+7600+37
  1122  003652'  067600                 INTX!MINUSX+7600
  1123  003654'  067737                 INTX!MINUSX+7600+MINSUY+37
  1124  003656'  040137                 INTX+MINSUY+37
  1125  003660'  047737                 INTX+7600+MINSUY+37
  1126  003662'  150012                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1127  003664'  114000                 POINT
  1128  003666'  001400                 1400
  1129  003670'  000600                 600
  1130  003672'  104020                 SHORTV!BLKOFF
  1131  003674'  057600                 INTX+17600                      ;SHORT VECTOR OCTAGON
  1132  003676'  057677                 INTX+17600+77
  1133  003700'  040077                 INTX+77
  1134  003702'  077677                 INTX!MINUSX+17600+77
  1135  003704'  077600                 INTX!MINUSX+17600
  1136  003706'  077777                 INTX!MINUSX+17600+MINSUY+77
  1137  003710'  040177                 INTX+MINSUY+77
  1138  003712'  057777                 INTX+17600+MINSUY+77
```

```
  1139  003714'  150013                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1140  003716'  114030                 POINT!BLKON
  1141  003720'  001360                 1360
  1142  003722'  000520                 520
  1143  003724'  110000                 LONGV                           ;LONG VECTOR OCTOGON BY LENGTH OF 137
  1144  003726'  040137                 INTX+137
  1145  003730'  000000                 0
  1146  003732'  040137                 INTX+137
  1147  003734'  000137                 137
  1148  003736'  040000                 INTX
  1149  003740'  000137                 137
  1150  003742'  060137                 INTX!MINUSX+137
  1151  003744'  000137                 137
  1152  003746'  060137                 INTX!MINUSX+137
  1153  003750'  000000                 0
  1154  003752'  060137                 INTX!MINUSX+137
  1155  003754'  020137                 MINUSX+137
  1156  003756'  040000                 INTX
  1157  003760'  020137                 MINUSX+137
  1158  003762'  040137                 INTX+137
  1159  003764'  020137                 MINUSX+137
  1160  003766'  150014                 DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
  1161  003770'  114120                 POINT!BLKOFF!LPOFF
  1162  003772'  001340                 1340
  1163  003774'  000440                 440
  1164  003776'  110000                 LONGV                           ;LONG VECTOR OCTOGON BY LENGTH OF 177
  1165  004000'  040177                 INTX+177
  1166  004002'  000000                 0
  1167  004004'  040177                 INTX+177
  1168  004006'  000177                 177
  1169  004010'  040000                 INTX
  1170  004012'  000177                 177
  1171  004014'  060177                 INTX!MINUSX+177
  1172  004016'  000177                 177
  1173  004020'  060177                 INTX!MINUSX+177
  1174  004022'  000000                 0
  1175  004024'  060177                 INTX!MINUSX+177
  1176  004026'  020177                 MINUSX+177
  1177  004030'  040000                 INTX
  1178  004032'  020177                 MINUSX+177
  1179  004034'  040177                 INTX+177
  1180  004036'  020177                 MINUSX+177
  1181
```

```
                                ;DISPLAY CHARACTER SET

004040' 150015                          DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004042' 154240                          CHARS1                          ;ENSURE NORMAL SIZE CHARS.
004044' 114140                          POINT!LPON
004046' 000100                          100
004050' 001677                          MAXY-100
004052' 170040                          STATSA!ITAL0
004054' 100000                          CHAR
004056' 163044                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT1"
004060' 170060                          STATSA!ITAL1
004062' 150016                          DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004064' 114000                          POINT
004066' 000100                          100
004070' 001647                          MAXY-130
004072' 100000                          CHAR
004074' 163035                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT1"
004076' 170040                          STATSA!ITAL0
004100' 150017                          DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004102' 114000                          POINT
004104' 000220                          220
004106' 001577                          MAXY-200
004110' 100000                          CHAR
004112' 163111                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT3"
004114' 170060                          STATSA!ITAL1
004116' 150020                          DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004120' 114000                          POINT
004122' 000220                          220
004124' 001547                          MAXY-230
004126' 100000                          CHAR
004130' 163102                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT3"
004132' 170040                          STATSA!ITAL0
004134' 150021                          DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004136' 114000                          POINT
004140' 000220                          220
004142' 001477                          MAXY-300
004144' 100000                          CHAR
004146' 163051                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT2"
004150' 170060                          STATSA!ITAL1
004152' 150022                          DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004154' 114000                          POINT
004156' 000220                          220
004160' 001447                          MAXY-330
004162' 100000                          CHAR
004164' 163042                          DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "PAT2"
004166' 161104                          DJMPR!XL                        ;DJMP RELATIVE TO THE TAG "TITLE0"
```

```
                                ;ASCII STRING FOLLOWED BY DPOP

004170' 040500  041502  042504  PAT1:   .ASCII  '@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_'
004176' 043506  044510  045512
004204' 046514  047516  050520
004212' 051522  052524  053526
004220' 054530  055532  056534
004226' 057536
004230' 020440  021442  022444          .ASCII  @ !"#$%&'()*+,-./0123456789:+<=>?@
004236' 023446  024450  025452
004244' 026454  027456  030460
004252' 031462  032464  033466
004260' 034470  025472  036474
004266' 037476
004270' 166000                          DPOP                    ;DISPLAY POP AND RESTORE

                                ;SHIFT-OUT ASCII STRING

004272'    016     000     001  PAT2:   .BYTE   16,0,1,2,3,4,5,6,7,10,11,12,13,14,15,16
004275'    002     003     004
004300'    005     006     007
004303'    010     011     012
004306'    013     014     015
004311'    016
004312'    020     021     022          .BYTE   20,21,22,23,24,25,26,27,30,31,32,33,34,35,36,37,17,0
004315'    023     024     025
004320'    026     027     030
004323'    031     032     033
004326'    034     035     036
004331'    037     017     000
004334' 166000                          DPOP                    ;DISPLAY POP AND RESTORE

                                ;LOWER CASE ASCII STRING

004336'    140     141     142  PAT3:   .BYTE   140,141,142,143,144,145,146,147
004341'    143     144     145
004344'    146     147
004346'    150     151     152          .BYTE   150,151,152,153,154,155,156,157
004351'    153     154     155
004354'    156     157
004356'    160     161     162          .BYTE   160,161,162,163,164,165,166,167
004361'    163     164     165
004364'    166     167
004366'    170     171     172          .BYTE   170,171,172,173,174,175,176,177
004371'    173     174     175
004374'    176     177
004376' 166000                          DPOP                    ;DISPLAY POP AND RESTORE

```

```
                                ;DISPLAY MODULE TITLE ON SCREEN

004400'                         TITLE0:
004400' 150023                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004402' 114000                          POINT
004404' 000600                          600
004406' 000320                          320
004410' 100000                          CHAR
004412' 042504  027503  030530          .ASCIZ  "DEC/X11 PDP-11 SYSTEM EXERCISER"
004420' 020061  042120  026520
004426' 030461  051440  051531
004434' 042524  020115  054105
004442' 051105  044503  042523
004450' 000122
004452' 114000                          POINT
004454' 000400                          400
004456' 000020                          20
004460' 100000                          CHAR
004462' 042504  043503  040522          .ASCIZ  "DECGRAPHIC-11 VS-60 ALPHAGRAPHIC DISPLAY SYSTEM"
004470' 044120  041511  030455
004476' 020061  051526  033055
004504' 020060  046101  044120
004512' 043501  040522  044120
004520' 041511  042040  051511
004526' 046120  054501  051440
004534' 051531  042524  000115

                                ;DISPLAY INTENSITY LEVELS

004542' 150024                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004544' 170040                          STATSA!ITAL0
004546' 114000                          POINT
004550' 000340                          340
004552' 001300                          1300
004554' 113604                          LONGV!INT7!LINE0
004556' 040400                          INTX+400
004560' 000000                          0
004562' 150025                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004564' 114000                          POINT
004566' 000340                          340
004570' 001240                          1240
004572' 113400                          LONGV!INT6
004574' 040400                          INTX+400
004576' 000000                          0
004600' 150026                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004602' 114000                          POINT
004604' 000340                          340
004606' 001200                          1200
004610' 113200                          LONGV!INT5
004612' 040400                          INTX+400
004614' 000000                          0
004616' 150027                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004620' 114000                          POINT
004622' 000340                          340
004624' 001140                          1140
004626' 113000                          LONGV!INT4
```

```
004630' 040400                          INTX+400
004632' 000000                          0
004634' 150030                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004636' 114000                          POINT
004640' 000340                          340
004642' 001100                          1100
004644' 112600                          LONGV!INT3
004646' 040400                          INTX+400
004650' 000000                          0
004652' 150031                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004654' 114000                          POINT
004656' 000340                          340
004660' 001040                          1040
004662' 112400                          LONGV!INT2
004664' 040400                          INTX+400
004666' 000000                          0
004670' 150032                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004672' 114000                          POINT
004674' 000340                          340
004676' 001000                          1000
004700' 112200                          LONGV!INT1
004702' 040400                          INTX+400
004704' 000000                          0
004706' 150033                          DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
004710' 114000                          POINT
004712' 000340                          340
004714' 000740                          740
004716' 112000                          LONGV!INT0
004720' 040400                          INTX+400
004722' 000000                          0


                                ;ASSOCITIVE NAME MATCH INTERRUPT SECTION

004724' 150000                  NMATCH: DNAME!0C                ;VARIABLE VALUE FOR THE DISPLAY
                                                                ;ASSOCITIVE NAME -- UPON EXECUTION
                                                                ; OF THIS INSTRUCTION A NAME MATCH
                                                                ; INTERRUPT SHOULD OCCUR
```

```

                                        ;LIGHT-PEN SWITCH SECTION
004726' 150034                                  DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
004730' 164600                                  CONSL1!BIT7                     ;DISABLE CONSOLE #1
004732' 117000                                  POINT!INT4
004734' 000340                                  340
004736' 000500                                  500
004740' 100000                                  CHAR
004742' 042520  020116  053523                  .ASCII  /PEN SWITCH IS /
004750' 052111  044103  044440
004756' 020123
004760' 050125  047440  020116  PENSW0: .ASCII  /UP ON CONSOLE 0 /      ;CHANGES TO "DN" UPON SWITCH DOWN
004766' 047503  051516  046117
004774' 020105  020060
005000' 164700                                  CONSL1!BIT7!BIT6                ;ENABLE CONSOLE #1
005002' 164200                                  CONSL0!BIT7                     ;DISABLE CONSOLE #0
005004' 150035                                  DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005006' 114000                                  POINT
005010' 000340                                  340
005012' 000500                                  500
005014' 100000                                  CHAR
005016' 042520  020116  053523                  .ASCII  /PEN SWITCH IS /
005024' 052111  044103  044440
005032' 020123
005034' 050125  047440  020116  PENSW1: .ASCII  /UP ON CONSOLE 1 /      ;CHANGE TO "DN" UPON SWITCH DOWN
005042' 047503  051516  046117
005050' 020105  020061
005054' 164300                                  CONSL0!BIT7!BIT6                ;ENABLE CONSOLE #0

```

```


                                        ;EDGE SCISSORING AT THE TOP OF FRAME 0

005056'                                 EDGESC:
005056' 150036                                  DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005060' 114000                                  POINT
005062' 000000                                  0
005064' 001677                                  MAXY-100
005066' 110000                                  LONGV
005070' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005072' 000200                                  200                             ;VECTOR OFF SCREEN TO ON SCREEN
005074' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005076' 020200                                  MINUSY!200                      ;VECTOR OFF SCREEN TO ON SCREEN
005100' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005102' 000200                                  200                             ;VECTOR OFF SCREEN TO ON SCREEN
005104' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005106' 020200                                  MINUSY!200                      ;VECTOR OFF SCREEN TO ON SCREEN
005110' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005112' 000200                                  200                             ;VECTOR OFF SCREEN TO ON SCREEN
005114' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005116' 020200                                  MINUSY!200                      ;VECTOR OFF SCREEN TO ON SCREEN
005120' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005122' 000200                                  200                             ;VECTOR OFF SCREEN TO ON SCREEN
005124' 040200                                  INTX!200                        ;VECTOR ON SCREEN TO OFF SCREEN
005126' 020200                                  MINUSY!200                      ;VECTOR OFF SCREEN TO ON SCREEN

                                        ;SUPER/SUB-SCRIPT CHARACTER SECTION

005130' 150037                                  DNAME!V0                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005132' 114000                                  POINT
005134' 001400                                  1400
005136' 001400                                  1400
005140' 154240                                  CHARS1                          ;NORMAL CHAR. SIZE
005142' 100000                                  CHAR
005144'    102     021     062                  .BYTE   102,SUPON,62,65,SUPOFF
005147'    065     023
005151'    102     022     062                  .BYTE   102,SUBON,62,65,SUBOFF
005154'    065     024
005156'    103     021     123                  .BYTE   103,SUPON,123,124,SUPOFF
005161'    124     023
005163'    103     022     123                  .BYTE   103,SUBON,123,124,SUBOFF
005166'    124     024

```

```
                                          ;FOUR SCALED ROTATED LETTERS

005170' 150040                                    DNAME!V0                    ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005172' 114000                                    POINT
005174' 000100                                    100
005176' 000000                                    0
005200' 170040                                    STATSA!ITAL0                ;NON-ITALIC
005202' 155400                                    CHRRT1                      ;ROTATE CHARACTERS
005204' 163024                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "SCLDCH"

                                          ;FOUR SCALED ROTATED ITALIZIED LETTERS
005206' 150041                                    DNAME!V0                    ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005210' 114000                                    POINT
005212' 000040                                    40
005214' 000000                                    0
005216' 170060                                    STATSA!ITAL1                ;ENABLE ITALIC
005220' 163016                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "SCLDCH"

                                          ;FOUR SCALED ITALIZIED LETTERS
005222' 150042                                    DNAME!V0                    ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005224' 114000                                    POINT
005226' 001600                                    1600
005230' 001600                                    1600
005232' 170060                                    STATSA!ITAL1
005234' 155000                                    CHRRT0                      ;NON-ROTATE,NON-ITALIC
005236' 163007                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "SCLDCH"

                                          ;FOUR SCALED LETTERS
005240' 150043                                    DNAME!V0                    ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005242' 114000                                    POINT
005244' 001600                                    1600
005246' 001540                                    1540
005250' 170040                                    STATSA!ITAL0
005252' 163001                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "SCLDCH"
005254' 161016                                    DJMPR!XL                    ;DJMP RELATIVE TO THE TAG "FRAME1"

005256' 154200                            SCLDCH: CHARS0                      ;CHAR SCALE 0
005260' 100000                                    CHAR
005262'    102        000                         .BYTE   102,0
005264' 154240                                    CHARS1                      ;CHAR SCALE 1
005266' 100000                                    CHAR
005270'    102        000                         .BYTE   102,0
005272' 154300                                    CHARS2                      ;CHAR SCALE 2
005274' 100000                                    CHAR
005276'    102        000                         .BYTE   102,0
005300' 154340                                    CHARS3                      ;CHAR SCALE 3
005302' 100000                                    CHAR
005304'    102        000                         .BYTE   102,0
005306' 154240                                    CHARS1
005310' 165000                                    DPOPNR                      ;DISPLAY POP AND NO RESTORE

```

```
                                          ;FRAME +X 1, +Y 1
                                          ;USE DJSRR TO DRAW SCALED BOXES FROM +2000X, +2000Y

005312'                                   FRAME1:
005312'                                   PX1PY1:
005312' 150044                                    DNAME!V0                    ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0
005314' 154024                                    VCTR00!4
005316' 114000                                    POINT
005320' 002000                                    BIT10
005322' 002000                                    BIT10
005324' 154037                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005326' 163040                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005330' 154036                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005332' 163036                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005334' 154035                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005336' 163034                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005340' 154034                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005342' 163032                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005344' 154033                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005346' 163030                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005350' 154032                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005352' 163026                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005354' 154031                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005356' 163024                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005360' 154030                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005362' 163022                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005364' 154027                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005366' 163020                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005370' 154026                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005372' 163016                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005374' 154025                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005376' 163014                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005400' 154024                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005402' 163012                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005404' 154023                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005406' 163010                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005410' 154022                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005412' 163006                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005414' 154021                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005416' 163004                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005420' 154020                                    VCTR00!A0                   ;CHANGE VECTOR SCALE
005422' 163002                                    DJSRR!XL                    ;DJSR RELATIVE TO THE TAG "DRWBOX"
005424' 154024                                    VCTR00!4                    ;RESET VECTOR SCALE
005426' 161016                                    DJMPR!XL                    ;DJMP RELATIVE TO THE TAG "FRAME2"

```

```
005430' 110000                          DRWBOX: LONGV
005432' 040500                                  INTX!500                        ;DRAW A BOX
005434' 000000                                  0
005436' 040000                                  INTX
005440' 000500                                  500
005442' 060500                                  INTX!MINUSX!500
005444' 000000                                  0
005446' 040000                                  INTX
005450' 020500                                  MINUSY!500
005452' 000000                                  0
005454' 000000                                  0
005456' 164000                                  DNOP
005460' 164000                                  DNOP
005462' 166000                                  DPOP                            ;DISPLAY POP AND RESTORE

                                        ;FRAME -X1, -Y1
                                        ;USE "BASIC VECTOR" TO DRAW AN "STAR"

005464'                                 FRAME2:
005464'                                 NX1NY1:
005464' 150045                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005466' 114000                                  POINT
005470' 020777                                  MINUSX!HALFX
005472' 020777                                  MINUSY!HALFX
005474' 120000                                  BASICV
005476' 042777                                  INTX!PATH0!HALFX
005500' 062777                                  INTX!PATH4!HALFX
005502' 046777                                  INTX!PATH1!HALFX
005504' 066777                                  INTX!PATH5!HALFX
005506' 052777                                  INTX!PATH2!HALFX
005510' 072777                                  INTX!PATH6!HALFX
005512' 056777                                  INTX!PATH3!HALFX
005514' 076777                                  INTX!PATH7!HALFX
005516' 062777                                  INTX!PATH4!HALFX
005520' 042777                                  INTX!PATH0!HALFX
005522' 066777                                  INTX!PATH5!HALFX
005524' 046777                                  INTX!PATH1!HALFX
005526' 072777                                  INTX!PATH6!HALFX
005530' 052777                                  INTX!PATH2!HALFX
005532' 076777                                  INTX!PATH7!HALFX
005534' 056777                                  INTX!PATH3!HALFX
```

```
                                        ;FRAME  +X 2, +Y 2
                                        ;TEST ALL STACK LEVELS WORK PROPERLY
                                        ;       DJSR DOWN 8 LEVELS AND DPOP BACK UP

005536'                                 FRAME3:
005536' 150046                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005540' 114000                                  POINT
005542' 004000                                  BIT11
005544' 005000                                  BIT11!1000

005546' 163001                                  DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "LEVEL0"

005550' 161077                                  DJMPR!XL                        ;DJMP RELATIVE TO THE TAG "FILE0A"

005552'                                 LEVEL0:
005552' 150047                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005554' 100000                                  CHAR
005556' 042514  042526  020114                  .ASCII  /LEVEL 0/<12>
005564' 005060
005566' 163001                                  DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "LEVEL1"
005570' 166000                                  DPOP

005572'                                 LEVEL1:
005572' 150050                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005574' 100000                                  CHAR
005576' 042514  042526  020114                  .ASCII  /LEVEL 1/<12>
005604' 005061
005606' 163001                                  DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "LEVEL2"
005610' 166000                                  DPOP

005612'                                 LEVEL2:
005612' 150051                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005614' 100000                                  CHAR
005616' 042514  042526  020114                  .ASCII  /LEVEL 2/<12>
005624' 005062
005626' 163001                                  DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "LEVEL3"
005630' 166000                                  DPOP

005632'                                 LEVEL3:
005632' 150052                                  DNAME!VO                        ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005634' 100000                                  CHAR
005636' 042514  042526  020114                  .ASCII  /LEVEL 3/<12>
005644' 005063
005646' 163001                                  DJSRR!XL                        ;DJSR RELATIVE TO THE TAG "LEVEL4"
005650' 166000                                  DPOP
```

```
005652'                             LEVEL4:
005652' 150053                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005654' 100000                              CHAR
005656' 042514  042526  020114              .ASCII  /LEVEL 4/<12>
005664' 005064
005666' 163001                              DJSRR!XL                ;DJSR RELATIVE TO THE TAG "LEVEL5"
005670' 166000                              DPOP

005672'                             LEVEL5:
005672' 150054                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005674' 100000                              CHAR
005676' 042514  042526  020114              .ASCII  /LEVEL 5/<12>
005704' 005065
005706' 163001                              DJSRR!XL                ;DJSR RELATIVE TO THE TAG "LEVEL6"
005710' 166000                              DPOP

005712'                             LEVEL6:
005712' 150055                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005714' 100000                              CHAR
005716' 042514  042526  020114              .ASCII  /LEVEL 6/<12>
005724' 005066
005726' 163001                              DJSRR!XL                ;DJSR RELATIVE TO THE TAG "LEVEL7"
005730' 166000                              DPOP

005732'                             LEVEL7:
005732' 150056                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005734' 100000                              CHAR
005736' 042514  042526  020114              .ASCII  /LEVEL 7/<12>
005744' 005067
005746' 166000                              DPOP
```

```
005750'                             FILE0A:
005750' 161016                              DJMPR!XL                ;DJMP RELATIVE TO THE TAG "FILE0B"
005752' 150057                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
005754' 117000                              POINT!INT4
005756' 000340                              340
005760' 000440                              440
005762' 164600                              CONSL1!BIT7             ;DISABLE CONSOLE #1
005764' 100000                              CHAR
005766' 044514  044107  026524              .ASCIZ  /LIGHT-PEN 0 HIT/
005774' 042520  020116  020060
006002' 044510  000124
006006'                             FILE0B:
006006' 161020                              DJMPR!XL                ;DJMP RELATIVE TO THE TAG "FILE0C"
006010' 150060                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
006012' 164200                              CONSL0!BIT7             ;DISABLE CONSOLE #0
006014' 164700                              CONSL1!BIT7!BIT6        ;ENABLE CONSOLE #1
006016' 114000                              POINT
006020' 000340                              340
006022' 000420                              420
006024' 100000                              CHAR
006026' 044514  044107  026524              .ASCIZ  /LIGHT-PEN 1 HIT/
006034' 042520  020116  020061
006042' 044510  000124

006046' 164300                              CONSL0!BIT7!BIT6        ;ENABLE CONSOLE #0

                                    ;DISPLAY THE REF. MENU BOX.

006050' 170003                      FILE0C: DMENU1                  ;ENABLE MENU
006052' 150061                              DNAME!VO                ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
006054' 114000                              POINT                   ;CLEAR OFFSET REGISTERS
006056' 010000                              OFFST0
006060' 010000                              OFFST0
006062' 114000                              POINT
006064' 000000                              0
006066' 000000                              0
006070' 110000                              LONGV
006072' 040177                              INTX!MAXMUX             ;DRAW  MAX X MENU VECTOR
006074' 000000                              0
006076' 040000                              INTX
006100' 001777                              MAXY                    ;DRAW VERT LINE
006102' 060177                              INTX!MINUSX!MAXMUX      ;DRAW - MENU VECTOR
006104' 000000                              0
006106' 040000                              INTX                    ;DRAW - VERT LINE
006110' 021777                              MINUSX!MAXY
006112' 040177                              INTX!MAXMUX             ;DRAW DIAG. LING
006114' 001777                              MAXY
006116' 000000                              0
006120' 021777                              MINUSX!MAXY             ;LOCATE TO BOTTOM RIGHT CORNOR
006122' 060177                              INTX!MINUSX!MAXMUX      ;DRAW - DIAG. LINE
006124' 001777                              MAXY
006126' 170002                              DMENU0                  ;DISABLE MENU
```

```
 1705  006130' 164000               FILEQ:  DNOP
 1706  006132' 164000                       DNOP
 1707  006134' 164000                       DNOP
 1708
 1709                               ;NOW GENERATE A DISPLAY STOP INTERRUPT
 1710                               ;       AND THEN CONTINUE DRAWING THE PICTURE
 1711
 1712  006136' 173400               FILEOE: DSTOP
 1713  006140' 160000                       DJMP
 1714  006142' 003150'              FILEOD: FRAME0+4
 1715
 1716  006144' 047104               DWNMSG: .ASCII  /DN/
 1717  006146' 050125               UPMSG:  .ASCII  /UP/
 1718
 1719                               ;PATCH AREA
 1720  006150' 000240                       NOP
 1721  006152' 000240                       NOP
 1722  006154' 000240                       NOP
 1723  006156' 000240                       NOP
 1724  006160' 000240                       NOP
 1725  006162' 000240                       NOP
 1726  006164' 000240                       NOP
 1727  006166' 000240                       NOP
 1728  006170' 000240                       NOP
 1729  006172' 000240                       NOP
 1730  006174' 000240                       NOP
 1731  006176' 000240                       NOP
 1732  006200' 000240                       NOP
 1733  006202' 000240                       NOP
 1734  006204' 000240                       NOP
 1735  006206' 000240                       NOP
 1736  006210' 000240                       NOP
 1737  006212' 000240                       NOP
 1738  006214' 000240                       NOP
 1739  006216' 000240                       NOP
 1740  006220' 000240                       NOP
 1741  006222' 000240                       NOP
 1742  006224' 000240                       NOP
 1743  006226' 000240                       NOP
 1744
 1745  006230' 000240               THEEND: NOP
 1746
 1747          000001                       .END
```

```
ABORT    003136R      625*    639*    660*    681*    702*    714     991#
ABSVCT=  144000       397#
ACSR     000102R      535#    631*    632*    633     652*    653*    654     673*    674*    675     695*    696     775*
                      830*    831     859*    863     871*    873     877     883     888     892     898     920*    921*
                      922     930*    931*    952*    961
ADDR     000006R      501#    606
ADDR22=  001000       553#
ANAME    003112R      759*    952     968*    969     971*    972     975     980#
ASB      000106R      539#
ASTAT    000104R      537#    628*    630     633     643*    649*    651     654     664*    670*    672     675     685*
                      692*    694     696     706*    773*    860*    919*    951*    954     960*    961
AWAS     000110R      540#
A0     = 177777      1500#   1502#   1504#   1506#   1508#   1510#   1512#   1514#   1516#   1518#   1520#   1522#   1524#
                     1526#   1528#   1530#   1532#
BASICS=  134000       396#
BASICV=  120000       394#   1559
BEGIN    000000R      498#    616     637     641     642     658     662     663     679     683     684     700     704
                      705     716     765     771     784     807     825     835     840     847     850     857     867
                      908     917     926     935     940     949     958     965
BEGNA    001226R      715     718#
BIT0   = 000001       446     471     553#    590     639     720
BIT1   = 000002       445     470     553#    660     692
BIT10  = 002000       434     438     468     553#    759     760     762     971     979     980    1498    1499
BIT11  = 004000       553#    628     649     670     762     973     979    1583    1584
BIT12  = 010000       553#    691     762     883     973     979
BIT13  = 020000       553#    793     877
BIT14  = 040000       479     553#    762     873     973     979
BIT15  = 100000       553#
BIT2   = 000004       553#    681     863     995     996
BIT3   = 000010       553#    702     995     996
BIT4   = 000020       466     474     553#    995     996
BIT5   = 000040       462     464     473     553#    752     969     995     996
BIT6   = 000100       463     464     553#    898     922     995     996    1383    1396    1668    1676
BIT7   = 000200       461     553#    892     995     996    1372    1383    1384    1396    1659    1667    1668    1676
BIT8   = 000400       441     459     553#    888
BIT9   = 001000       433     435     439     458     553#
BLKOFF=  000020       415#   1104    1130    1161
BLKON  = 000030       416#   1091    1117    1140
BREAK$=  104407       553#    641     642     662     663     683     684     704     705
BR1      000012R      503#    728     734     736     738     740
BR2      000013R      504#
BTOD$  = 104421       553#
CDATA$=  104412       553#
CHAR   = 100000       388#   1190    1197    1204    1211    1218    1225    1283    1293    1376    1389    1432    1476
                     1479    1482    1485    1592    1600    1608    1616    1624    1632    1640    1648    1660    1672
CHARS0=  154200       461#    462     463     464    1475
CHARS1=  154240       462#   1185    1431    1478    1487
CHARS2=  154300       463#   1481
CHARS3=  154340       464#   1484
CHRRT0=  155000       458#    459    1464
CHRRT1=  155400       459#   1449
CONFIG   000056R      523#
CONSL0=  164000       440#    995    1384    1396    1667    1676
CONSL1=  164400       441#    996    1372    1383    1659    1668
CONT     002100R      824#    852     905     978
CSRA     000100R      533#    629*    650*    671*    693*    774*    829*    861*    872*    929*    953*
```

```
DATCK$= 104411         553#
DATER$= 104404         553#
DELAY   000232R        559#   741    792
DELAY1  000234R        560#   742    815
DJMP   = 160000        432#   433    434   1713
DJMPR  = 161000        433#  1049   1227   1473   1533   1588   1654   1665
DJSR   = 162000        434#   435
DJSRR  = 163000        435#  1030   1032   1046   1048   1191   1198   1205   1212   1219   1226   1450   1457
                      1465   1472   1501   1503   1505   1507   1509   1511   1513   1515   1517   1519   1521
                      1523   1525   1527   1529   1531   1586   1595   1603   1611   1619   1627   1635   1643
DLTXRG  003140R        755*   757    793    795*   797    799*   801*   803    805*   822    992#
DLTYRG  003142R        756*   758    796*   800*   802*   806*   823    993#
DMENU0= 170002         445#   446   1703
DMENU1= 170003         446#  1680
DNAME  = 150000        475#   760    976    999   1017   1028   1031   1037   1044   1047   1087   1100   1113
                      1126   1139   1160   1184   1193   1200   1207   1214   1221   1279   1305   1313   1320
                      1327   1334   1341   1348   1355   1366   1371   1385   1403   1427   1444   1452   1459
                      1467   1495   1555   1581   1591   1599   1607   1615   1623   1631   1639   1647   1655
                      1666   1681
DNOP   = 164000        437#   438    439    440    441    724    875    890   1546   1547   1705   1706   1707
DONTST  001214R        714#
DPCTST  001076R        691#
DPOP   = 166000        438#  1084   1242   1258   1274   1548   1596   1604   1612   1620   1628   1636   1644
                      1651
DPOPNR= 165000         439#  1488
DRWBOX  005430R       1501   1503   1505   1507   1509   1511   1513   1515   1517   1519   1521   1523   1525
                      1527   1529   1531   1535#
DSTOP  = 173400        444#  1712
DVID1   000014R        505#
DWNMSG  006144R        879    894   1716#
EABITS  003116R        622*   763    983#
EDGESC  005056R       1402#
EDGE0  = 176040        473#   474
EDGE1  = 176060        474#
ENDIT$= 104413         553#   807    850
END$   = 104410        553#   716
ERRTYP  000106R        538#   635*   656*   677*   698*   782*   833*   838*   865*   906*   924*   933*   938*
                       956*   963*
EXIT$  = 104400        553#   765    825
EXSTOP  002112R        776    829#
FAST    003122R        718    722*   985#
FILEQ   006130R       1705#
FILE0A  005750R        746    749*   820*   875*  1588   1653#
FILE0B  006006R        747    750*   821*   890*  1654   1664#
FILE0C  006050R       1665   1680#
FILE0D  006142R        623*   624*  1714#
FILE0E  006136R        725   1712#
FIRST   003124R        744    748*   986#
FRAME)  003144R        555    723    995#  1714
FRAME1  005312R       1473   1493#
FRAME2  005464R       1533   1553#
FRAME3  005536R       1580#
GETPA$= 104415         553#   616
GRAPHX= 120000         392#   394   1045
GRAPHY= 124000         393#  1029
GRPINC  003236R        751*   816*   817    819*  1027#
```

```
GTASNA  000256R        569#   762*   974*
GTCONS  000260R        570#   871    872
GTDLY0  003126R        741*   790*   792*   987#
GTDLY1  003130R        742*   913*   815*   988#
GTDNE1  000300R        578#   728*   734*
GTDONE  000276R        577#   609    727*   733*
GTLPEN  002210R        735    855#
GTLPH   000302R        579#   735*
GTLPH1  000304R        580#   736*
GTMACH  002710R        739    947#
GTNAME  000262R        571#   773    860    919    951    953
GTNAMM  000312R        583#   739*
GTNAM1  000314R        584#   740*
GTPC    000236R        561#   605    693    694*   695    708*   764*   824*
GTREL   000246R        565#   671    672*   673    687*   763*
GTSHIF  002544R        737    915#
GTSOTM  000306R        581#   737*
GTSOT1  000310R        582#   738*
GTSPPT  000270R        574#   691*   709*   752*
GTSR    000240R        562#   774    775    859    861    920    921
GTSR1   000250R        566#   829    830    929    930
GTSTAK  000264R        572#
GTSTOP  001572R        733    769#
GTTERM  000266R        573#
GTXOFF  000252R        567#   629    630*   631    645*   757*   822*
GTXPOS  000242R        563#
GTYOFF  000254R        568#   650    651*   652    666*   758*   823*
GTYPOS  000244R        564#
GTZOFF  000274R        576#
GTZPOS  000272R        575#
GWBUF$= 104414         553#
HALFX  = 000777        483#  1557   1558   1560   1561   1562   1563   1564   1565   1566   1567   1568   1569
                      1570   1571   1572   1573   1574   1575
HRDCNT  000044R        518#
HRDER$= 104405         553#   637    658    679    700    784    835    840    867    908    926    935    940
                       958    965
HRDPAS  000050R        520#
ICONT   000036R        515#   594    596    598*   602    604*
ICOUNT  000040R        516#
IDNUM   000122R        545#
INCR   = 000100        455#   751    817    819   1027
INIT    000030R        512#
INTR    000120R        544#   593*   601*
INTX   = 040000        479#  1005   1008   1011   1014   1022   1039   1092   1093   1094   1095   1096   1097
                      1098   1099   1105   1106   1107   1108   1109   1110   1111   1112   1118   1119   1120
                      1121   1122   1123   1124   1125   1131   1132   1133   1134   1135   1136   1137   1138
                      1144   1146   1148   1150   1152   1154   1156   1158   1165   1167   1169   1171   1173
                      1175   1177   1179   1311   1318   1325   1332   1339   1346   1353   1360   1408   1410
                      1412   1414   1416   1418   1420   1422   1536   1538   1540   1542   1560   1561   1562
                      1563   1564   1565   1566   1567   1568   1569   1570   1571   1572   1573   1574   1575
                      1689   1691   1693   1695   1697   1701
INT0   = 002000        404#  1359
INT1   = 002200        405#  1352
INT2   = 002400        406#  1345
INT3   = 002600        407#  1338
INT4   = 003000        408#  1004   1331   1373   1656
```

```
INT5   = 003200              409#    1324
INT6   = 003400              410#    1317
INT7   = 003600              411#    1310
ITAL0  = 000040              450#    1189    1199    1213    1306    1448    1471
ITAL1  = 000060              451#    1192    1206    1220    1456    1463
JMPF0B   003132R             746*     749     820     989#
JMPF0C   003134R             747*     750     821     990#
LEVEL0   005552R            1586     1590#
LEVEL1   005572R            1595     1598#
LEVEL2   005612R            1603     1606#
LEVEL3   005632R            1611     1614#
LEVEL4   005652R            1619     1622#
LEVEL5   005672R            1627     1630#
LEVEL6   005712R            1635     1638#
LEVEL7   005732R            1643     1646#
LINE0  = 000004              418#    1004    1018    1310
LINE1  = 000005              419#    1007
LINE2  = 000006              420#    1010
LINE3  = 000007              421#    1013
LONGV  = 110000              390#    1004    1007    1010    1013    1021    1038    1143    1164    1310    1317    1324    1331
                            1338     1345    1352    1359    1407    1535    1688
LPDARK= 000300               449#
LPLITE= 000200               448#    1003
LPOFF  = 000100              413#    1161
LPON   = 000140              414#    1000    1186
MAP22$= 104416               553#
MAXMUX= 000177               480#    1689    1693    1697    1701
MAXX   = 001777              481#     483    1005    1011
MAXY   = 001777              482#    1002    1009    1015    1188    1196    1203    1210    1217    1224    1406    1692    1696
                            1698     1700    1702
MINSUY= 000100               486#    1097    1098    1099    1110    1111    1112    1123    1124    1125    1136    1137    1138
MINUSX= 020000               484#     799     800     803    1009    1011    1095    1096    1097    1108    1109    1110    1121
                            1122     1123    1134    1135    1136    1150    1152    1154    1155    1157    1159    1171    1173
                            1175     1176    1178    1180    1540    1557    1693    1696    1700    1701
MINUSY= 020000               485#    1411    1415    1419    1423    1543    1558
MODNAM   000000R             499#
MODSP    000224R             513      551#
MOTION   003120R             743*     788     880*    886*    895*    901*    984#
MSGN$  = 104403              553#
MSGS$  = 104402              553#
MSG$   = 104401              553#
NAMESW   003114R             780      787*    977*    981#
NMATCH   004724R             760*     975*    976*   1366#
NOPDON   002174R             727      845#
NULL   = 000000              553#     637     658     679     700     784     835     840     867     908     926     935     940
                             958      965
NX1NY1   005464R            1554#
OCTPIC   003544R            1049     1086#
OFFST0= 010000               399#    1683    1684
OFFST1= 012000               400#
OFFST2= 014000               401#
OFFST3= 016000               402#
OPEN   = 000000              500      506     507     508     509     526     527     528     529     530     531     532     533
                             535      537     539     540     542     543     544     553#    556     557
OTOA$  = 104420              553#
PASCNT   000034R             514#     588
```

```
PATH0  = 002000              423#    1560    1569
PATH1  = 006000              424#    1562    1571
PATH2  = 012000              425#    1564    1573
PATH3  = 016000              426#    1566    1575
PATH4  = 022000              427#    1561    1568
PATH5  = 026000              428#    1563    1570
PATH6  = 032000              429#    1565    1572
PATH7  = 036000              430#    1567    1574
PAT1     004170R            1191     1198    1230#
PAT2     004272R            1219     1226    1246#
PAT3     004336R            1205     1212    1262#
PENSW0   004760R             753*     879*    885*   1380#
PENSW1   005034R             754*     894*    900*   1393#
PIRQ$  = 000004              553#     771     847     857     917     949
POINT  = 114000              391#    1000    1018    1024    1034    1041    1088    1101    1114    1127    1140    1161    1186
                            1194     1201    1208    1215    1222    1280    1290    1307    1314    1321    1328    1335    1342
                            1349     1356    1373    1386    1404    1428    1445    1453    1460    1468    1497    1556    1582
                            1656     1669    1682    1685
POPSP  = 005726              553#
POPSP2= 022626               553#
PRTY   = 000000              504      553#
PRTY0  = 000000              553#
PRTY1  = 000040              553#
PRTY2  = 000100              553#
PRTY3  = 000140              553#
PRTY4  = 000200              503      553#
PRTY5  = 000240              553#
PRTY6  = 000300              553#
PRTY7  = 000340              553#
PS     = 177776              553#
PSW    = 177776              553#
PUSH   = 005746              553#
PUSH2  = 024646              553#
PX1PY1   005312R            1494#
RANDS$ = 104407              553#
RANNUM   000054R             522#
RBUFEA   000230R             557#     617
RBUFPA   000226R             556#     623     764
RBUFVA   000224R             555#     616
RELATP= 130000               395#     396    1091    1104
RELTST   000764R             670#
RESTR    002050R             789      791     809     814     818     820#
RESTRT   000430R             541      589     597     599     603     605#
RES1     000056R             524#
RES2     000060R             525#
RSTRT    000112R             541#
SBADR    000102R             534#
SCLDCH   005256R            1450     1457    1465    1472    1475#
SHORTV= 104000               389#    1117    1130
SINE     003312R            1030     1032    1046    1048    1053#
SOFCNT   000042R             517#
SOFER$= 104406               553#
SOFPAS   000046R             519#
SPOINT   000032R             513#
SPSIZ  = 000040                1#     546
SR1      000016R             506#     590     720
```

```
SR2      000020R             507#
SR3      000022R             508#
SR4      000024R             509#
START    000316R             512     588#     614     837     842     869     910     928     937     942     967
STAT     000026R             511#
STATE  = 176000              468#    470      473
STATSA = 170000              443#    445     1189    1192    1199    1206    1213    1220    1306    1448    1456    1463    1471
STATSB = 174000              453#    468      751     817     819    1003    1027
STATSC = 154000              457#    458      461     466
STOPRT   001600R             771     773#
STRNG0 = 176002              470#    471
STRNG1 = 176003              471#
SUBOFF = 000024              491#   1435     1439
SUBON  = 000022              490#   1435     1439
SUPOFF = 000023              489#   1433     1437
SUPON  = 000021              488#   1433     1437
SVR0     000062R             526#
SVR1     000064R             527#
SVR2     000066R             528#
SVR3     000070R             529#
SVR4     000072R             530#
SVR5     000074R             531#
SVR6     000076R             532#
SYSCNT   000052R             521#
THEEND   006230R            1745#
TITLED   004400R            1227    1278#
TRPDFD = 000022              553#
UPMSG    006146R             753     754      885     900    1717#
VCTR00 = 154020              466#   1496     1500    1502    1504    1506    1508    1510    1512    1514    1516    1518    1520
                            1522    1524     1526    1528    1530    1532
VECTOR   000010R             502#    611
V0     = 000062              492#    999     1000#   1017    1018#   1028    1029#   1031    1032#   1037    1038#   1044    1045#
                            1047    1048#    1087    1088#   1100    1101#   1113    1114#   1126    1127#   1139    1140#   1160
                            1161#   1184     1185#   1193    1194#   1200    1201#   1207    1208#   1214    1215#   1221    1222#
                            1279    1280#    1305    1306#   1313    1314#   1320    1321#   1327    1328#   1334    1335#   1341
                            1342#   1348     1349#   1355    1356#   1371    1372#   1385    1386#   1403    1404#   1427    1428#
                            1444    1445#    1452    1453#   1459    1460#   1467    1468#   1495    1496#   1555    1556#   1581
                            1582#   1591     1592#   1599    1600#   1607    1608#   1615    1616#   1623    1624#   1631    1632#
                            1639    1640#    1647    1648#   1655    1656#   1666    1667#   1681    1682#
WASADR   000104R             536#
WDFR     000116R             543#    592*     600*
WDTO     000114R             542#
XDOFF    000540R             628#
XFLAG    000005R             500#
XL     = 000020             1030#   1032#    1046#   1048#   1049#   1191#   1198#   1205#   1212#   1219#   1226#   1227#   1450#
                            1457#   1465#    1472#   1473#   1501#   1503#   1505#   1507#   1509#   1511#   1513#   1515#   1517#
                            1519#   1521#    1523#   1525#   1527#   1529#   1531#   1533#   1586#   1588#   1595#   1603#   1611#
                            1619#   1627#    1635#   1643#   1654#   1665#
YDOFF    000652R             649#
.      = 006232R            1030    1032     1046    1048    1049    1191    1198    1205    1212    1219    1226    1227    1450
                            1457    1465     1472    1473    1501    1503    1505    1507    1509    1511    1513    1515    1517
                            1519    1521     1523    1525    1527    1529    1531    1533    1586    1588    1595    1603    1611
                            1619    1627     1635    1643    1654    1665

. ABS.   000000      000
```

```
         006232      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XVSAC0,XVSAC0/SOL/CRF:SYM=DDXCOM,XVSAC0
 RUN-TIME: 3 4 .7 SECONDS
 RUN-TIME RATIO: 30/8=3.4
 CORE USED:  8K  (15 PAGES)
```