

VSAC DEC/X11 SVSTEM EXEPCISER MODULE
XVSACO.P11 12-OCT-78 12:23

MACY11 30A(1052) 12-OCT-78 17:07 PAGE 2

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E872C-MC
PRODUCT NAME: CXVSACC 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 FOUR FRAMES. THE MODULE DISPLAYS A SPECIAL TEST PATTERN THAT CONSISTS OF VS-60'S DISPLAY. THE MODULES WILL VERIFY TO THE BY EXECUTING ALL OF THE VS-60'S HIGH DESCRIPTION OF THE SUB-PICTURES. THE VS-60 OPERATIONS AND PROVIDE A HIGH DESCRIPTION OF THE DISPLAY INSTRUCTIONS. EACH SUB-PICTURE INCLUDES A LIST OF TABLED TO DISPLAY. THE SUB-PICTURES USED FOR THE SUB-PICTURE. BEFORE REGISTER TEST IS PERFORMED TO ENSURE THE TEST PATTERN CONFIRMS A READ/WRITE REGISTER TEST IS PERFORMED. IF BIT 0 OF SRI IS CLEARED, IN THE BASIC HARDWARE INTERFACED. AS EACH UNIQUE SUB-PICTURE IS ENTERED, THE DISPLAY NAME REGISTERS IS LOADED WITH A UNIQUE VALUE. IF AN ERROR IS DETECTED, REGISTER OF THE NAME REGISTER NAME CAN BE READ TO DETERMINE THE CURRENT SUB-PICTURE IF A PROBLEM ARISES. 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 REQUIREMENTS:
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.
EXECUTION TIME

- 4. VSA RUNNING ALONE ON PDP-11/05 TAKES APPROXIMATELY 60 SECONDS. RUN ON THE 32K BOUNDRY. WHEN RUNNING WITH "RELOCATION" ENABLED, THIS MODULE WILL ONLY RUN ON THE 8, 16 AND 24K RELOCATION PASSES (REF. 5.). 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, SRI: 0

REQUIRED PARAMETERS:

NONE, HOWEVER IF THIS MODULE IS NOT CONFIGURED ACROSS A 8, 16 AND 24K BOUNDARY,
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 BOUNDARY.

6. DEVICE/OPTION SETUP

THE VS-60 MUST HAVE THE POWER ON.

7. MODULE OPERATION

THE MODULE WILL BEGIN BY TESTING THE ABILITY 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 "SRI" IS TESTED. IF BIT 0 OF SRI
IS SET, THE MODULE WILL DESTROY THE ENTIRE VISUAL PICTURE BY LOADING
DISPLAY NOPS THRU THE PICTURE BUFFER. SETTING OF BIT 0 IN SRI WILL ENABLE
THE VS-60 TO ACCESS THE UNIBUS AT THE WORST CASE RATE.
RESETTING OF BIT 0 IN SRI WILL NOT RESTORE THE PICTURE BUFFER.

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

TEST PATTERN DESCRIPTION:

FRAME 0 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 LARGED
RECTANGLE. EACH SIDE OF THE RECTANGLE IS DISPLAYED
USING A DIFFERENT LINE TYPE IE: SOLID, DASH, DOT, DASH
-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 WAVES PATTERNS
ARE DISPLAYED. ON THE FIRST SINE WAVE APPEARS THE BOTTOM
SUPERIMPOSED ON A HORIZONTAL LINE ACROSS THE SCREEN. A
OF THE SCREEN AND EXPANDS FROM LEFT TO RIGHT. A
THE SECOND SINE WAVE APPEARS FROM TOP TO BOTTOM. A
VERTICAL LINE AT THE LEFT EXPANSION OF THE SINE WAVE
FROM BOTTOM TO TOP. THE EXPANSION OF THE SINE WAVE
WAVES IS A FUNCTION OF THE DISPLAY RATE. THE VS-60
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 OCTAGONS IS DISPLAYED
IN THE UPPER MODE. THE OCTAGONS ARE DISPLAYED
LONG VECTOR MODE, AND THE MIDDLE MOST USING
SHORT VECTOR MODE, AND THE INNER MOST TWO USING
RELATIVE POINT MODE. THE USE OF RELATIVE POINT
POINT MODE CAUSES THE TWO INNER OCTAGONS
TO BEACH DISPLAYED AS EIGHT INTENSIFIED POINTS
FOR EACH ONE. ALTERNATE OCTAGONS STARTING
WITH THE INNER ONE. ALTERNATE OCTAGONS TO TEST
IN THE OPERATION OF THE BLINK MODE. THE FOLLOWING MODES ARE USED

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 SCREENS. 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 UPPER CASE CHARACTERS LEFT TO RIGHT). (OCTAL CODES 100-137 AND 40-77 32 LOWER CASE ASCII TO RIGHT). THE SECOND PAIR DISPLAYS THE 140-177 SPECIAL CHARACTERS (OCTAL CODES 0-37 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 DISPLAYED EIGHT TO THE 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 REFERENCED BY TWO VECTORS. THE FIRST STARTS FROM THE LOWER LEFT TO UPPER RIGHT CORNER. THE SECOND STARTS FROM AT 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 MODE
LONCY 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 AN "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 FOLLOWING 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 BY NEXT IS THE LETTER
 "B". THE SUPER-SCRIPT ON "B" IS WITH THE SUPER-SCRIPT ON "A"
 NUMBERS 2 AND 5 ARE THE NEXT BYTES. SIZE AND ASCEND IS THE CODE
 BY HALF THE SIZE OF "B". SHOULD BE A RETURN TO THE
 PREVIOUS SIZE AND "Y" POSITION. SHOULD BE A RETURN
 AS THE INITIAL "B" THEN FOLLOWED BY THE NUMBERS 2 AND 5. IN SIZE
 WITH "SUB-SCRIPT ON" THE NUMBERS BY ONE SIZE. "B" WHICH WILL
 AND DECENDING BYTE IS THE SIZE AND "Y" POSITION.
 RETURN TO THE ORIGINAL SIZE AND "Y" POSITION.
 THE FOLLOWING MODES USED ARE IN THE SUB-PICTURE:

- POINT TO AN X,Y POSITION
- CHARS1
- CHARS2
- CHARS3
- CHRON
- SUPONFF
- SUBON
- SUBOFF
- POINT TO AN X,Y POSITION
- DISPLAY IN CHARACTER MODE
- LOAD CHAR. SCALE TO NORMAL MODE
- DISABLE SUPER-SCRIPT ASCII MODE
- DISABLE SUPER-SCRIPT ASCII MODE
- 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 EACH OF THE CORNERS OF FRAME 0,
 FOUR "B"'S ARE DISPLAYED. THE UPPER RIGHT CORNER OF FRAME 0,
 DIFFERENT SIZES STARTED FROM THE SMALLEST TO LARGEST.
 THE PATTERN IS REPEATED WITH THE CHARACTER "I" ENABLED.
 TO VERIFY CHARACTER ROTATE, THE SAME PROCEDURE IS REPEATED
 IN THE LOWER LEFT CORNER OF FRAME 0. THE FOUR LETTERS SHOULD
 APPEAR THE SAME AS ABOVE WITH THE EXCEPTION THE CHARACTERS
 SHOULD BE ROTATED BY 90 DEGREES.
 THE FOLLOWING MODES ARE USED IN THE SUB-PICTURE:

- POINT TO X,Y POSITION
- STATSA
- CHRR1
- DJSRR1
- CHRR2
- DJMPR
- DPOPNR
- CHARS0
- CHARS1
- CHARS2
- CHARS3
- POINT TO X,Y POSITION
- LOAD STATUS REG. ROTATE A SUB-ROUTINE
- ENABLE CHARACTER ROTATE A SUB-ROUTINE
- DISABLE JSR RELATIVE TO A SUB-ROUTINE
- DISABLE JMP RELATIVE TO A SUB-ROUTINE
- DISABLE POP AND NO RESTORE
- ENABLE CHARACTER SIZE 0
- ENABLE CHARACTER SIZE 1
- ENABLE CHARACTER SIZE 2
- ENABLE CHARACTER SIZE 3

THE SUB-PICTURE CONSISTS OF SIXTEEN DIFFERENT SIZE SQUARES STARTING FROM A COMMON POINT. THE COMMON POINT IS RELATIVE 0,0 FROM FRAME 1. THE VECTOR IS LOADED WITH THE LARGEST VALUE AND A SUB-PICTURE IS DISPLAYED WITH A 200 UNIT SQUARE. THE VECTOR SCALE IS INDICED BY IS ONE AND THEN IT IS DRAWN AGAIN. THE PROCEDURE IS REPEATED UNTIL ALL VALUES OF VECTOR SCALE HAVE BEEN LOADED. THE FRAME IS BEING DISPLAYED AT PLUS IX PLUS IY SECTOR. THE FOLLOWING MODES ARE USED IN THE SUB-PICTURE:

```

POINT TO X,Y COORDINATE
DJSRR TO A SUB-ROUTINE
DJMPR RELATIVE TO MORE DISPLAY CODE
LONGV VECTOR TO MODE 17
VCTR00-17 ENABLE VECTOR TO MODE 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 TO X,Y COORDINATE
BASICV BASIC VECTOR MODE
PATH0-7 ENABLE PATH (DIRECTION) TO BE DRAWN
  
```

FRAME3 L. STACK LEVEL TEST:

THE SUB-PICTURE DISPLAYS EIGHT STATES 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 SUB-ROUTINE. 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. EXECUTING THE VS60 INSTRUCTION SET.
WITH BIT 0 CLEARED THE VS60 WILL BE EXECUTED WITH VS60 NOP'S.
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 BOUNDARY, THIS MODULE WILL NOT BE RUN.
THIS CAN BE DEFEATED BY THE OPERATOR NOT CONFIGURING THE MODULE ACROSS
A 8, 16 AND 24K BOUNDARY AND CLEARING BIT 10 (2000) OF LOCATION "STAT" OF THIS MODULE.
THE MOTION OF THE TEST PATTERN CAN BE STOPPED BY DEPRESSING THE TEST
"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. ENVIRONMENT

#1 11/10 WITH 16K OF MEMORY
RK-11-D DISK CONTROLLER WITH 1 DRIVE CONSOLE
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 CONSOLE
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


```

386 ;VS-60 INSTRUCTION SET
387
388 100000 CHAR=100000 ;DISPLAY IN CHARACTER MODE
389 104000 SHORTV=104000 ;SHORT VECTOR
390 110000 LONGV=110000 ;LONG VECTOR MODE
391 114000 POINT=114000 ;POINT MODE
392 120000 GRAPHX=120000 ;GRAPH PLOT X MODE
393 124000 GRAPHY=124000 ;GRAPH PLOT Y MODE
394 120000 BASICV=GRAPHX ;BASIC VECTOR MODE
395 130000 RELATP=130000 ;RELATIVE POINT MODE
396 134000 BASICS=RELATP+4000 ;BASIC SHORT VECTOR MODE
397 144000 ABSVCT=144000 ;ABSOLUTE VECTOR MODE
398
399 010000 OFFST0=10000 ;
400 012000 OFFST1=12000 ;ENABLE OFFSET OF 0
401 014000 OFFST2=14000 ;ENABLE OFFSET OF 1
402 016000 OFFST3=16000 ;ENABLE OFFSET OF 2
403 ;ENABLE OFFSET OF 3
404 002000 INT0=2000 ;ENABLE INTENSITY LEVEL 0
405 002200 INT1=2200 ;1
406 002400 INT2=2400 ;2
407 002600 INT3=2600 ;3
408 003000 INT4=3000 ;4
409 003200 INT5=3200 ;5
410 003400 INT6=3400 ;6
411 003600 INT7=3600 ;LEVEL 7
412
413 LPOFF=100 ;
414 LPON=140 ;
415 BLKOFF=20 ;DISABLE BLINK
416 BLKON=30 ;ENABLE BLINK
417
418 000004 LINE0=4 ;ENABLE LINE TYPE 0
419 000005 LINE1=5 ;ENABLE LINE TYPE 1
420 000006 LINE2=6 ;ENABLE LINE TYPE 2
421 000007 LINE3=7 ;ENABLE LINE TYPE 3
422
423 002000 PATH0=2000 ;DIRECTION 0
424 006000 PATH1=6000 ;DIRECTION 1
425 012000 PATH2=12000 ;DIRECTION 2
426 016000 PATH3=16000 ;DIRECTION 3
427 020000 PATH4=20000 ;4
428 026000 PATH5=26000 ;5
429 032000 PATH6=32000 ;6
430 036000 PATH7=36000 ;7
431
432 160000 DJMP=160000 ;DISPLAY ABSOLUTE JUMP
433 161000 DJMPR=DJMP+BIT9 ;DISPLAY RELATIVE JUMP
434 162000 DJSR=DJMP+BIT10 ;DISPLAY JSR ABSOLUTE
435 163000 DJSRR=DJSR+BIT9 ;DISPLAY JSR RELATIVE
436
437 164000 DNOP=164000 ;
438 DPOP=DNOP+BIT10 ;POP AND RESTORE
439 DPOPNR=DNOP+BIT9 ;POP AND NO RESTORE
440 164000 CONSL0=DNOP ;CONSOLE 0
441 164400 CONSL1=DNOP+BIT8 ;CONSOLE 1

```

```

442 170000 STATSA=170000 ;
443 173400 DSTOP=173400 ;
444 170002 DMENU0=STATSA+BIT1 ;DISABLE MENU
445 170003 DMENU1=DMENU0+BIT0 ;
446
447 000200 LPLITE=200 ;
448 000300 LPDARK=300 ;
449 000400 ITAL0=40 ;DISABLE ITALIC CHARACTERS
450 000600 ITAL1=60 ;
451
452 174000 STATSB=174000 ;
453
454 000100 INCR=100 ;ENABLE "GRAPH PLOT INCREMENT REG. CHANGE"
455
456 154000 STATSC=154000 ;
457 155000 CHRR0=STATSC+BIT9 ;DISABLE CHAR ROTATE
458 155400 CHRR1=CHRR0+BIT8 ;
459
460 154200 CHARSO=STATSC+BIT7 ;LOAD CHARACTER SCALE TO 1/2
461 154300 CHARS1=CHARSO+BIT5 ;1
462 154300 CHARS2=CHARSO+BIT6 ;1/2
463 154340 CHARS3=CHARSO+BIT6+BIT5 ;2
464
465 154020 VCTR00=STATSC+BIT4 ;LOAD VECTOR SCALE REGISTER
466
467 176000 STATE=STATSB+BIT10 ;
468
469 176002 STRNG0=STATE+BIT1 ;DISABLE CHARACTER STRING TERMINATE
470 176003 STRNG1=STRNG0+BIT0 ;
471
472 176040 EDGE0=STATE+BIT5 ;DISABLE EDGE INTERRUPT
473 176060 EDGE1=EDGE0+BIT4 ;
474 150000 DNAME=150000 ;LOAD DISPLAY NAME REGISTER
475
476 ;MORE EQUATES
477
478 040000 INTX=BIT14 ;INTENSIFY
479 000177 MAXMUX=177 ;MAX. MENU X WIDTH
480 001777 MAXX=1777 ;MAX. X AXIS LENGTH
481 001777 MAXY=1777 ;MAX. Y AXIS LENGTH
482 000777 HALFX=MAXX/2 ;HALF OF MAXIMUM LENGTH
483 020000 MINUSX=20000 ;NEGATIVE SIGN BIT
484 020000 MINUSY=20000 ;NEGATIVE SIGN BIT
485 000100 MINSUV=100 ;NEGATIVE SIGN BIT <SHORT VECTOR MODE>
486
487 000021 SUPON=21 ;SUPER-SCRIPT ENABLE
488 000023 SUPOFF=23 ;SUPER-SCRIPT DISABLE
489 000022 SUBON=22 ;SUB-SCRIPT ENABLE
490 000024 SUBOFF=24 ;SUB-SCRIPT DISABLE

```

```

492 000000-
493 000000-
494
495
496
497
498 000000-
499 000000- 051526 041501 040
500 000005- 000
501 000006- 172000
502 000010- 000320
503 000012- 000
504 000013- 000
505 000014- 000001
506 000016- 000000
507 000020- 000000
508 000022- 000000
509 000024- 000000
510
511 000026- 142000
512 000030- 000316-
513 000032- 000324-
514
515 000034- 000000
516 000036- 000000
517 000040- 000000
518 000044- 000000
519 000046- 000000
520 000050- 000000
521 000052- 000000
522 000054- 000000
523 000056- 000000
524 000058- 000000
525 000060- 000000
526 000062- 000000
527 000064- 000000
528 000066- 000000
529 000070- 000000
530 000072- 000000
531 000074- 000000
532 000076- 000000
533 000100- 000000
534 000102- 000000
535 000104- 000000
536 000106- 000000
537 000110- 000000
538 000112- 000430-
539 000114- 000000
540 000116- 000000
541 000120- 000000
542 000122- 000075
543
544

```

```

IOMODP <VSAC> 172000,320,4,656,0,75
MODULE 142000,VSAC,172000,320,4,656,0,75
TITLE VSAC DEC/X11 SYSTEM EXERCISER MODULE
DDXCOM VERSION 6 23-MAV-78
- LIST BIN
*****
BEGIN:
MODNAM: - ASCII /VSAC / ;MODULE NAME.
XFLAG: - BYTE OPEN ;USED TO KEEP TRACK OF WBUF USAGE
ADDR: 172000+0 ;1ST DEVICE ADDR.
VECTOR: 320+0 ;1ST DEVICE VECTOR.
BR1: - BYTE PRTV4+0 ;1ST BR LEVEL.
BR2: - BYTE PRTV+0 ;2ND BR LEVEL.
DVID1: +1 ;DEVICE INDICATOR 1.
SR1: OPEN ;SWITCH REGISTER 1
SR2: OPEN ;SWITCH REGISTER 2
SR3: OPEN ;SWITCH REGISTER 3
SR4: OPEN ;SWITCH REGISTER 4
*****
STAT: 142000 ;STATUS WORD.
INIT: START ;MODULE START ADDR.
SPINT: MODSP ;MODULE STACK POINTER.
PASCNT: 0 ;PASS COUNTER.
ICONT: 0 ;# OF ITERATIONS PER PASS=0
ICOUNT: 0 ;LOC TO COUNT ITERATIONS
SOFCNT: 0 ;LOC TO SAVE TOTAL SOFT ERRORS
HRDCNT: 0 ;LOC TO SAVE TOTAL HARD ERRORS
SOFPAS: 0 ;LOC TO SAVE SOFT ERRORS PER PASS
HRDPAS: 0 ;LOC TO SAVE HARD ERRORS PER PASS
SYSCNT: 0 ;# OF SYS ERRORS ACCUMULATED
RANUM: 0 ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
CONFIG: 0 ;RESERVED FOR MONITOR USE
RES1: 0 ;RESERVED FOR MONITOR USE
RES2: 0 ;RESERVED FOR MONITOR USE
SVR0: OPEN ;LOC TO SAVE R0.
SVR1: OPEN ;LOC TO SAVE R1.
SVR2: OPEN ;LOC TO SAVE R2.
SVR3: OPEN ;LOC TO SAVE R3.
SVR4: OPEN ;LOC TO SAVE R4.
SVR5: OPEN ;LOC TO SAVE R5.
SVR6: OPEN ;LOC TO SAVE R6.
CSRA: OPEN ;ADDR OF CURRENT CSR.
SBADR: OPEN ;ADDR OF GOOD DATA, OR
ACSR: OPEN ;CONTENTS OF CSR.
WASADR: OPEN ;ADDR OF BAD DATA, OR
ASATT: OPEN ;STATUS REG CONTENTS.
ERRTYP: OPEN ;TYPE OF ERROR.
ASR: OPEN ;EXPECTED DATA.
AWAS: OPEN ;ACTUAL DATA.
RSRRT: RSRRT ;RESTART ADDRESS AFTER END OF PASS
WDT0: OPEN ;WORDS TO MEMORY PER ITERATION
WDFR: OPEN ;WORDS FROM MEMORY PER ITERATION
INTR: OPEN ;# OF INTERRUPTS PER ITERATION
IDNUM: 75 ;MODULE IDENTIFICATION NUMBER=75
-REPT SPSIZ ;MODULE STACK STARTS HERE.
-NLIST

```

```

548
549
550
551 000224-
552
553
554
555 000224- 003144-
556 000226- 000000
557 000230- 000000

```

```

-WORD 0
- LIST
- ENDR
MODSP:
*****
RBUFVA: FRAMED
RBUFGA: OPEN
RBUFEA: OPEN

```

```

554 ;GTPASS: 1
559 DELAY: 3 ;PASS COUNTER
560 DELAY1: 20 ;PICTURE MOTION DELAY FACTOR
561 GTPC: 172000 ;GRAPHLOT MOTION DELAY FACTOR
562 GTSR: 172002 ;DISPLAY PC (0,0,C)
563 GTKPOS: 172004 ;DISPLAY STATUS REG.
564 GTVPOS: 172006 ;DISPLAY X REGISTER
565 GTREL: 172010 ;DISPLAY Y REGISTER
566 GTSRI: 172012 ;DISPLAY RELOCATE REG.
567 GTKOFF: 172014 ;MISC. STATUS REG. #1
568 GTXOFF: 172016 ;DISPLAY X POS OFFSET REG
569 GTASNA: 172020 ;DISPLAY V POS OFFSET REG
570 GTCONS: 172022 ;DISPLAY ASSOC. NAME REG.
571 GTNAME: 172024 ;DISPLAY CONSOLE STATUS REGISTER
572 GTSAN: 172026 ;DISPLAY NAME REGISTER
573 GTTERM: 172030 ;DISPLAY STACK CONTENTS
574 GTSPT: 172032 ;DISPLAY CHARACTER TERMINATE REG.
575 GTZPOS: 172034 ;DISPLAY Z POS REG.
576 GTZOFF: 172036 ;DISPLAY Z OFFSET REG.
577 GTDONE: 320 ;DISPLAY DONE VECTOR
578 GTONE: 324
579 GTLPH: 324 ;DISPLAY LIGHT-PEN VECTOR
580 GTLPH1: 326
581 GTSOTM: 330 ;DISPLAY SHIFT-OUT/ TIME-OUT VECTOR
582 GTSOT1: 332
583 GTNAM#: 334 ;DISPLAY NAME MATCH VECTOR
584 GTNAM1: 336
585 ;INITILIZE VS-60 ADDRESSES AND VECTORS
586
587
588 000316 005767 177512 START: TST PASCNT ;HAS A PASS BEEN MADE YET?
589 000322 001042 HNE RESTRT ;YES LEAVE
590 000324 032767 000001 177464 BIT #BIT0,SRI ;NO - IS NOP OR INSTRUCT DESIRED
591 000332 001021 BNE 2S ;RR IF NOP
592 000334 012767 000707 177554 MOV #455,WDR ;455 WORDS FROM MEM/ITERATION
593 000342 012767 000010 177550 #,INTR ;# INTERRUPTS/ITERATION
594 000350 005767 177462 TST ICNT ;IS ICNT ZERO
595 000354 001404 BEQ 1S ;YES BR TO SET TO 1
596 000356 026727 177454 005000 CWP ICNT,#5000 ;YES IT LESS THAN 5000
597 000364 002421 BLT RESTRT ;OPERATOR MUST HAVE ALREADY SET IT UP
598 000366 012767 000001 177442 1S: MOV #1,ICNT ;IT WAS ZERO - SET TO 1
599 000374 000415 BR RESTRT
600 000376 012767 000001 177512 2S: MOV #1,WDR #1 WORD FR MEM/ITERATION
601 000404 012767 000001 177596 CWP #1,INTR ;1 INTERRUPT/ITERATION
602 000412 026727 177420 005000 CMP ICNT,#5000 ;IS IT GREATER THAN ZERO
603 000420 003003 BGT RESTRT
604 000422 012767 060000 177406 MOV #6000,ICNT ;NO - USE 60000 TO START WITH
605 000430 012701 000238 RESTRT: MOV #GTPC,R1 ;LOAD POINTER
606 000434 016700 177346 ADDP R0 ;LOAD VALUE
607 000440 010021 MOV R0,(R1)+ ;LOAD VALUE INTO ADDRESS
608 000442 005720 TST (R0)+ ;TEST FOR BUS ERROR AND UPDATE RO
609 000444 022701 000276 CWP #CTDONE,R1 ;TEST IF DONE ADDRESS SETUP
610 000450 001373 BNE 4S ;BR IF NOT
611 000452 016700 177332 MOV VECTOR,R0 ;LOAD VECTOR ADDRESS
612 000456 010021 MOV R0,(R1)+ ;LOAD VECTOR VALUE
613 000460 005720 TST (R0)+ ;ADJUST R0

```

```

614 000462 022701 000316 CWP #START,R1 ;TEST IF DONE VECTOR SETUP
615 000466 001373 BNE 2S ;BR IF NOT
616 000470 104415 000000 000224 GETPAS,BEGIN,RBUFVA ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
617 000476 016700 177526 MOV RBUFVA,R0 ;GET EA BITS
618 000502 006000 ROR R0
619 000504 006000 ROR R0
620 000506 000300 SWAP R0 ;MOVE BITS 4 + 5 INTO 10 AND 11
621 000510 042700 171777 RIC #171777,R0 ;MASK
622 000514 010067 002376 MOV R0,EABITS ;SAVE MY EA BITS
623 000520 016767 177502 MOV RBUFVA,FILEOD ;LOAD PHYSICAL ADDRESS OF THE STARTING LOC. OF T
624 000526 05767 000004 005406 ADD #4,FILEOD ;UPDATE ADDRESS
625 000534 005067 002376 CLR ABORT ;CLEAR ABORT TESTING FLAG

```

```
626 ;TEST THAT THE X DYNAMIC OFFSET REGISTER CAN BE LOADED
627
628 XDOFF: MOV #RIT1,ASTAT ;LOAD EXPECTED VALUE
629 MOV CTXOFF,CSRA ;LOAD THE BUS ADDRESS
630 MOV ASTAT,&CTXOFF ;LOAD THE REGISTER
631 1S: MOV &CTXOFF,ACSR ;READ THE REGISTER
632 BIC #17000,ACSR ;MASK TO OTHER BITS
633 CMP ASTAT,ACSR ;TEST IF EQUAL
634 BEQ 2S ;BR IF SAME
635 MOV #25,ERRTYP ;BIT STUCK
636 ;*****
637 HRDRS,BEGIN,NULL ; X DYNAMIC OFFSET REGISTER FAILED TO LOAD PROPERLY
638 ;*****
639 BIS #RIT0,ABORT ;INDICATE "MAJOR ERROR"
640
641 2S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
642 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
643 ASR ASTAT ;ADJUST DATA PATTERN
644 BNE 1S ;BR IF MORE BITS TO TEST
645 CLR &CTXOFF ;ENSURE CLEAR REGISTER
646
647 ;TEST THAT THE Y DYNAMIC OFFSET REGISTER CAN BE LOADED
648
649 YDOFF: MOV #RIT1,ASTAT ;LOAD EXPECTED VALUE
650 MOV CTYOFF,CSRA ;LOAD THE BUS ADDRESS
651 1S: MOV ASTAT,&CTYOFF ;LOAD THE REGISTER
652 MOV &CTYOFF,ACSR ;READ THE REGISTER
653 BIC #17000,ACSR ;MASK TO OTHER BITS
654 CMP ASTAT,ACSR ;TEST IF EQUAL
655 BEQ 2S ;BR IF SAME
656 MOV #25,ERRTYP ;BIT STUCK
657 ;*****
658 HRDRS,BEGIN,NULL ; Y DYNAMIC OFFSET REGISTER FAILED TO LOAD PROPERLY
659 ;*****
660 MOV #RIT1,ABORT ;INDICATE "MAJOR ERROR"
661
662 2S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
663 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
664 ASR ASTAT ;ADJUST DATA PATTERN
665 BNE 1S ;BR IF MORE BITS TO TEST
666 CLR &CTYOFF ;ENSURE CLEAR REGISTER
667
```

```
668 ;TEST THAT THE RELOCATE REGISTER CAN BE LOADED
669
670 RELTST: MOV #RIT1,ASTAT ;LOAD EXPECTED VALUE
671 MOV CTREL,CSRA ;LOAD THE BUS ADDRESS
672 1S: MOV ASTAT,&CTREL ;LOAD THE REGISTER
673 MOV &CTREL,ACSR ;READ THE REGISTER
674 BIC #17000,ACSR ;MASK TO OTHER BITS
675 CMP ASTAT,ACSR ;TEST IF EQUAL
676 BEQ 2S ;BR IF SAME
677 MOV #25,ERRTYP ;BIT STUCK
678 ;*****
679 HRDRS,BEGIN,NULL ; RELOCATE REGISTER FAILED TO LOAD PROPERLY
680 ;*****
681 BIS #RIT2,ABORT ;INDICATE "MAJOR ERROR"
682
683 2S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
684 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
685 ASR ASTAT ;ADJUST DATA PATTERN
686 BNE 1S ;BR IF MORE BITS TO TEST
687 CLR &CTREL ;ENSURE CLEAR REGISTER
688
689 ;TEST THAT THE D.P.C. REGISTER CAN BE LOADED
690
691 DPCTST: MOV #RIT12,&GTSPTT ;SET MAINT SW #1
692 MOV #RIT1,ASTAT ;LOAD EXPECTED VALUE
693 MOV CTPC,CSRA ;LOAD THE BUS ADDRESS
694 1S: MOV ASTAT,&CTPC ;LOAD THE REGISTER
695 MOV &CTPC,ACSR ;READ THE REGISTER
696 CMP ASTAT,ACSR ;TEST IF EQUAL
697 BEQ 2S ;BR IF SAME
698 MOV #25,ERRTYP ;BIT STUCK
699 ;*****
700 HRDRS,BEGIN,NULL ; D.P.C. FAILED TO LOAD PROPERLY
701 ;*****
702 BIS #RIT3,ABORT ;INDICATE "MAJOR ERROR"
703
704 2S: BREAKS,BEGIN ;TEMPORARY RETURN TO MONITOR...
705 BREAKS,BEGIN ;THEN CONTINUE AT NEXT INSTRUCTION.
706 ASL ASTAT ;ADJUST DATA PATTERN
707 BNE 1S ;BR IF MORE BITS TO TEST
708 CLR &CTPC ;ENSURE CLEAR REGISTER
709 CLR &GTSPTT ;CLEAR MAINT SW #1
710
711 ;NOW TEST IF ANY "MAJOR ERRORS" HAVE BEEN FOUND
712 ; AND DROP MODULE IF ANY WERE PRESENT
713
714 DONTST: TST ABORT ;TEST FOR "MAJOR ERRORS"
715 BEQ BEGNA ;BR IF NONE AND START DISPLAY SECTION
716 ENDS,BEGIN ;DROP MODULE BECAUSE OF A FATAL REGISTER ERROR
```

```

717 ;TEST IF "DISPLAY NOP MODE"
718 REGNA: TST FAST ;TEST IF RUNNING "FAST" MODE
719 BNE 2S ;BR IF YES
720 BIT #BIT0,SRI ;TEST IF BIT 0 OF SRI = 1
721 BEQ 3S ;BR IF CLEARED
722 INC FAST ;SET FLAG
723 MOV #FRAME0,R0 ;LOAD PNTNTR
724 MOV #NDOP,(R0)+ ;LOAD DISPLAY NOP INTO BUFFER
725 CMP #FILE00,R0 ;TEST IF AT END
726 BNE 1S ;BR IF NOT
727 MOV #NOPDON,@CTDONE 2S: ;LOAD RETURN VECTOR ON STOP INTR.
728 MOVB SRI,@GTDBE1 ;LOAD RETURN BR LEVEL
729 BR 5S ;BR TO START DISPLAY
730
731 ;NORMAL INTERRUPT VECTOR SETUP
732
733 3S: MOV #CTSTOP,@CTDONE ;LOAD STOP VECTOR
734 MOVB SRI,@CTDBE1 ;LOAD LIGHT-PEN VECTOR
735 MOV #GTLPEN,@GTLPH ;LOAD SHIFT-OUT VECTOR
736 MOVB SRI,@GTLPH ;LOAD SHIFT-OUT VECTOR
737 MOV #GTSOFT,@GTSOFT ;LOAD NAME MATCH VECTOR
738 MOVB SRI,@GTSOFT ;LOAD NAME MATCH VECTOR
739 MOV #GTMACH,@GTMACH ;LOAD PICTURE MOTION DELAY FACTOR
740 MOVB SRI,@GTMACH ;LOAD GRAPHPLOT DELAY FACTOR
741 MOV #DELAY,@CTDLY0 ;CLEAR "STOP MOTION" FLAG
742 MOV #DELAY,@CTDLY1 ;TEST IF FIRST TIME EVER
743 CLR MOTION ;BR IF NOT
744 TST FIRST ;SAVE DISPLAY JUMP RELATIVE
745 BNE 4S ;ENSURE NOT THE FIRST TIME IS SET
746 MOV #FILE0A,JMPFOB ;LOAD RELATIVE JUMP OVER LIGHT PEN #0 MESSAGE
747 MOV #FILE0B,JMPFOC ;LOAD RELATIVE JUMP OVER LIGHT PEN #1 MESSAGE
748 #1,FIRST ;LOAD GRAPHPLOT INCREMENT VALUE
749 MOV JMPFOB,FILE0A ;RESET CPU STACK POINTER
750 MOV JMPFOC,FILE0B ;RESET PEN SWITCH MESSAGE FOR #0
751 #1,GRPINC ;LOAD VALUE TO BE LOADED INTO X DYNAMIC OFFSET R
752 #2000,DLTXRG ;LOAD VALUE TO BE LOADED INTO Y DYNAMIC OFFSET R
753 #1,GRPINC ;LOAD X DYNAMIC OFFSET
754 #1,GRPINC ;LOAD Y DYNAMIC OFFSET
755 #1,GRPINC ;PRESET THE ASSOCIATIVE NAME VALUE
756 #1,GRPINC ;PRESET THE ASSOCIATIVE NAME MATCH
757 #1,GRPINC ;SET INSTRUCTION
758 #1,GRPINC ;ENABLE NAME MATCH INTERRUPT
759 #1,GRPINC ;LOAD D.P.U. RELOCATE REG.
760 #1,GRPINC ;START DISPLAY
761 #1,GRPINC ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
762 #1,GRPINC
763 #1,GRPINC
764 #1,GRPINC
765 #1,GRPINC
766 #1,GRPINC
  
```

```

767 ;RETURN HERE IF INTERRUPT VIA DONE (STOP) FLAG AND BIT 0 OF SRI IS CLEARED
768
769 001572"
770 GTSTOP:
771 001572" 000004 000000" 001603"
772 ;PIRQ0,BEGIN,STOVRT ;QUEUE UP TO CONTINUE AT STOVRT AND RTI
773 MOV #GTNAME,ASTAT ;READ NAME REGISTER
774 MOV #CTSR,CSRA ;LOAD BUS ADDRESS OF STATUS REG #1
775 MOV #GTSR,ACSR ;READ THE STATUS REGISTER
776 BPL EXSTOP ;BR IF EXTERNAL STOP EVENT
777
778 ;TEST THAT A NAME MATCH INTERRUPT HAS OCCURRED BEFORE STOP INTERRUPT
779
780 001624" 005767 001264 TST NAMESW ;TEST NAME INTR. SOFT FLAG
781 001630" 011006 BNE 1S ;BR IF NAME INTR. OCCURRED
782 001632" 012767 000023 176246 MOV #23,ERRTYP ;DEV FAILED TO INTERRUPT
783 *****
784 001640" 104405 000000" 000000 HRDERS,BEGIN,NULL ;NAME MATCH INTERRUPT FAILED TO OCCUR BEFORE STOP INTERR
785 *****
786
787 001646" 005067 001242 1S: CLR NAMESW ;CLEAR NAME INTR. SOFT FLAG
788 001652" 005767 001242 TST MOTION ;TEST IF "STOP MOTION" IS SET
789 001656" 001074 BNE RESTR ;BR IF YES
790 001660" 005367 001242 DEC CTDLY0 ;DECREMENT PICTURE MOTION DELAY
791 001664" 001074 BNE RESTR ;RESET PICTURE MOTION DELAY
792 001666" 016767 176340 001232 MOV #DELAY,@CTDLY0 ;TEST IF NEGATIVE POLARITY ?
793 001674" 032767 020000 001235 BIT #BIT13,DLTXRG ;BR IF NEG.
794 001702" 001320 BNE 2S ;ADJUST X DYNAMIC OFFSET
795 001704" 162767 000001 001222 SUB #1,DLTXRG ;ADJUST Y DYNAMIC OFFSET
796 001712" 162767 000001 001222 #1,DLTXRG ;TEST IF FINISHED ALL POS. OFFSETS ?
797 001720" 026727 001214 000000 CMP #DLTXRG,#0 ;BR IF NOT
798 001726" 001031 BNE 3S ;PRESET X OFFSET VALUE
799 001730" 012767 020000 001202 MOV #MINUSX,DLTXRG ;PRESET Y OFFSET VALUE
800 001736" 026767 000001 001166 ADD #1,DLTXRG ;UPDATE X DYNAMIC OFFSET VALUE
801 001740" 026767 000001 001166 ADD #1,DLTXRG ;UPDATE Y DYNAMIC OFFSET VALUE
802 001752" 062767 000001 001162 ADD #1,DLTXRG ;TEST IF FINISHED ALL NEG. OFFSETS ?
803 001760" 026727 001154 024000 CMP #DLTXRG,#MINUSX+4000 ;BR IF NOT
804 001766" 001074 BNE #2000,DLTXRG ;RELOAD X DYNAMIC OFFSET VALUE
805 001770" 012767 002000 001142 MOV #2000,DLTXRG ;RELOAD Y DYNAMIC OFFSET VALUE
806 001776" 012767 002000 001136 ENDITS,BEGIN ;SIGNAL END OF ITERATION.
807 002004" 104413 000000" ;MONITOR SHALL TEST END OF PASS
808
809 002010" 000417 BR RESTR
  
```

```

810
811
812
813 002012 005367 001112
814 002016 001014
815 002020 016767 176210 001102
816 002026 005267 001204
817 002032 022767 174110 001176
818 002040 011067 174100 001166
819 002046 011067 174100 001166
820 002050 016767 001056 003672
821 002056 016767 001052 003722
822 002064 016777 001050 176160
823 002072 016777 001044 176154
824 002100 007777 000001 176130
825 002106 104400 000000
826
827
828
829
830 002112 016767 176132 175760
831 002120 017767 176124 175754
832 002126 105767 175750
833 002132 100010
834 002134 012767 000011 175744
835 002142 104405 000000 000000
836
837 002150 000167 176142
838 002154 012767 000011 175724
839
840 002162 104405 000000 000000
841
842 002170 000167 176122
843
844
845 002174
846
847 002174 000004 000000 002202
848
849 002202 104413 000000
850 002202 104413 000000
851
852 002206 000734

```

```

853
854
855
856
857 002210 000004 000000 002216
858
859 002216 017767 176016 175656
860 002224 016767 176008 175640
861 002230 005067 000276
862 002240 005067 000276
863 002244 032767 000004 175630
864 002252 001410 000011 175624
865 002254 012767 000011 175624
866
867 002262 104405 000000 000000
868
869 002270 000167 176022
870
871 002274 017767 175760 175600
872 002302 016767 175752 175570
873 002310 032767 040000 175564
874 002316 014057
875 002320 012767 164000 003422
876 002330 005267 000210
877 002332 032767 020000 175542
878 002340 001410 003576 002410
879 002342 016767 000544
880 002350 005267 000544
881 002354 005267 000162
882 002360 000413
883 002370 001407 010000
884 002372 016767 003550 002360
885 002400 005067 000514
886 002404 005267 000132
887 002408 005267 000400 175464
888
889 002416 001405
890 002420 012767 164000 003360
891 002426 005267 000110
892 002432 032767 000200 175442
893 002440 014107
894 002444 016767 003476 002364
895 002450 005267 000444
896 002454 005267 000062
897 002460 000413
898 002462 032767
899 002470 001407
900 002472 016767 003450 002334
901 002480 005067 000414
902 002484 005267 000032
903 002510 005267 000026
904 002514 001402
905 002516 000167 177356
906 002522 012767 000011 175356
907
908 002530 104405 000000 000000

```

```

909 ;*****
910 JMP START ;START AGAIN
911 002536 000167 175554 20S: C ;NON-ZERO IF THE FLAG WAS KNOWN
912 ;RETURN HERE IF A MISC. VS60 INTERRUPT
913
914 GTSHIF:
915 002544
916 -----
917 002544 000004 000000 002552 PIRQS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
918 -----
919 002552 017767 175504 175324 1S: MOV #GTNAME,ASTAT ;READ D.P.U. NAME REGISTER
920 002560 016767 175454 175314 MOV #GTSR,ACSR ;LOAD BUS ADDRESS
921 002566 017767 175446 175306 MOV #GTSR,ACSR ;READ REGISTER
922 002574 032767 000100 175300 BIT #BIT6,ACSR ;TEST IF "SHIFT-OUT" FLAG WAS SET
923 ;BR IF NOT
924 002580 012767 000044 175274 MOV #4,ERRTYP ;FLAG SHOULD NOT BE SET
925 ;*****
926 002612 104405 000000 000000 HRDRS,BEGIN,NULL ;UNEXPECTED SHIFT-OUT FLAG SET
927 ;*****
928 002620 000167 175472 JMP START ;START AGAIN
929 002624 016767 175420 175246 2S: MOV #GTSR1,CSRA ;LOAD BUS ADDRESS
930 002630 017767 175412 175242 MOV #GTSR1,ACSR ;READ REGISTER
931 002640 042767 003515 175234 BIC #3515,ACSR ;MASK TO UNWANTED BITS
932 002646 001410 BEQ 3S ;BR IF NONE
933 002650 012767 000044 175230 MOV #4,ERRTYP ;FLAG SHOULD NOT BE SET
934 ;*****
935 002656 104405 000000 000000 HRDRS,BEGIN,NULL ;VS-60 MAJOR ERROR FLAG WAS SET
936 ;*****
937 002664 000167 175426 JMP START ;START AGAIN
938 002670 012767 000011 175210 3S: MOV #11,ERRTYP ;ILLEGAL INTERRUPT
939 ;*****
940 002676 104405 000000 000000 HRDRS,BEGIN,NULL ;INTERRUPT DETECTED BUT NO FLAG WAS SET
941 ;*****
942 002704 000167 175406 JMP START ;START AGAIN
943

```

```

944 ;RETURN HERE IF NAME MATCH INTERRUPT
945
946 GTMATCH:
947
948 002710
949 002710 000004 000000 002716 PIRQS,BEGIN,1S ; QUEUE UP TO CONTINUE AT 1S AND RTI
950 -----
951 002716 017767 175340 175160 1S: MOV #GTNAME,ASTAT ;READ NAME REGISTER
952 002724 016767 000162 175150 MOV #GTNAME,ACSR ;READ REGISTER AGAIN
953 002732 016767 175324 175140 MOV #GTNAME,CSRA ;LOAD BUS ADDRESS
954 002740 005767 175140 TST #ASTAT ;TEST IF NAME MATCH FLAG IS SET ?
955 002744 00406 BMT 4S ;BR IF YES
956 002746 012767 000011 175132 MOV #11,ERRTYP ;ILLEGAL INTERRUPT
957 ;*****
958 002754 104405 000000 000000 HRDRS,BEGIN,NULL ;INTERRUPT DETECTED BUT NO FLAG WAS SET
959 ;*****
960 002762 042767 170000 175114 4S: BIC #170000,ASTAT ;MASK TO BITS
961 002770 006767 175110 175104 CMP #ASTAT,ACSR ;TEST IF EXPECTED
962 002776 001410 BEQ 2S ;BR IF SAME
963 003000 012767 000011 175100 MOV #11,ERRTYP ;ILLEGAL INTERRUPT
964 ;*****
965 003006 104405 000000 000000 HRDRS,BEGIN,NULL ;UNEXPECTED NAME MATCH INTERRUPT
966 ;*****
967 003014 000167 175276 JMP START ;START AGAIN
968 003020 006767 000066 000060 2S: ASR #NAME,ANAME ;ADJUST THE EXPECTED NAME VALUE
969 003024 022767 000040 000060 CMP #BITS,ANAME ;TEST IF COMPLETE
970 003032 001003 BNE 3S ;BR IF NOT
971 003034 012767 002000 000050 MOV #BIT10,ANAME ;YES, RESET EXPECTED NAME VALUE
972 003042 016767 006044 000040 COPY #NAME
973 003050 052767 054000 000032 BIS #BIT14|BIT12|BIT11,10S ;ADD SEARCH CODE AND ENABLE
974 003056 016777 000026 175172 MOV #10,GTASNA ;LOAD ASSOCIATIVE NAME REGISTER
975 003064 016767 000022 001632 JMP #NAME,NMATCH ;LOAD THE DISPLAY BUFFER VALUE
976 003072 052767 150060 001624 BIC #NAMES,NMATCH ;LOCATE ON
977 003100 005267 000010 SET #SOFT SWICH SAVING MATCH OCCURRED
978 003104 000167 176770 JMP CONT ;CONTINUE THE SUP-PICTURE
979 10S: BIT14|BIT12|BIT11|BIT10 ;SEARCH CODE AND DISPLAY NAME VALUE
980 ANAME: BIT10
981 NAMESW: 0
982
983 EABITS: 0
984 MOTION: 0 ;NON-ZERO IF STOP MOTION
985 FAST: 0
986 FIRST: 0
987 GTDLV0: 100
988 GTDLV1: 100
989 JMPF0B: 0
990 JMPF0C: 0
991 ABORT: 0
992 DLTTRG: 0
993 003142 000000 DLTTRG: 0

```

994
995 003144* 164374
996 003146* 164774
997
998
999 003150* 150000
1000 003152* 114140
1001 003154* 000000
1002 003156* 001777
1003 003160* 174200
1004 003162* 113004
1005 003164* 041777
1006 003166* 000000
1007 003170* 110005
1008 003172* 040300
1009 003174* 021777
1010 003176* 110006
1011 003200* 061777
1012 003202* 000300
1013 003204* 110307
1014 003208* 240000
1015 003210* 001777
1016
1017 003212* 150001
1018 003214* 114000
1019 003216* 000000
1020 003220* 000200
1021 003222* 110000
1022 003224* 041200
1023 003226* 000300
1024 003228* 114000
1025 003232* 000440
1026 003234* 000200
1027 003236* 174104
1028 003240* 150902
1029 003242* 000000
1030 003244* 163322
1031 003246* 150003
1032 003250* 163320
1033
1034 003252* 114000
1035 003254* 000200
1036 003256* 000340
1037 003260* 150904
1038 003262* 110000
1039 003264* 000000
1040 003266* 001200
1041 003270* 114300
1042 003272* 000200
1043 003274* 000100
1044 003276* 000065
1045 003300* 120000
1046 003302* 163003
1047 003304* 150906
1048 003310* 123900
1049 003310* 161115

;FRAME CY OV
FRAME0: CONSL0IBIT7IBIT6IBIT5IBIT4IBIT3IBIT2 ;ENABLE CONSOLE 0
CONSL1IBIT7IBIT6IBIT5IBIT4IBIT3IBIT2 ;ENABLE CONSOLE 1
;DISPLAY OUTER REF. BOX WITH DIFFERENT LINE TYPES
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT I L P O N
0
MAXY
STATSR I L P L I T E
LONGV I I N T 4 I L I N E 0
INTX I M A X X ;TOP LINE
0
LONGV I L I N E 1 ;RIGHT LINE
INTX
WINUSY I M A X Y
LONGV I L I N E 2 ;BOTTOM LINE
INTX I M I N U S X I M A X X
0
LONGV I L I N E 3 ;LEFT LINE
INTX
MAXY
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT I L I N E 0
400
200
LONGV ;DRAW BASE REF. VECTOR
INTX+1200
0
POINT
440
200
COPINC: STATSR I I N C R + 4
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
GRAPHY
DJSRR I X L ;DJSR RELATIVE TO THE TAG "SINE"
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
DJSRR I X L ;DJSR RELATIVE TO THE TAG "SINE"
POINT
200
40
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
LONGV
INTX
1200
POINT
200
100
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
GRAPHY
DJSRR I X L ;DJSR RELATIVE TO THE TAG "SINE"
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
DJSRR I X L ;DJSR RELATIVE TO THE TAG "SINE"
DJSRR I X L ;DJSR RELATIVE TO THE TAG "SINE"
DJSRR I X L ;DJSR RELATIVE TO THE TAG "OCTPIC"

1050
1051
1052
1053 003312* 000200 000205 000212
1054 003320* 000217 000224 000231
1055 003326* 000236 000243 000247
1056 003334* 000253
1057 003336* 000257 000262 000265
1058 003344* 000270 000272 000274
1059 003352* 000276 000277 000277
1060 003360* 000277
1061 003362* 000277 000276 000275
1062 003370* 000274 000272 000267
1063 003376* 000264 000261 000256
1064 003404* 000252
1065 003406* 000246 000241 000235
1066 003414* 000230 000223 000216
1067 003422* 000211 000203 000176
1068 003430* 000171
1069 003432* 000163 000155 000151
1070 003440* 000144 000137 000133
1071 003446* 000127 000123 000117
1072 003454* 000114
1073 003456* 000111 000106 000104
1074 003462* 000102 000101 000100
1075 003472* 000100 000100 000100
1076 003500* 000101
1077 003502* 000102 000104 000106
1078 003510* 000111 000113 000117
1079 003516* 000122 000126 000132
1080 003518* 000127
1081 003526* 000144 000151 000156
1082 003534* 000163 000170 000175
1083
1084 003542* 166000
1085

;DATA STRING FOR A SINE WAVE
SINE: .WORD 0200,0205,0212,0217,0224,0231,0236,0243,0247,0253
.WORD 0257,0262,0265,0270,0272,0274,0276,0277,0277,0277
.WORD 0277,0276,0275,0274,0272,0267,0264,0261,0256,0252
.WORD 0246,0241,0235,0230,0223,0216,0211,0203,0176,0171
.WORD 0163,0156,0151,0144,0137,0133,0127,0123,0117,0114
.WORD 0111,0106,0104,0102,0101,0100,0100,0100,0100,0101
.WORD 0102,0104,0106,0111,0113,0117,0122,0126,0132,0137
.WORD 0144,0151,0156,0163,0170,0175
DPOP ;DISPLAY POP AND RESTORE

1086 003544* 150007
1087 003544* 114000
1088 003546* 114000
1089 003550* 001434
1090 003552* 000724
1091 003554* 130030
1092 003558* 041600
1093 003560* 041607
1094 003562* 040007
1095 003564* 061600
1096 003566* 061600
1097 003570* 061707
1098 003572* 040107
1099 003574* 041707
1100 003576* 150010
1101 003600* 114000
1102 003602* 001430
1103 003604* 000710
1104 003606* 130020
1105 003610* 043600
1106 003612* 043617
1107 003614* 040017
1108 003616* 063617
1109 003620* 063600
1110 003622* 040117
1111 003624* 040117
1112 003626* 043717
1113 003630* 150011
1114 003632* 114000
1115 003634* 001420
1116 003636* 000660
1117 003640* 104030
1118 003642* 047600
1119 003644* 047637
1120 003646* 040037
1121 003650* 067537
1122 003652* 067600
1123 003654* 067737
1124 003656* 040137
1125 003658* 040137
1126 003662* 150012
1127 003664* 114000
1128 003666* 001400
1129 003668* 000600
1130 003672* 104020
1131 003674* 057600
1132 003676* 057677
1133 003700* 040077
1134 003702* 040077
1135 003704* 077600
1136 003706* 077777
1137 003710* 040177
1138 003712* 057777

OCTPIC:

DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
1434
724
RELATPIBLKON
INTX+1600 ;SHORT VECTOR OCTAGON
INTX+1600+7
INTX+7
INTXIMINUSX+1600+7
INTXIMINUSX+1600
INTXIMINUSX+1600+MINSUY+7
INTX+MINSUY+7
INTX+1600+MINSUY+7
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
1430
710
RELATPIBLKOFF
INTX+3600 ;SHORT VECTOR OCTAGON
INTX+3600+17
INTX+17
INTXIMINUSX+3600+17
INTXIMINUSX+3600
INTXIMINUSX+3600+MINSUY+17
INTX+MINSUY+17
INTX+3600+MINSUY+17
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
1420
660
SHORTVIBLKON
INTX+7600 ;SHORT VECTOR OCTAGON
INTX+7600+37
INTX+37
INTXIMINUSX+7600+37
INTXIMINUSX+7600
INTXIMINUSX+7600+MINSUY+37
INTX+MINSUY+37
INTX+7600+MINSUY+37
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
1400
600
SHORTVIBLKOFF
INTX+17600 ;SHORT VECTOR OCTAGON
INTX+17600+77
INTX+77
INTXIMINUSX+17600+77
INTXIMINUSX+17600
INTXIMINUSX+17600+MINSUY+77
INTX+MINSUY+77
INTX+17600+MINSUY+77

1139 003714* 150013
1140 003716* 114030
1141 003720* 001360
1142 003724* 000520
1143 003724* 110000
1144 003726* 040137
1145 003730* 000900
1146 003732* 040137
1147 003734* 000137
1148 003736* 040000
1149 003740* 000137
1150 003742* 060137
1151 003744* 000137
1152 003746* 000137
1153 003750* 000000
1154 003752* 060137
1155 003754* 020137
1156 003756* 040000
1157 003760* 020137
1158 003762* 040137
1159 003764* 020137
1160 003766* 150014
1161 003770* 114040
1162 003772* 001540
1163 003774* 000440
1164 003776* 110000
1165 004000* 040177
1166 004002* 000900
1167 004004* 040177
1168 004006* 000177
1169 004010* 040000
1170 004014* 000177
1171 004016* 000177
1172 004018* 000177
1173 004020* 060177
1174 004022* 000000
1175 004024* 060177
1176 004026* 020177
1177 004030* 040000
1178 004032* 020177
1179 004034* 040177
1180 004036* 020177
1181

DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT IBLKON
1360
520
LONGV ;LONG VECTOR OCTAGON BY LENGTH OF 137
INTX+137
0
INTX+137
137
INTX
137
INTXIMINUSX+137
137
INTXIMINUSX+137
0
INTXIMINUSX+137
MINUSX+137
INTX
MINUSX+137
INTX+137
MINUSX+137
DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT IBLKOFF ILPOFF
1340
440
LONGV ;LONG VECTOR OCTAGON BY LENGTH OF 177
INTX+177
0
INTX+177
177
INTX
177
INTXIMINUSX+177
177
INTXIMINUSX+177
0
INTXIMINUSX+177
MINUSX+177
INTX
MINUSX+177
INTX
MINUSX+177
INTX+177
MINUSX+177

```

1182 ;DISPLAY CHARACTER SET
1183
1184 DNAME IVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1185 CHARST ;ENSURE NORMAL SIZE CHARS.
1186 POINT I L P O N
1187 100
1188 MAXY-100
1189 STATA I I T A L O
1190 CHAR
1191 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT1"
1192 STATA I I T A L 1
1193 DNAME I V O ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1194 POINT
1195 100
1196 MAXY-130
1197 CHAR
1198 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT1"
1199 STATA I I T A L O
1200 DNAME I V O ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1201 POINT
1202 220
1203 MAXY-200
1204 CHAR
1205 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT3"
1206 STATA I I T A L 1
1207 DNAME I V O ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1208 POINT
1209 220
1210 MAXY-230
1211 CHAR
1212 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT3"
1213 STATA I I T A L O
1214 DNAME I V O ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1215 POINT
1216 220
1217 MAXY-300
1218 CHAR
1219 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT2"
1220 STATA I I T A L 1
1221 DNAME I V O ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO
1222 POINT
1223 220
1224 MAXY-330
1225 CHAR
1226 DJSR R I X L ;DJSR RELATIVE TO THE TAG "PAT2"
1227 DJMP R I X L ;DJMP RELATIVE TO THE TAG "TITLE0"

```

```

1228 ;ASCII STRING FOLLOWED BY DPOP
1229 PAT1: .ASCII "0123456789:;<=>?@
1230 "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
1231 "abcdefghijklmnopqrstuvwxyz
1232 "0123456789:;<=>?@
1233 "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
1234 "abcdefghijklmnopqrstuvwxyz
1235 "0123456789:;<=>?@
1236 "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
1237 "abcdefghijklmnopqrstuvwxyz
1238 "0123456789:;<=>?@
1239 "ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
1240 "abcdefghijklmnopqrstuvwxyz
1241 "0123456789:;<=>?@
1242 DPOP ;DISPLAY POP AND RESTORE
1243
1244 ;SHIFT-OUT ASCII STRING
1245 PAT2: .BYTE 16,0,1,2,3,4,5,6,7,10,11,12,13,14,15,16
1246
1247
1248
1249
1250
1251
1252 .BYTE 20,21,22,23,24,25,26,27,30,31,32,33,34,35,36,37,17,0
1253
1254
1255
1256
1257 DPOP ;DISPLAY POP AND RESTORE
1258
1259
1260 ;LOWER CASE ASCII STRING
1261 PAT3: .BYTE 140,141,142,143,144,145,146,147
1262
1263
1264
1265 .BYTE 150,151,152,153,154,155,156,157
1266
1267
1268
1269 .BYTE 160,161,162,163,164,165,166,167
1270
1271
1272 .BYTE 170,171,172,173,174,175,176,177
1273
1274 DPOP ;DISPLAY POP AND RESTORE
1275

```

1274
1275
1276 004400*
1279 004406* 150023
1280 004402* 114000
1281 004404* 000400
1282 004406* 000320
1283 004410* 100000
1284 004412* 042504 027503 030530
1285 004420* 020061 042120 026520
1286 004426* 030464 051440 051531
1287 004434* 042524 020115 054135
1288 004442* 051105 044573 042523
1289 004450* 000122
1290 004452* 114000
1291 004454* 000400
1292 004456* 000320
1293 004460* 100000
1294 004462* 042504 043503 040522
1295 004470* 044120 041511 030455
1296 004476* 020061 051526 033055
1297 004504* 020060 046121 044120
1298 004512* 043501 040522 044120
1299 004520* 041511 042040 051511
1300 004526* 046120 054501 051440
1301 004534* 051531 042524 000115

;DISPLAY MODULE TITLE ON SCREEN

TITLE0: DNAME#V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0

POINT
500
320

CHAR
.ASCIZ "DEC/X11 PDP-11 SYSTEM EXERCISER"

POINT

400

20

CHAR

.ASCIZ "DECGRAPHIC-11 VS-6^ ALPHAGRAPHIC DISPLAY SYSTEM"

1302
1303
1304
1305 004542* 150024
1306 004544* 170340
1307 004546* 114000
1308 004550* 000340
1309 004552* 001300
1310 004554* 113604
1311 004556* 114000
1312 004560* 000200
1313 004562* 150025
1314 004564* 114000
1315 004566* 000340
1316 004570* 001240
1317 004572* 001200
1318 004574* 040400
1319 004576* 000300
1320 004600* 150126
1321 004602* 114000
1322 004604* 000340
1323 004606* 001200
1324 004610* 113200
1325 004612* 040400
1326 004614* 000400
1327 004616* 150327
1328 004620* 114000
1329 004622* 000340
1330 004624* 001140
1331 004626* 113000

;DISPLAY INTENSITY LEVELS

DNAME#V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0

STATUS#ITALO

POINT

340

1300

LONGV#INT7#LINE0

INTX#400

0

DNAME#V0

POINT

1240

LONGV#INT6

INTX#400

0

DNAME#V0

POINT

340

1200

LONGV#INT5

INTX#400

0

DNAME#V0

POINT

340

1200

LONGV#INT4

INTX#400

0

DNAME#V0

POINT

340

1140

LONGV#INT4

1332 004630* 040400
1333 004632* 000400
1334 004634* 150030
1335 004636* 114000
1336 004640* 000340
1337 004642* 001300
1338 004644* 112500
1339 004646* 040400
1340 004650* 000000
1341 004652* 150031
1342 004654* 114000
1343 004656* 000340
1344 004660* 001040
1345 004662* 112400
1346 004664* 040400
1347 004666* 000000
1348 004670* 150032
1349 004672* 114000
1350 004674* 000340
1351 004676* 001300
1352 004700* 112200
1353 004702* 040400
1354 004704* 000300
1355 004706* 150033
1356 004710* 114000
1357 004712* 000340
1358 004714* 000740
1359 004716* 112000
1360 004720* 040400
1361 004722* 000300
1362
1363
1364
1365
1366 004724* 150000
1367
1368
1369

INTX#400

0

DNAME#V0

POINT

340

1100

LONGV#INT3

INTX#400

0

DNAME#V0

POINT

340

1040

LONGV#INT2

INTX#400

0

DNAME#V0

POINT

340

1000

LONGV#INT1

INTX#400

0

DNAME#V0

POINT

340

740

LONGV#INT0

INTX#400

0

;ASSOCIATIVE NAME MATCH INTERRUPT SECTION

NMATCH: DNAME#0C

;VARIABLE VALUE FOR THE DISPLAY
;ASSOCIATIVE NAME -- UPON EXECUTION
; OF THIS INSTRUCTION A NAME MATCH
; INTERRUPT SHOULD OCCUR

```

1370 ;LIGHT-PEN SWITCH SECTION
1371 004726 150034 DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
1372 004730 164600 CONSLIBIT7 ;DISABLE CONSOLE #1
1373 004734 170000 POINTINT4
1374 004736 000340 340
1375 004736 000500 500
1376 004740 100000 CHAR
1377 004742 042520 020116 053523 .ASCII /PEN SWITCH IS /
1378 004750 052111 044103 044440
1379 004756 020123
1380 004760 050125 047440 020116 PENSWO: .ASCII /UP ON CONSOLE 0 / ;CHANGES TO "DN" UPON SWITCH DOWN
1381 004766 047503 051516 046117
1382 004774 020105 020060
1383 005000 164700 CONSLIBIT7BIT6 ;ENABLE CONSOLE #1
1384 005002 164200 CONSLIBIT7 ;DISABLE CONSOLE #0
1385 005004 150635 DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
1386 005006 114000 POINT
1387 005010 000340 340
1388 005012 000500 500
1389 005014 100000 CHAR
1390 005016 042520 020116 053523 .ASCII /PEN SWITCH IS /
1391 005024 052111 044103 044440
1392 005032 020123
1393 005034 050125 047440 020116 PENSW1: .ASCII /UP ON CONSOLE 1 / ;CHANGE TO "DN" UPON SWITCH DOWN
1394 005036 047503 051516 046117
1395 005050 020105 020061
1396 005054 164300 CONSLIBIT7BIT6 ;ENABLE CONSOLE #0
1397

```

```

1398 ;EDGE SCISSORING AT THE TOP OF FRAME 0
1399 EDGESC:
1400 DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
1401 POINT
1402 005056 150036 0
1403 005060 114000 WXY-100
1404 005062 000000 LONGV
1405 005064 001677
1406 005066 110000
1407 005070 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1408 005072 000200 200 ;VECTOR OFF SCREEN TO ON SCREEN
1409 005074 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1410 005076 020200 MINUSV1200 ;VECTOR OFF SCREEN TO ON SCREEN
1411 005100 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1412 005102 000200 200 ;VECTOR OFF SCREEN TO ON SCREEN
1413 005104 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1414 005106 020200 MINUSV1200 ;VECTOR OFF SCREEN TO ON SCREEN
1415 005110 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1416 005112 000200 200 ;VECTOR OFF SCREEN TO ON SCREEN
1417 005114 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1418 005116 020200 MINUSV1200 ;VECTOR OFF SCREEN TO ON SCREEN
1419 005120 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1420 005122 000200 200 ;VECTOR OFF SCREEN TO ON SCREEN
1421 005124 040200 INTX1200 ;VECTOR ON SCREEN TO OFF SCREEN
1422 005126 020200 MINUSV1200 ;VECTOR OFF SCREEN TO ON SCREEN
1423
1424 ;SUPER/SUB-SCRIPT CHARACTER SECTION
1425
1426 DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V"
1427 POINT
1428 005132 114000 1400
1429 005136 001400 1400
1430 005140 154240 ;NORMAL CHAR. SIZE
1431 005142 100000 CHAR
1432 005144 102 021 062 .BYTE 102,SUPON,62,65,SUPOFF
1433 005147 065 023 062 .BYTE 102,SUBON,62,65,SUBOFF
1434 005151 065 024 062 .BYTE 103,SUPON,123,124,SUPOFF
1435 005154 103 021 123 .BYTE 103,SUBON,123,124,SUBOFF
1436 005156 124 023 123 .BYTE 103,SUBON,123,124,SUBOFF
1437 005161 124 022 123 .BYTE 103,SUBON,123,124,SUBOFF
1438 005166 124 024
1439
1440
1441

```

```

1442 ;FOUR SCALED ROTATED LETTERS
1443
1444 DNAMEI V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0"
1445 POINT
1446 100
1447 0
1448 STATAI IAL0 ;NON-ITALIC
1449 CHRRTI ;ROTATE CHARACTERS
1450 DJSRRI XL ;DJSR RELATIVE TO THE TAG "SCLDCH"
1451 ;FOUR SCALED ROTATED ITALIZED LETTERS
1452 DNAMEI V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0"
1453 POINT
1454 40
1455 0
1456 STATAI IAL1 ;ENABLE ITALIC
1457 DJSRRI XL ;DJSR RELATIVE TO THE TAG "SCLDCH"
1458 ;FOUR SCALED ITALIZED LETTERS
1459 DNAMEI V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0"
1460 POINT
1461 1600
1462 1600
1463 STATAI IAL1
1464 CHRRTI ;NON-ROTATE, NON-ITALIC
1465 DJSRRI XL ;DJSR RELATIVE TO THE TAG "SCLDCH"
1466 ;FOUR SCALED LETTERS
1467 DNAMEI V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0"
1468 POINT
1469 1600
1470 1540
1471 STATAI IAL0
1472 DJSRRI XL ;DJSR RELATIVE TO THE TAG "SCLDCH"
1473 DJMPRI XL ;DJMP RELATIVE TO THE TAG "FRAME1"
1474
1475 SCLDCH: CHARSO ;CHAR SCALE 0
1476 CHAR
1477 -BYTE 102,0
1478 CHAR S1 ;CHAR SCALE 1
1479 CHAR
1480 -BYTE 102,0
1481 CHAR S2 ;CHAR SCALE 2
1482 CHAR
1483 -BYTE 102,0
1484 CHAR S3 ;CHAR SCALE 3
1485 CHAR
1486 -BYTE 102,0
1487 CHAR S1
1488 CHAR S1 ;CHAR SCALE 1
1489 DPOPNR ;DISPLAY POP AND NO RESTORE

```

```

1490 ;FRAME +X 1, +Y 1
1491 ;USE DJSRRI TO DRAW SCALED BOXES FROM +2000X, +2000Y
1492
1493 FRAME1:
1494 FXIPV1:
1495 DNAMEI V0 ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "V0"
1496 VCTROO I4
1497 POINT
1498 BIT0
1499 BIT1
1500 VCTROO IAO ;CHANGE VECTOR SCALE
1501 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1502 VCTROO IAO ;CHANGE VECTOR SCALE
1503 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1504 VCTROO IAO ;CHANGE VECTOR SCALE
1505 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1506 VCTROO IAO ;CHANGE VECTOR SCALE
1507 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1508 VCTROO IAO ;CHANGE VECTOR SCALE
1509 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1510 VCTROO IAO ;CHANGE VECTOR SCALE
1511 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1512 VCTROO IAO ;CHANGE VECTOR SCALE
1513 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1514 VCTROO IAO ;CHANGE VECTOR SCALE
1515 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1516 VCTROO IAO ;CHANGE VECTOR SCALE
1517 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1518 VCTROO IAO ;CHANGE VECTOR SCALE
1519 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1520 VCTROO IAO ;CHANGE VECTOR SCALE
1521 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1522 VCTROO IAO ;CHANGE VECTOR SCALE
1523 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1524 VCTROO IAO ;CHANGE VECTOR SCALE
1525 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1526 VCTROO IAO ;CHANGE VECTOR SCALE
1527 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1528 VCTROO IAO ;CHANGE VECTOR SCALE
1529 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1530 VCTROO IAO ;CHANGE VECTOR SCALE
1531 DJSRRI XL ;DJSR RELATIVE TO THE TAG "DRWBOX"
1532 VCTROO I4 ;RESET VECTOR SCALE
1533 DJMPRI XL ;DJMP RELATIVE TO THE TAG "FRAME2"
1534

```

1535 005430* 110000
1536 005432* 040500
1537 005434* 000000
1538 005436* 040000
1539 005440* 000500
1540 005442* 060500
1541 005444* 000000
1542 005446* 040000
1543 005450* 020500
1544 005452* 000000
1545 005454* 000000
1546 005456* 164000
1547 005460* 164000
1548 005462* 166000
1549
1550
1551
1552
1553
1554 005464*
1555 005464* 150045
1556 005466* 114000
1557 005470* 020777
1558 005472* 020777
1559 005474* 120000
1560 005476* 042777
1561 005500* 062777
1562 005502* 046777
1563 005504* 067777
1564 005506* 052777
1565 005510* 072777
1566 005512* 056777
1567 005514* 076777
1568 005516* 067777
1569 005520* 042777
1570 005522* 066777
1571 005524* 046777
1572 005526* 076777
1573 005530* 052777
1574 005532* 076777
1575 005534* 056777

DRWBOX: LONGV
INTX1500 ;DRAW A BOX
0
INTX
500
INTX1MINUSX1500
0
INTX
MINUSY1500
0
DNOP
DNOP ;DISPLAY POP AND RESTORE
DPOP
;FRAME -X1, -Y1
;USE "BASIC VECTOR" TO DRAW AN "STAR"
FRAME2:
NXINV1: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
MINUSX1HALFX
MINUSY1HALFX
BASICV
INTXIPATH01HALFX
INTXIPATH41HALFX
INTXIPATH31HALFX
INTXIPATH51HALFX
INTXIPATH21HALFX
INTXIPATH61HALFX
INTXIPATH71HALFX
INTXIPATH81HALFX
INTXIPATH91HALFX
INTXIPATH101HALFX
INTXIPATH111HALFX
INTXIPATH121HALFX
INTXIPATH131HALFX
INTXIPATH141HALFX
INTXIPATH151HALFX
INTXIPATH161HALFX
INTXIPATH171HALFX
INTXIPATH181HALFX
INTXIPATH191HALFX

1576
1577
1578
1579
1580 005536*
1581 005536* 150046
1582 005540* 114000
1583 005542* 020000
1584 005544* 005000
1585
1586 005546* 163001
1587
1588 005550* 161077
1589
1590 005552*
1591 005552* 150047
1592 005554* 100000
1593 005556* 042514 042526 020114
1594 005558* 005060
1595 005560* 163001
1596 005570* 166000
1597
1598 005572*
1599 005574* 150050
1600 005576* 100000
1601 005576* 042514 042526 020114
1602 005580* 005061
1603 005580* 153001
1604 005610* 166000
1605
1606 005612*
1607 005612* 150051
1608 005614* 100000
1609 005616* 042514 042526 020114
1610 005624* 005062
1611 005626* 163001
1612 005630* 166000
1613
1614 005632*
1615 005632* 150052
1616 005634* 100000
1617 005636* 042514 042526 020114
1618 005644* 005063
1619 005646* 163001
1620 005650* 166000
1621

;FRAME +X 2, +Y 2
;TEST ALL STACK LEVELS WORK PROPERLY
; DJSR DOWN 8 LEVELS AND DPOP BACK UP
FRAME3: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
POINT
BIT11
BIT1111000
DJSRR1XL ;DJSR RELATIVE TO THE TAG "LEVEL0"
DJMPR1XL ;DJMP RELATIVE TO THE TAG "FILE0A"
LEVEL0: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
CHAR
-ASCII /LEVEL 0/<12>
DJSRR1XL ;DJSR RELATIVE TO THE TAG "LEVEL1"
DPOP
LEVEL1: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
CHAR
-ASCII /LEVEL 1/<12>
DJSRR1XL ;DJSR RELATIVE TO THE TAG "LEVEL2"
DPOP
LEVEL2: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
CHAR
-ASCII /LEVEL 2/<12>
DJSRR1XL ;DJSR RELATIVE TO THE TAG "LEVEL3"
DPOP
LEVEL3: DNAMEIVO ;LOAD D.P.U. NAME REGISTER WITH THE VALUE OF "VO"
CHAR
-ASCII /LEVEL 3/<12>
DJSRR1XL ;DJSR RELATIVE TO THE TAG "LEVEL4"
DPOP

